Boltzmann machine learning with a variational quantum algorithm

Yuta Shingu, Yuya Seki, Shohei Watabe, Suguru Endo, Yuichiro Matsuzaki, Shiro Kawabata, Tetsuro Nikuni, and Hideaki Hakoshima

Phys. Rev. A **104**, 032413 — Published 16 September 2021

DOI: 10.1103/PhysRevA.104.032413

Boltzmann machine learning with a variational quantum algorithm

Yuta Shingu,[1,2] Yuya Seki,[2] Shohei Watabe,[1] Suguru Endo,[3,4] Yuichiro Matsuzaki,[2,*] Shiro Kawabata,[2] Tetsuro Nikuni,[1,†] and Hideaki Hakoshima[2,‡]

[1]*Department of Physics, Faculty of Science Division I, Tokyo University of Science, Shinjuku, Tokyo 162-8601, Japan.*
[2]*Device Technology Research Institute, National Institute of Advanced Industrial Science and Technology (AIST), 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan.*
[3]*Department of Materials, University of Oxford, Parks Road, Oxford OX1 3PH, United Kingdom*
[4]*NTT Computer and Data Science laboratories, NTT corporation, Musashino, Tokyo 180-8585, Japan*

A Boltzmann machine is a powerful tool for modeling probability distributions that govern the training data. A thermal equilibrium state is typically used for the Boltzmann machine learning to obtain a suitable probability distribution. The Boltzmann machine learning consists of calculating the gradient of the loss function given in terms of the thermal average, which is the most time consuming procedure. Here, we propose a method to implement the Boltzmann machine learning by using Noisy Intermediate-Scale Quantum (NISQ) devices. We prepare an initial pure state that contains all possible computational basis states with the same amplitude, and apply a variational imaginary time simulation. Readout of the state after the evolution in the computational basis approximates the probability distribution of the thermal equilibrium state that is used for the Boltzmann machine learning. We perform the numerical simulations of our scheme and confirm that the Boltzmann machine learning works well. Our scheme leads to a significant step toward an efficient machine learning using quantum hardware.

I. INTRODUCTION

Developing efficient learning algorithms for Boltzmann machine (BM) [1, 2] is an important issue of machine learning. A Boltzmann machine is a parametric stochastic model for the statistical machine learning, which aims to extract patterns from data sets. A restricted Boltzmann machine (RBM) [3–5], a variant of BM, is a versatile model to represent an unknown distribution behind a given data set because a RBM can approximate any discrete distribution [6]. A restricted Boltzmann machine has a variety of applications such as dimensionality reduction [7], collaborative filtering [8], classification [9], quantum simulation [10–12], topic modelling [13], feature learning [14], and deep learning [15–18]; thus developing efficient learning algorithms of BM has benefits for the applications of machine learning. However, the learning process of the BM requires calculations of expectation values with respect to the Gibbs state, which are in general computationally hard. To mitigate the computational cost, many approximate algorithms on classical computers have been developed [19–25].

Quantum machine learning has been proposed for enhancing the performance of machine learning tasks by using quantum computers. There are many algorithms of quantum machine learning with fault-tolerant quantum computers, such as quantum support vector machine [26], linear regression [27], data fitting [28], and quantum principal component analysis [29]. In the Noisy Intermediate-Scale Quantum (NISQ) era [30–32], great efforts have been devoted to develop quantum algorithms. Among them, variational quantum algorithms (VQAs) have been considered to be the first useful application on NISQ computers, because they only need shallow quantum circuits [33–37]. In VQAs, trial wave functions are generated from parametrized shallow depth quantum circuits and parameters are optimized by classical computers. Parametrized quantum circuits as a quantum analogue of a classical neural network may enable us to implement machine learning tasks, e.g., quantum circuit learning for supervised learning [38] and data-driven quantum circuit learning for generative modelling [39]. As quantum circuits are used for generating the trial wave function, one can leverage exponentially increasing Hilbert space in the number of qubits, which may enhance the representability of the model significantly.

The BM learning based on VQAs can be in principle realized; the required Gibbs state can be prepared by leveraging the thermofield-double technique combined with the variational imaginary time simulation [36]. Although this naive technique can implement even the quantum BM learning [40], this technique needs two copies of quantum states, which is not favorable for NISQ devices.

In this paper, we propose a new scheme to implement the BM learning by using NISQ devices based on the variational imaginary time simulation. We use a pure initial state that contains all possible computational basis states with the same amplitude (that is a separable state of $|+\rangle\cdots|+\rangle$, where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$ is an eigenstate of $\hat{\sigma}_x$), and perform the imaginary time evolution on the initial state to obtain the desired state. By reading out the state in the computational basis after applying the variational imaginary time simulation algorithm, the probability distribution becomes the same as what is required for the BM learning. Our proposal focuses

* matsuzaki.yuichiro@aist.go.jp
† nikuni@rs.tus.ac.jp
‡ hakoshima-hideaki@aist.go.jp

on special cases to simulate the classical BM where only diagonal terms of its Hamiltonian are relevant. The possible advantage of our scheme is that we do not need two copies of quantum states unlike the scheme in [36], and this reduces the number of qubits to implement the algorithm.

The rest of this paper is organized as follows. In section II, we explain the standard setup of BM learning. In section III, we give the outline of the variational imaginary time simulation. In section IV, we propose our algorithm of BM learning with the variational imaginary time simulation. In section V, we show the numerical results of our algorithm. In section VI, we compare our method with other schemes of BM learning. Finally, we conclude and summarize this paper in section VII.

II. BOLTZMANN MACHINE LEARNING

A Boltzmann machine is a generative model that generates binary data according to a certain probability distribution. In this paper, we focus on the fully visible BM, where all the visible units are connected to each other, because such a concise model is suitable to demonstrate our scheme for small systems.

Let $\boldsymbol{\sigma} \in \{-1,1\}^N$ be a set of binary parameters where σ_i denotes i-th component and N is the total number of the units. First, we define the learning model $P(\boldsymbol{\sigma}|\boldsymbol{u})$ of BM by the Boltzmann distribution:

$$P(\boldsymbol{\sigma}|\boldsymbol{u}) = \frac{1}{Z(\boldsymbol{u})} \exp\left[-H(\boldsymbol{\sigma},\boldsymbol{u})\right], \tag{1}$$

where $Z(\boldsymbol{u})$ is the partition function, and $H(\boldsymbol{\sigma},\boldsymbol{u})$ denotes the Ising Hamiltonian:

$$H(\boldsymbol{\sigma},\boldsymbol{u}) = -\sum_{i<j} J_{ij}\sigma_i\sigma_j - \sum_{i=1}^{N} h_i\sigma_i. \tag{2}$$

Here, $\boldsymbol{u}$ is a vector of the parameters of the Hamiltonian $\boldsymbol{u} = (J_{12}, J_{13}, \cdots, J_{N-1N}, h_1, \cdots, h_N)^T$.

We define an empirical distribution $P_D(\boldsymbol{\sigma})$ derived from a training data-set as follows:

$$P_D(\boldsymbol{\sigma}) = \frac{1}{D}\sum_{d=1}^{D} \delta(\boldsymbol{\sigma},\ \boldsymbol{\sigma}^{(d)}), \tag{3}$$

where $\delta(\boldsymbol{x},\ \boldsymbol{y})$ denotes a Kronecker delta of N dimensions, D denotes the total number of the training data, and $\boldsymbol{\sigma}^{(d)} \in \{-1,1\}^N$ $(d = 1, \cdots, D)$ denotes d-th binary values of the training data.

The purpose of BM is to find optimal parameters $\boldsymbol{u}$ such that the distribution in Eq. (3) can be well reproduced by $P(\boldsymbol{\sigma}|\boldsymbol{u})$. For this purpose, we need to minimize the Kullback-Leibler divergence (KLD):

$$\mathrm{KL}(P_D||P) = \sum_{\boldsymbol{\sigma}\in\{-1,1\}^N} P_D(\boldsymbol{\sigma}) \log \frac{P_D(\boldsymbol{\sigma})}{P(\boldsymbol{\sigma}|\boldsymbol{u})}, \tag{4}$$

which is a measure to quantify a distance between two probability distributions. The KLD is always greater than or equal to zero, $\mathrm{KL}(P_D||P) \geq 0$, and the KLD becomes zero if and only if two probability distributions are identical. Minimization of the KLD over the parameter $\boldsymbol{u}$ is equivalent to the maximum likelihood estimation as a function of $\boldsymbol{u}$. We use the gradient method to minimize the KLD over $\boldsymbol{u}$ at the s-th update:

$$\boldsymbol{u}[s+1] = \boldsymbol{u}[s] - \eta \left[\frac{\partial}{\partial \boldsymbol{u}}\mathrm{KL}(P_D||P)\right]\Big|_{\boldsymbol{u}=\boldsymbol{u}[s]}, \tag{5}$$

where η denotes a learning rate satisfying $0 < \eta \ll 1$, $\boldsymbol{u}[s] = (J_{12}[s], \cdots, J_{N-1N}[s], h_1[s], \cdots, h_N[s])^T$ denotes the set of the learning parameter at the s-th update, and $\partial/\partial\boldsymbol{u} = (\partial/\partial J_{12}, \cdots, \partial/\partial J_{N-1N}, \partial/\partial h_1, \cdots, \partial/\partial h_N)^T$ denotes the nabla operator of the parameter $\boldsymbol{u}$. By rewriting the derivative in Eq. (5), we obtain the following expression:

$$\frac{\partial}{\partial \boldsymbol{u}}\mathrm{KL}(P_D||P) = -\left\langle \frac{\partial H(\boldsymbol{\sigma},\boldsymbol{u})}{\partial \boldsymbol{u}} \right\rangle + \frac{1}{D}\sum_{d=1}^{D} \frac{\partial H(\boldsymbol{\sigma}=\boldsymbol{\sigma}^{(d)},\boldsymbol{u})}{\partial \boldsymbol{u}}. \tag{6}$$

Here, the first term is the expectation value associated with $P(\boldsymbol{\sigma}|\boldsymbol{u})$, where $\langle f(\boldsymbol{\sigma})\rangle = \sum_{\boldsymbol{\sigma}} f(\boldsymbol{\sigma})P(\boldsymbol{\sigma}|\boldsymbol{u})$ denotes a thermal equilibrium average of $f(\boldsymbol{\sigma})$. The second term is the arithmetic mean with respect to training data, such as $\frac{1}{D}\sum_{d=1}^{D} \frac{\partial H(\boldsymbol{\sigma}=\boldsymbol{\sigma}^{(d)},\boldsymbol{u})}{\partial J_{12}} = -\sum_{\boldsymbol{\sigma}} \sigma_1\sigma_2 P_D(\boldsymbol{\sigma})$. This does not depend on the distribution $P(\boldsymbol{\sigma}|\boldsymbol{u})$ obtained by the learning algorithm, and can be calculated before the learning algorithm. Therefore we only have to calculate the first term every time we update the parameter $\boldsymbol{u}$. However, a brute force approach to calculate the first term needs $O(2^N)$ steps, which is not tractable. In this paper, we propose a method to use a quantum algorithm to calculate the thermal equilibrium average.

III. VARIATIONAL IMAGINARY TIME SIMULATION

We here review the variational imaginary time simulation, which is compatible with NISQ devices. The Wick-rotated Schrödinger equation describing the imaginary time evolution can be written as [36, 41]:

$$\frac{d\left|\psi(\tau)\right\rangle}{d\tau} = -(\hat{H} - \langle\psi(\tau)|\hat{H}|\psi(\tau)\rangle)\left|\psi(\tau)\right\rangle. \tag{7}$$

Here, this dynamics conserves the norm of $|\psi(\tau)\rangle$ due to the expectation value $\langle\psi(\tau)|\hat{H}|\psi(\tau)\rangle$ in Eq. (7). The state at time τ is expressed as:

$$|\psi(\tau)\rangle = \frac{\exp(-\hat{H}\tau)|\psi(0)\rangle}{\sqrt{\langle\psi(0)|\exp(-2\hat{H}\tau)|\psi(0)\rangle}}, \tag{8}$$

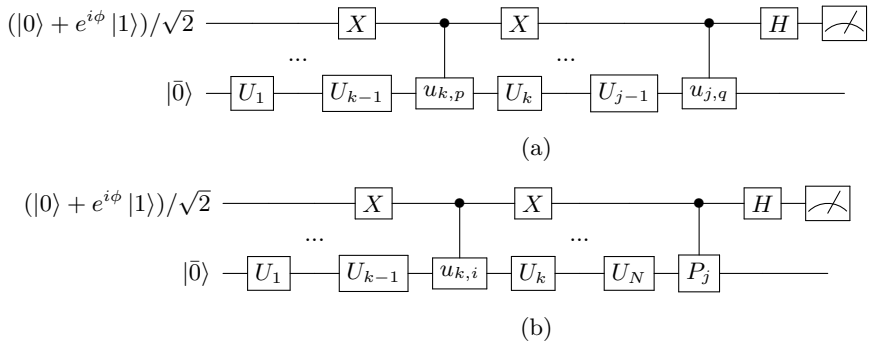

FIG. 1: Quantum circuits for calculating (a) $\Re(e^{i\phi}\langle\bar{0}|\hat{\mathcal{U}}_{k,p}^{\dagger}\hat{\mathcal{U}}_{j,q}|\bar{0}\rangle)$ and (b) $\Re(e^{i\phi}\langle\bar{0}|\hat{\mathcal{U}}_{k,i}^{\dagger}\hat{P}_j\hat{U}|\bar{0}\rangle)$ in order to obtain $M_{k,j}$ in Eq. (18) and C_k in Eq. (19). The upper horizontal line (the lower line) represents the ancillary qubit (the qubits of the system). We prepare the initial state $(|0\rangle + e^{i\phi}|1\rangle)/\sqrt{2}$. X and H denote the Pauli X gate and the Hadamard gate, and U_k ($k = 1, \cdots, N_{\rm p}$) is a parametrized unitary gate constituting the variational circuit. Our ansatz circuit is shown in Fig. 2. The expectation values of Z-measurement on the ancillary qubit give (a) $\Re(e^{i\phi}\langle\bar{0}|\hat{\mathcal{U}}_{k,p}^{\dagger}\hat{\mathcal{U}}_{j,q}|\bar{0}\rangle)$ and (b) $\Re(e^{i\phi}\langle\bar{0}|\hat{\mathcal{U}}_{k,i}^{\dagger}\hat{P}_j\hat{U}|\bar{0}\rangle)$. Therefore by choosing $\phi = 0, \pi/2$, we can obtain both the real and imaginary part of Eqs. (18) and (19).

whose norm is unity.

Instead of directly simulating the non-unitary imaginary time evolution Eq. (7), the variational imaginary time simulation algorithm [41] employs the parametrized wave function $|\varphi(\vec{\theta}(\tau))\rangle$ on a parametrized quantum circuit:

$$\begin{aligned} |\varphi(\vec{\theta}(\tau))\rangle &= \hat{U}(\vec{\theta}(\tau))|\bar{0}\rangle \\ \hat{U}(\vec{\theta}) &= \hat{U}_{N_{\rm p}}(\theta_{N_{\rm p}})\cdots\hat{U}_2(\theta_2)\hat{U}_1(\theta_1), \end{aligned} \tag{9}$$

where $\hat{U}_k(\theta_k)$ ($k = 1, \ldots, N_{\rm p}$) corresponds to a parametrized gate constituting the variational quantum circuit, N_p is a total number of the parameters, θ_k is a real parameter and $|\bar{0}\rangle$ is an initial state of this ansatz which is usually chosen to be equal to $|\psi(0)\rangle$.

Then, Eq. (7) is mapped onto the evolution of the parameters. Here, we use the McLachlan variational principle [42, 43] to derive the time derivative equations for $\vec{\theta}(\tau)$. We minimize the distance between the exact evolution and that of the parametrized trial state as

$$\delta\left\|\left(\frac{\partial}{\partial\tau} + \hat{H} - \langle\varphi(\vec{\theta}(\tau))|\hat{H}|\varphi(\vec{\theta}(\tau))\rangle\right)|\varphi(\vec{\theta}(\tau))\rangle\right\| = 0, \tag{10}$$

where the expectation value $\langle\varphi(\vec{\theta}(\tau))|\hat{H}|\varphi(\vec{\theta}(\tau))\rangle$ plays a role of normalizing the trial state $|\varphi(\vec{\theta}(\tau))\rangle$. This variational principle leads to

$$M\frac{\partial\vec{\theta}(\tau)}{\partial\tau} = \vec{C}, \tag{11}$$

where

$$M_{k,j} = \Re\left(\frac{\partial\langle\varphi(\vec{\theta}(\tau))|}{\partial\theta_k}\frac{\partial|\varphi(\vec{\theta}(\tau))\rangle}{\partial\theta_j}\right), \tag{12}$$

$$C_k = -\Re\left(\langle\varphi(\vec{\theta}(\tau))|\hat{H}\frac{\partial|\varphi(\vec{\theta}(\tau))\rangle}{\partial\theta_k}\right). \tag{13}$$

Here, $\||\varphi\rangle\| \equiv \langle\varphi|\varphi\rangle$, and $\Re(z)$ denotes the real part of z. The size of M and $\vec{C}$ does not depend on the number of the qubits, but depends on the number of the parameters, $N_{\rm p}$. The Euler method with a small parameter $\delta\tau$ can approximately simulate the imaginary time evolution (11), given by

$$\vec{\theta}(\tau+\delta\tau) \simeq \vec{\theta}(\tau) + M^{-1}(\tau)\cdot\vec{C}(\tau)\delta\tau. \tag{14}$$

Note that each element of M and $\vec{C}$ can be efficiently evaluated on quantum circuits shown in Fig. 1. The derivative of parametrized gates can generally be represented as follows:

$$\frac{\partial\hat{U}_k}{\partial\theta_k} = \sum_i a_{k,i}\hat{U}_k\hat{u}_{k,i}. \tag{15}$$

Here, $\hat{u}_{k,i}$ is a unitary operator, with $a_{k,i}$ being a complex coefficient. For example, $d\hat{U}_k/d\theta_k = -i\hat{U}_k(\theta_k)\hat{\sigma}_x$ for a single-qubit rotation $U_k(\theta_k) = \exp(-i\theta_k\hat{\sigma}_x)$, and one can see $a_{k,i} = -i$ and $\hat{u}_{k,i} = \hat{\sigma}_x$. Thus, the derivative of the parametrized state $|\varphi(\vec{\theta}(\tau))\rangle$ is

$$\frac{\partial|\varphi(\vec{\theta}(\tau))\rangle}{\partial\theta_k} = \sum_i a_{k,i}\hat{\mathcal{U}}_{k,i}|\bar{0}\rangle, \tag{16}$$

where

$$\hat{\mathcal{U}}_{k,i} = \hat{U}_{N_{\rm p}}\cdots\hat{U}_{k+1}\hat{U}_k\hat{u}_{k,i}\hat{U}_{k-1}\cdots\hat{U}_1. \tag{17}$$

Using Eq. (17), one can express $M_{k,j}$ as

$$M_{k,j} = \sum_{p,q} \Re \left(a^*_{k,p} a_{j,q} \langle \bar{0} | \hat{\mathcal{U}}^\dagger_{k,p} \hat{\mathcal{U}}_{j,q} | \bar{0} \rangle \right). \tag{18}$$

For $\vec{C}$, assuming that the Hamiltonian $\hat{H}$ can be decomposed as $\hat{H} = \sum_j f_j \hat{P}_j$, where f_j is real and $\hat{P}_j$ is a tensor product of Pauli operators, we obtain

$$C_k = -\sum_{i,j} \Re \left(a_{k,i} f_j \langle \bar{0} | \hat{U}^\dagger(\vec{\theta}) \hat{P}_j \hat{\mathcal{U}}_{k,i} | \bar{0} \rangle \right). \tag{19}$$

It is known that one can calculate $M_{k,j}$ and C_k by using the so-called Hadamard test with controlled unitary gates with the unitary operators of $\hat{\mathcal{U}}^\dagger_{k,p}\hat{\mathcal{U}}_{j,q}$ and $\hat{U}^\dagger(\vec{\theta})\hat{P}_j\hat{\mathcal{U}}_{k,i}$ [44]. However, such controlled unitary gates acting on many qubits require a large depth on the circuit, which may not be suitable for NISQ devices. Instead, to compute the elements of M and $\vec{C}$, we can use a simplified quantum circuit as shown in Fig. 1. Although this circuit contains the controlled unitary gates, only the control of $\hat{u}_{k,p}$ and $\hat{P}_j$ is required, and therefore the depth of this circuit is much shorter than that with the standard Hadamard test using controlled unitary gates $\hat{\mathcal{U}}^\dagger_{k,p}\hat{\mathcal{U}}_{j,q}$ and $\hat{U}^\dagger(\vec{\theta})\hat{P}_j\hat{\mathcal{U}}_{k,i}$. The detailed calculation of this simplified Hadamard test in Fig. 1 is explained in Appendix A.

The necessary number of the circuits to evaluate the matrix M is equivalent to the independent elements of the matrix. When we evaluate the vector C, we need to decompose the Hamiltonian and it needs more circuits proportional to the number of the terms in the Hamiltonian. These elements are determined by the total number of the parameters $N_{\rm p}$. Note that the matrix M is a Hermitian $N_{\rm p} \times N_{\rm p}$ matrix and the vector $\vec{C}$ is a $N_{\rm p}$ dimensional vector, and so the total number of the independent components is $O(N^2_{\rm p})$.

In Appendix B, we discuss the effect of the shot noise in obtaining the elements of the matrix M and the vector $\vec{C}$ with an accuracy ε_s. We show an important relation between the KLD and the fidelity (or the trace distance) for sufficiently small ε_s, and we give a derivation of the condition for the required number of measurements N_m [34, 45, 46]. In order to obtain the expectation values within the deviation ε_s, the required number of measurements is $N_m = O(T^2/\varepsilon_s^2)$ on the time interval $0 \le \tau \le T$. If this condition is satisfied by taking a large number of measurements, we obtain both the KLD and the trace distance within the accuracy ε_s, except for a constant factor.

The previous result [47] showed that the indirect measurements via the ancillary qubits in Fig. 1 can be replaced by the direct measurements on the qubits of the system. This method mainly has two advantages. First, one can reduce the depth of the quantum circuit. Second, it allows one to replace the measurement on the ancillary qubits with the direct measurements of the system, which remove the use of the long-range gates between the ancillary qubit and the other qubits. Both these two advantages are favorable for current NISQ devices. However, since the realized dynamics by using the method in [47] is the same as the dynamics realized by Fig. 1 under an assumption of no noise, we adopt the quantum circuit Fig. 1 in this paper.

IV. BM LEARNING WITH VARIATIONAL IMAGINARY TIME SIMULATION

In this section, we describe our proposal of a quantum algorithm to calculate the thermal equilibrium average in Eq. (6). In particular, we adopt the variational imaginary time simulation described in the previous section. For this purpose, we replace σ_i in the Hamiltonian (2) with the Pauli matrix $\hat{\sigma}_{z,i}$. We define $|0\rangle$ and $|1\rangle$ as the eigenstates of $\hat{\sigma}_z$ given by $\hat{\sigma}_z |0\rangle = |0\rangle$ and $\hat{\sigma}_z |1\rangle = -|1\rangle$.

We thus introduce the quantum Hamiltonian as follows:

$$\hat{H}(\boldsymbol{u}) = -\sum_{i<j} J_{ij} \hat{\sigma}_{z,i} \hat{\sigma}_{z,j} - \sum_{i=1}^{N} h_i \hat{\sigma}_{z,i} \tag{20}$$

By using the variational imaginary time simulation, we can prepare a quantum state:

$$|\psi_{\boldsymbol{u}}\rangle = \sqrt{\frac{2^N}{Z(\boldsymbol{u})}} e^{-\hat{H}(\boldsymbol{u})/2} |+ + \cdots +\rangle, \tag{21}$$

using the initial state $|+ + \cdots +\rangle = |+\rangle \otimes |+\rangle \otimes \cdots \otimes |+\rangle$. Since $\partial \hat{H}(\boldsymbol{u})/\partial \boldsymbol{u}$ consists only of the Pauli matrix $\hat{\sigma}_z$, only diagonal elements of the density matrix $|\psi_{\boldsymbol{u}}\rangle \langle \psi_{\boldsymbol{u}}|$ are involved in calculating $\left\langle \psi_{\boldsymbol{u}} \middle| \partial \hat{H}(\boldsymbol{u})/\partial \boldsymbol{u} \middle| \psi_{\boldsymbol{u}} \right\rangle$. Also, every diagonal element of $|\psi_{\boldsymbol{u}}\rangle \langle \psi_{\boldsymbol{u}}|$ is the same as that in the Gibbs state $e^{-\hat{H}(\boldsymbol{u})}/Z(\boldsymbol{u}) (= \sum_{\sigma} P(\boldsymbol{\sigma}|\boldsymbol{u}) |\boldsymbol{\sigma}\rangle \langle \boldsymbol{\sigma}|)$. So, once Eq. (21) is prepared, one can calculate the thermal equilibrium average in the right hand side of Eq. (6) as follows:

$$\left\langle \frac{\partial H(\boldsymbol{\sigma}, \boldsymbol{u})}{\partial \boldsymbol{u}} \right\rangle = \left\langle \psi_{\boldsymbol{u}} \middle| \frac{\partial \hat{H}(\boldsymbol{u})}{\partial \boldsymbol{u}} \middle| \psi_{\boldsymbol{u}} \right\rangle. \tag{22}$$

Note that both Eq. (8) and Eq. (21) are normalized, and therefore when the initial state $|\psi(0)\rangle = |+ + \cdots +\rangle$ evolves from $\tau = 0$ to $\tau = 1/2$ following Eq. (7), two states are equal. This evolution is required for every update of the parameter $\boldsymbol{u}$ in the learning process.

The above scheme can be also applied to the learning process of the RBM [3–5]. Since the RBM has connections only between a hidden unit and a visible unit, we can construct the Hamiltonian for the RBM by simply removing the interactions within the hidden unit and within the visible unit from Eq. (20). The most time-consuming part of the learning process of the RBM is to calculate the thermal equilibrium average as well.

The average is given by the same equation (22). Consequently, we can use our scheme to update the parameters in the Hamiltonian for the RBM.

Algorithm 1 Optimization of learning parameter

Initialize $\boldsymbol{u}$ with the normal distribution
for $s = 1, 2, \ldots, N_{\rm step}$ **do**
  $\tau \leftarrow 0$
  $\vec{\theta} \leftarrow \vec{0}$
  repeat
    Calculate matrix M using Eq. (18)
    Calculate vector $\vec{C}$ using Eq. (19)
    $\vec{\theta} \leftarrow \vec{\theta} + M^{-1}\vec{C}\,\delta\tau$
    $\tau \leftarrow \tau + \delta\tau$
  until $\tau = 1/2$
  $|\psi_{\boldsymbol{u}}\rangle \leftarrow |\varphi(\vec{\theta})\rangle$ using Eq. (9)
  $$\frac{\partial \mathrm{KL}}{\partial \boldsymbol{u}} \leftarrow -\left\langle \psi_{\boldsymbol{u}} \left| \frac{\partial \hat{H}(\boldsymbol{u}')}{\partial \boldsymbol{u}'}\right|_{\boldsymbol{u}'=\boldsymbol{u}} \right| \psi_{\boldsymbol{u}} \right\rangle + \frac{1}{D}\sum_{d=1}^{D} \left.\frac{\partial H(\boldsymbol{\sigma}=\boldsymbol{\sigma}^{(i)}, \boldsymbol{u}')}{\partial \boldsymbol{u}'}\right|_{\boldsymbol{u}'=\boldsymbol{u}}$$
  $\boldsymbol{u} \leftarrow \boldsymbol{u} - \eta \dfrac{\partial \mathrm{KL}}{\partial \boldsymbol{u}}$
end for
return $\boldsymbol{u}$

V. RESULTS

A. A first check of our scheme

1. *Datasets from true Boltzmann distribution and our algorithm*

In this section, we show the performance of our scheme by numerical simulations. For this purpose, we need to generate a training data set. First, we randomly generate L sets of parameters $\boldsymbol{u}_l^*$ $(l = 1, 2, \cdots, L)$. We use a normal distribution with a mean of 0 and a variance of 1, $\mathcal{N}(0, 1)$, to generate each component of $\boldsymbol{u}_l^*$ for all l. We call $\boldsymbol{u}_l^*$ true parameters. Second, for a given $\boldsymbol{u}_l^*$, we generate training data $\boldsymbol{\sigma}^{(i,l)} \in \{-1, 1\}^N$ $(i = 1, \cdots, D)$ based on a probability distribution of $P(\boldsymbol{\sigma}|\boldsymbol{u}_l^*)$, which provides us with L groups of the training data. Third, we calculate $P_D^{(l)}(\boldsymbol{\sigma})$ in Eq. (3) by using each group of the training data. In our numerical simulations, we adopt $N = 4$, $L = 30$, and $D = 1000$.

Our simulation method is summarized in Algorithm 1. Here, we choose the number of iteration steps $N_{\rm step} = 100$, a fixed step size of the imaginary time evolution $\delta\tau = 0.1$, and the learning rate $\eta = 0.1$ in the numerical simulations.

Based on the training data, we run our algorithm as follows. First, by using one of the groups of the training data generated by $P(\boldsymbol{\sigma}|\boldsymbol{u}_1^*)$, we perform numerical simulations of our scheme where we choose our ansatz circuit for the variational imaginary time simulation as Fig. 2, and obtain the learning parameters $\boldsymbol{u}_{\rm opt}^{(l)}$ after the optimization by using the gradient method Eq. (5) from $s = 0$ to $N_{\rm step}$. For an initial guess of $\boldsymbol{u}$ for the simulation ($\boldsymbol{u}[0]$ in Eq. (5)), we use the normal distribution with $\mathcal{N}(0, 1)$ at each time. Second, we then obtain L sets of learning parameters after repeating the same numerical simulations but by using each group of the training data sampled from $P(\boldsymbol{\sigma}|\boldsymbol{u}_2^*)$, $P(\boldsymbol{\sigma}|\boldsymbol{u}_3^*)$, ..., and $P(\boldsymbol{\sigma}|\boldsymbol{u}_L^*)$. We repeat this process L times. Note that we set the number of each sampling to 10^4 in order to obtain the expectation value in Eq. (22) and to perform the variational imaginary time simulation using the quantum circuits in Fig. 1.

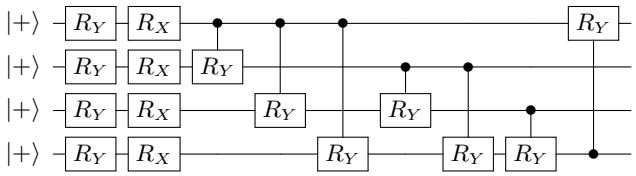

FIG. 2: Our ansatz circuit for the variational imaginary time simulation with 4 qubits. We prepare the initial state $|++++\rangle$. R_X and R_Y represent rotational gates $R_X(\theta) = e^{-i\theta\hat{\sigma}_x/2}$ and $R_Y(\theta') = e^{-i\theta'\hat{\sigma}_y/2}$ with different parameter angles such as θ and θ'.

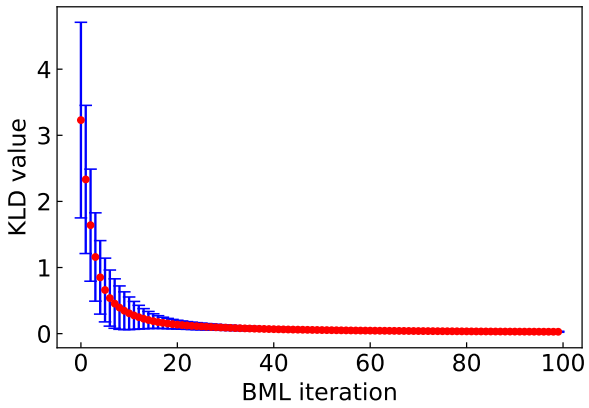

FIG. 3: The KLD between the true distribution and the distribution obtained from the imaginary time evolution at each iteration step s. We set the total iteration step $N_{\rm step} = 100$ and plot the KLD from $s = 0$ to $s = N_{\rm step}$. We take an average of KLD over all groups of training data at each step, and show each mean value and standard deviation in this plot. The KLD converges to zero, and thus our scheme based on the imaginary time evolution can reproduce the training data.

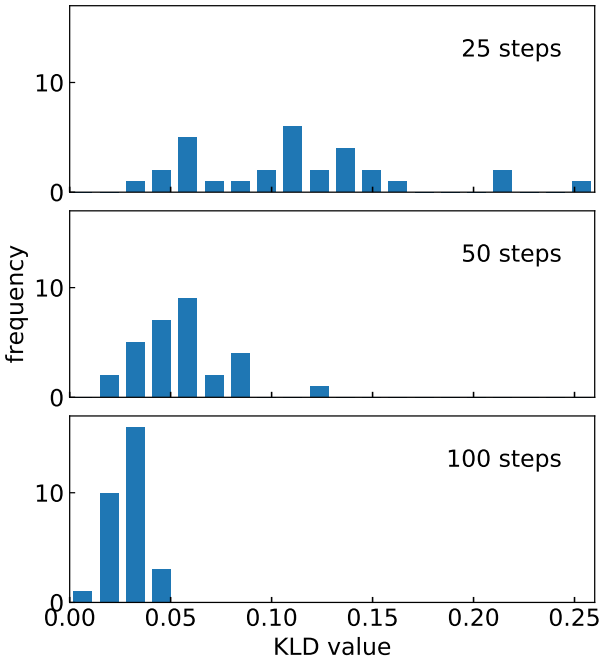

FIG. 4: The histogram of the values of the KLD at $s =$ 25, 50, and 100 iteration step, respectively. We set the total iteration step $N_{\rm step} = 100$. The range of the KLD values is from 0 to 0.26, and we divide it into 20 bins. We plot the histogram as the number of the counts in each bin.

2. *Numerical simulations*

To evaluate the performance of our scheme, we consider the KLD between the true distribution $P(\boldsymbol{\sigma}|\boldsymbol{u}_l^*)$ and estimated distribution $P(\boldsymbol{\sigma}|\boldsymbol{u}^{(l)}[s])$ in Eq. (1) optimized by the variational imaginary time simulation, where $\boldsymbol{u}^{(l)}[s]$ denotes a learning parameter at the s-th step. Figure 3 shows the average and standard deviation of such KLD at s-th step over all the true parameter sets $\boldsymbol{u}_l^*$ $(l = 1, 2, \ldots, L)$. These results show that all the average and the standard deviation of the KLD converge to zero. Our scheme can thus successfully reproduce the probability distribution that has been generated the training data, and show the high performance of our proposal as BM learning.

Furthermore, we investigate how the convergence speed of the KLD depends on the given true parameters $\boldsymbol{u}_l^*$. Figure 4 shows the histogram of the values of the KLD at $s =$25, 50, and 100 iteration step, respectively. At 25th iteration step, the values of the KLD strongly depends on the true parameters $\boldsymbol{u}_l^*$. However, as we increase the iteration step, the standard deviation becomes smaller, and most of the KLD becomes close to zero at 100th iteration step.

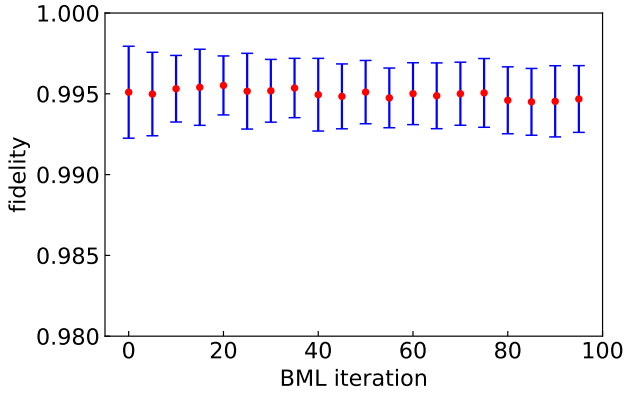

FIG. 5: The fidelity between $|\psi_{\boldsymbol{u}}\rangle$ in Eq. (21) and $|\varphi(\vec{\theta}(\tau = 1/2))\rangle$ in Eq. (9) with respect to the iteration step s. The red points and the blue bars represent the average values and the standard deviation values of the fidelity over all groups of training data at each iteration step s. The fidelity is around or more than 0.995 at every iteration step. These results show that our ansatz circuit is suitable to generate $|\psi_{\boldsymbol{u}}\rangle$ in Eq. (21) with a reasonably good approximation.

In addition, we examine whether the variational imaginary time simulation actually provides us with $|\psi_{\boldsymbol{u}}\rangle$ in Eq. (21). It is worth mentioning that the state obtained by the variational imaginary time simulation $|\varphi(\vec{\theta}(\tau = 1/2))\rangle$ in Eq. (9) could be different from $|\psi_{\boldsymbol{u}}\rangle$ in Eq. (21), because its performance strongly depends on the ansatz of the quantum circuit. To investigate how close the state $|\varphi(\vec{\theta}(\tau = 1/2))\rangle$ is to the state $|\psi_{\boldsymbol{u}}\rangle$, we calculate the fidelity:

$$F(|\psi_{\boldsymbol{u}}\rangle, |\varphi(\vec{\theta}(\tau = 1/2))\rangle) = |\langle\psi_{\boldsymbol{u}}|\varphi(\vec{\theta}(\tau = 1/2))\rangle|^2. \tag{23}$$

The fidelity is more than 0.995 at every iteration step (Fig. 5). These results show the good agreement between $|\psi_{\boldsymbol{u}}\rangle$ and $|\varphi(\vec{\theta}(\tau = 1/2))\rangle$. From these results, we can exclude the possibility that our machine learning algorithm may work well accidentally, because we only use the expectation values in our scheme and not use the full information of the obtained state.

B. Depth number dependence

In this subsection, we evaluate the required number of gates in order to achieve a sufficiently high fidelity between $|\psi_{\boldsymbol{u}}\rangle$ and $|\varphi(\vec{\theta}(\tau = 1/2))\rangle$. Similar to Sec. V A, we randomly generate L sets of true parameters $\boldsymbol{u}_l^*$ $(l = 1, 2, \cdots, L)$, and let each component of $\boldsymbol{u}_l^*$ follow the normal distribution with $\mathcal{N}(0, 1)$.

In Fig. 6, we describe our ansatz circuit, where $N_{\rm depth}$ denotes the number of the repetitions. $N_{\rm depth}$ represents

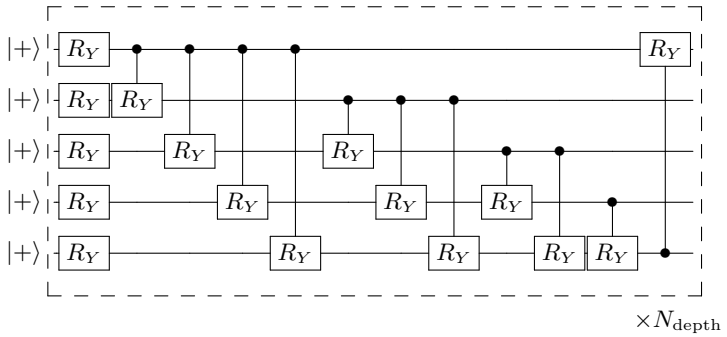

FIG. 6: Our ansatz circuit for the case of $N = 5$ qubits. This is almost the same as Fig. 2 but repetitions. $N_{\rm depth}$ denotes the number of repetitions.

the power of the expression for our ansatz circuit. We perform the imaginary time evolution with this circuit, and investigate how the fidelity increases with $N_{\rm depth}$. We adopt parameters of $N = 5, 6$, $L = 100$, $N_{\rm depth} = 1, 2, 3$, $\delta\tau = 0.1$, and 10^4 times measurements to obtain each matrix (vector) element of M ($\vec{C}$).

Figures 7(a) and 7(b) show the average fidelity against $N_{\rm depth}$ at $N = 5$ and 6 for a given set of the true parameters $\boldsymbol{u}_l^*$ ($l = 1, 2, \cdots, L$), respectively. The definition of the fidelity is given by Eq. (23). Importantly, the values of the fidelity is around 0.985(0.98) for $N = 5$ ($N = 6$), even with $N_{\rm depth} = 1$ and do not have a clear dependence on $N_{\rm depth}$. These results show that our ansatz circuit is suitable for our purpose even with $N_{\rm depth} = 1$.

C. More general dataset

In order to check the validity of our scheme, we investigate whether our scheme can fit a data set generated from a bars and stripes (BAS) [48], which cannot be exactly described by the Boltzmann distribution. The way to generate the 4-dimensional BAS dataset is shown below: Firstly, we randomly choose one from the following matrices $\left\{\begin{pmatrix} -1 & -1 \\ -1 & -1 \end{pmatrix}, \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}\right\}$ with a probability of $1/4$. Secondly, we convert the matrix into a 4-dimensional vector where a matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is stochastically converted to either $\begin{pmatrix} a & c & b & d \end{pmatrix}^T$ or $\begin{pmatrix} a & b & c & d \end{pmatrix}^T$ with a probability of $1/2$ where a, b, c, d denote 1 or -1. Finally, we can get the dataset to repeat this procedure as many times as needed. The distribution generated by the BAS has a feature that specific data such as $\begin{pmatrix} 1 & 1 & 1 & -1 \end{pmatrix}^T$ does not appear at all. On the other hand, in the distribution of the BM, every data could appear with non-zero probability. In this sense, it is not straightforward whether our scheme based on the BM could reproduce the dataset of BAS or not.

To simulate our scheme, we generate training data $\boldsymbol{\sigma}^{(i)} \in \{-1, 1\}^N$ ($i = 1, \cdots, D$) from BAS. We also set up L sets of initial values of learning parameters $\boldsymbol{u}^{(l)}[0]$ ($l = 1, \cdots, L$) from the standard multivariate normal distribution to show that the results do not depend on specific initial values. We adopt parameters of $N = 4, L = 30, D = 1000, N_{\rm step} = 100, \eta = 0.1$, and $\delta\tau = 0.1$. Also, we use the quantum circuit in Fig. 2, which is the same as that described in Section V.

To evaluate the performance of our scheme toward the BAS dataset, we consider the KLD between the BAS distribution and the estimated distribution $P(\boldsymbol{\sigma} \mid \boldsymbol{u}^{(l)}[s])$ in Eq. (1) in the same way as Section V. Figure 8 shows the average and standard deviation of the KLD at s-th step over the learning parameters $\boldsymbol{u}^{(l)}[s]$ ($l = 1, 2, \cdots, L$). The value of the KLD is around or less than 0.153 at 100th iteration step. So our simulation shows reasonable fittings with the given data.

Moreover, we compare the estimated distribution at 100th iteration step with the BAS distribution, and examine how the result varies with different initial parameters $\boldsymbol{u}^{(l)}[0]$. Figure 9 shows the BAS distribution and the average and standard deviation of the estimated distributions at 100th step over the training parameters $\boldsymbol{u}^{(l)}[N_{\rm step}]$ ($l = 1, 2, \cdots, L$). From this graph, we can confirm that the estimated distribution is capable of reproducing the BAS distribution, which is compatible with the results of the KLD in Fig. 8. As described above, there remain finite values of the obtained distribution although the corresponding values are zero for the BAS distribution in Fig. 9. This deviation stems from the BM learning itself, not our scheme utilizing NISQ devices to obtain the required thermal average. The learning performance can be improved by using the RBM learning with more ancillary qubits required in our scheme.

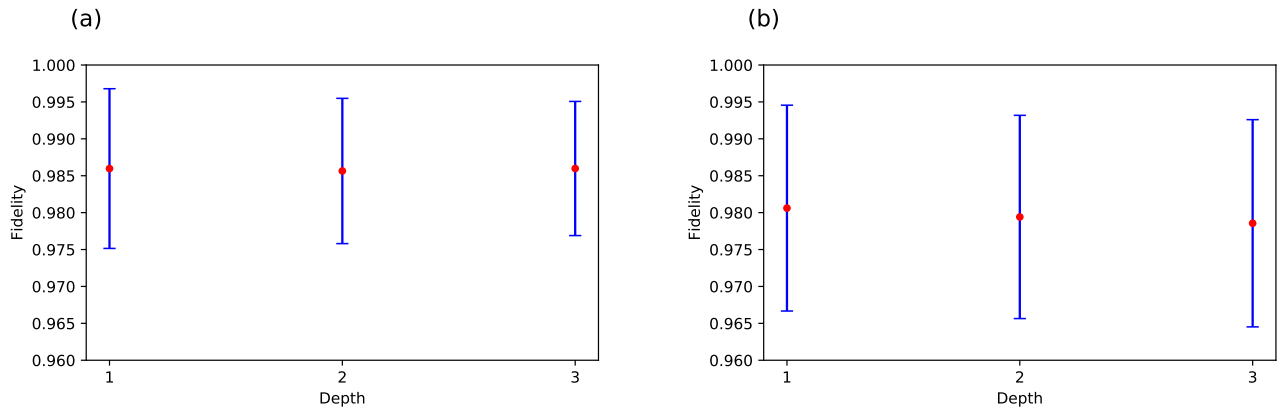

FIG. 7: Plot of the average fidelity in Eq. (23) against $N_{\rm depth}$ at $N = 5$ qubits in Fig. 7(a) and $N = 6$ qubits in Fig. 7(b). We use the ansatz circuit in Fig. 6 in order to calculate $|\varphi(\vec{\theta}(\tau = 1/2))\rangle$ using the variational imaginary time simulation. The red points (the blue bars) denote the average values (the standard deviation) of the fidelity over all the true parameters $\boldsymbol{u}_l^*$ $(1 \leq l \leq L)$; we choose $L = 100$.

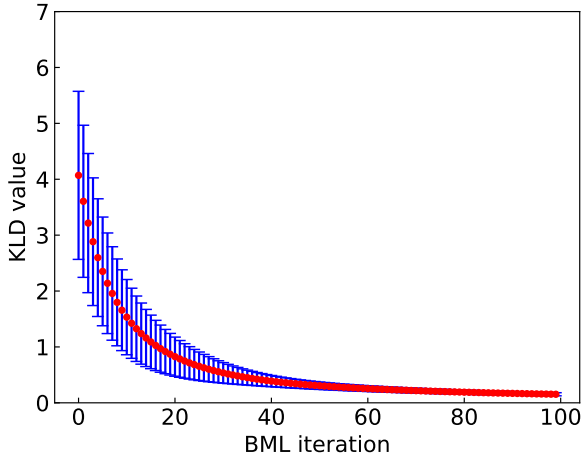

FIG. 8: The KLD between the BAS distribution and the distribution obtained from the imaginary time evolution at each iteration step. We take an average of KLD over all groups of training data at each step, and show each mean value and standard deviation in this plot.

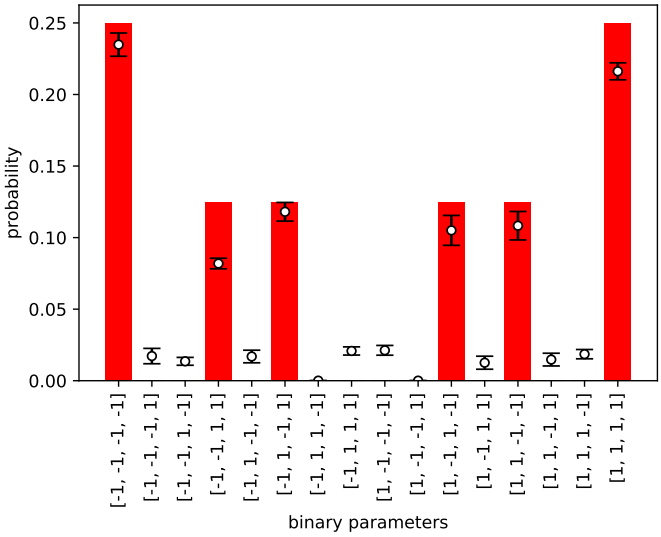

FIG. 9: Comparison between the BAS distribution and the distribution obtained from the imaginary time evolution. Especially, for the imaginary time evolution, we plot the average and standard deviation of the distribution at the 100th iteration step. We set the total iteration step $N_{\rm step} = 100$.

VI. COMPARISON WITH OTHER SCHEMES OF BOLTZMANN MACHINE LEARNING

Let us compare our scheme with other approaches for BM learning with classical and quantum algorithms.

The standard approach on classical computers is to approximate the expectation values with sample averages using the Markov-chain Monte Carlo (MCMC) method [49]. Calculation of the averages using the MCMC method typically requires many samplings, leading to long computational time for the learning process. However, it has been shown that the averages calculated from a few steps of the Markov chain are sufficient for the learning of RBM in practice [19, 20]. This method is called contrastive divergence (CD) learning, and faster than the standard MCMC method. However, a drawback of the CD learning is that the approximation introduces a bias which can prevent the model from converging to the optimal parameters [50]. To reduce the bias, parallel tempering Monte Carlo is applied for the learning of RBM [51, 52]. In particular, the authors of Ref. [51] has revealed that the method is as efficient as the CD learning in terms of the computational time.

In addition to classical approaches, there are other previous studies on the preparation of Eq. (21) by using quantum annealing [53, 54]. They construct the Hamiltonian whose ground state of which is equal to Eq. (21). However, such a Hamiltonian includes many-body (more than three-body) interactions and is difficult to realize experimentally. Moreover, with the NISQ algorithms, there is another method to create a thermal equilibrium state by minimizing the free energy function with a fixed temperature [55–60]. Nevertheless, the free energy function is determined by the energy expectation value and the von Neumann entropy, and it is a non-trivial problem to calculate the von Neumann entropy on a relatively shallow quantum circuit.

An interesting question for future work is whether our scheme is more useful due to the quantum properties or not, compared with the classical scheme of BM learning algorithms such as Markov-chain Monte Carlo samplings [49]. In Appendix C, we give a detailed comparison with the CD learning for the case of a few qubits $N = 4$. We show that our scheme exhibits rapid convergence. However, the efficiency of the method may depend on various factors such as a data set and the number of qubits. Therefore, in order to answer the above question, we need a careful benchmark of the performance of our scheme by using an actual NISQ device, because it is difficult to obtain an analytical proof of quantum speedup as with many other NISQ algorithms. Actually, it is an open and central question whether a VQA has quantum advantage or not [32]. However, without such an assessment about the quantum speedup, our work presented in this paper could still contribute to the society for the following two reasons. First, most of the implementations of the BM learning are heuristic, and thus they may not always be able to find the suitable solution if we focus on just a specific implementation. Therefore it is better if we have more options to perform the BM learning for searching the suitable solutions. Second, a quantum algorithm based on gate operations is compatible with a protocol called blind quantum computation [61]. Here, a client can safely delegate a quantum algorithm to a server who has a gate-type quantum computer, and the server cannot steal any information about what the client performs with the server machine where the security is based on fundamental physics law such as no-signaling principle [62]. Since the training data could contain personal information that should not be leaked to the third party, such a secure BM learning could have a significant importance in the society. Therefore, in principle, our scheme can be implemented with blind quantum computation, while information-theoretically secure BM learning is not known in classical implementations. It is worth mentioning that such a secure quantum computer in the NISQ era was recently discussed in Ref. [63].

VII. CONCLUSION

In conclusion, we propose a scheme to implement the BM learning based on the variational imaginary time simulation with NISQ devices. Our approach contrasts to the previous approach of Ref [36], which prepares thermal equilibrium states for quantum Hamiltonians by using two copies of quantum states. A key feature of our approach is to focus on classical cases when the Hamiltonian for the BM has only diagonal terms in the computational basis. We use a pure state whose distribution mimics the thermal equilibrium distribution. The potential advantage of our scheme is that we do not need two copies of quantum states unlike the scheme of [36], which reduces the number of qubits. Our results show potential for an efficient use of the NISQ device for the BM learning.

ACKNOWLEDGMENTS

We thank Dr. Muneki Yasuda for useful comments on BM. We also thank Takashi Imoto and Atsuki Yoshinaga for valuable discussions. This work was supported by Leading Initiative for Excellent Young Researchers MEXT Japan and JST presto (Grant No. JPMJPR1919) Japan. This work was partly supported by MEXT Q-LEAP (JPMXS0118068682), and JST ERATO (JPMJER1601). This paper is partly based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO), Japan. S.W. was supported by Nanotech CUPAL, National Institute of Advanced Industrial Science and Technology (AIST).

We performed the numerical calculations in Figs. 3 and 4 by using Qiskit, a open-source library for numerical simulations of quantum algorithms provided by Ref. [64].

While this manuscript was being written, an independent article [65] proposed quantum Boltzmann machine, which prepares a Gibbs state by leveraging thermofield-double technique combined with the variational imaginary time simulation (although this method for the Gibbs state preparation was already proposed in a published work [36]).

Appendix A: Details of the Hadamard test

Here, we explain the details of obtaining $M_{k,j}$ by implementing the circuit in Fig. 1. Especially, for a given set of k, p, j, q, we show the detailed calculation of the term $\Re\left(a^*_{k,p}a_{j,q}\,\langle\bar{0}|\,\hat{\mathcal{U}}^\dagger_{k,p}\hat{\mathcal{U}}_{j,q}\,|\bar{0}\rangle\right)$, which corresponds to the right side of Eq. (18). Let us define two real variables B and ϕ by $a^*_{k,p}a_{j,q} = Be^{i\phi}$, and also $\hat{T}_{i,j} = \hat{U}_j\hat{U}_{j-1}\cdots\hat{U}_{i+1}\hat{U}_i$ $(i \le j)$, which can simplify the form as $\hat{\mathcal{U}}_{k,i} = \hat{T}_{k,N_p}\hat{u}_{k,i}\hat{T}_{1,k-1}$. As shown in

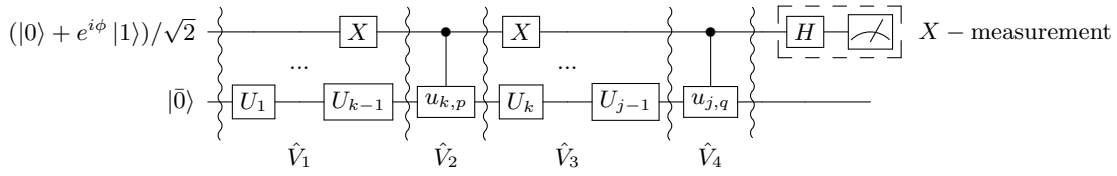

FIG. 10: Division of the unitary operation of the quantum circuit in Fig. 1 (a) by $\hat{V}_1, \hat{V}_2, \hat{V}_3, \hat{V}_4$ in order to explain how to obtain the expectation values $\Re\Big(a^*_{k,p} a_{j,q} \langle\bar{0}| \hat{\mathcal{U}}^\dagger_{k,p} \hat{\mathcal{U}}_{j,q} |\bar{0}\rangle \Big)$.

Fig. 10, we separately consider the several gate operations (which we classify as $\hat{V}_1, \hat{V}_2, \hat{V}_3, \hat{V}_4$) of Fig. 1, the X-measurement that consists of the Hadamard gate, and the Z-measurement on the ancillary qubit. Firstly, let us explain the changes in the quantum state via the implementation of the several gates.

$$
\begin{aligned}
&\frac{1}{\sqrt{2}}(|0\rangle + e^{i\phi}|1\rangle) \otimes |\bar{0}\rangle \\
\xrightarrow{\hat{V}_1} &\frac{1}{\sqrt{2}}(|1\rangle + e^{i\phi}|0\rangle) \otimes \hat{T}_{1,k-1}|\bar{0}\rangle \\
\xrightarrow{\hat{V}_2} &\frac{1}{\sqrt{2}}\Big(|1\rangle \otimes \hat{u}_{k,p}\hat{T}_{1,k-1}|\bar{0}\rangle + e^{i\phi}|0\rangle \otimes \hat{T}_{1,k-1}|\bar{0}\rangle \Big) \\
\xrightarrow{\hat{V}_3} &\frac{1}{\sqrt{2}}\Big(|0\rangle \otimes \hat{T}_{k,j-1}\hat{u}_{k,p}\hat{T}_{1,k-1}|\bar{0}\rangle + e^{i\phi}|1\rangle \otimes \hat{T}_{1,j-1}|\bar{0}\rangle \Big) \\
\xrightarrow{\hat{V}_4} &\frac{1}{\sqrt{2}}\Big(|0\rangle \otimes \hat{T}_{k,j-1}\hat{u}_{k,p}\hat{T}_{1,k-1}|\bar{0}\rangle + e^{i\phi}|1\rangle \otimes \hat{u}_{j,q}\hat{T}_{1,j-1}|\bar{0}\rangle \Big) \\
&= |\Phi_{k,j,p,q}\rangle
\end{aligned}
\tag{A1}
$$

where $|\Phi_{k,j,p,q}\rangle$ denotes $\hat{V}_4\hat{V}_3\hat{V}_2\hat{V}_1\Big(\frac{1}{\sqrt{2}}(|0\rangle + e^{i\phi}|1\rangle) \otimes |\bar{0}\rangle\Big)$. Secondly, let us explain how the X-measurement on the ancillary qubit provides the value of $\Re\Big(e^{i\phi}\langle\bar{0}|\hat{\mathcal{U}}^\dagger_{k,p}\hat{\mathcal{U}}_{j,q}|\bar{0}\rangle \Big)$ as follows:

$$
\begin{aligned}
&\langle\Phi_{k,j,p,q}| X\otimes I |\Phi_{k,j,p,q}\rangle \\
=&\frac{1}{2}\Big(\langle 1|\otimes\langle\bar{0}|\hat{T}^\dagger_{1,k-1}\hat{u}^\dagger_{k,p}\hat{T}^\dagger_{k,j-1} + e^{-i\phi}\langle 0|\otimes\langle\bar{0}|\hat{T}^\dagger_{1,j-1}\hat{u}^\dagger_{j,q} \Big) \\
&\Big(|0\rangle\otimes\hat{T}_{k,j-1}\hat{u}_{k,p}\hat{T}_{1,k-1}|\bar{0}\rangle + e^{i\phi}|1\rangle\otimes\hat{u}_{j,q}\hat{T}_{1,j-1}|\bar{0}\rangle \Big) \\
=&\frac{1}{2}\Big(e^{-i\phi}\langle\bar{0}|\hat{T}^\dagger_{1,j-1}\hat{u}^\dagger_{j,q}\hat{T}_{k,j-1}\hat{u}_{k,p}\hat{T}_{1,k-1}|\bar{0}\rangle \\
&+ e^{i\phi}\langle\bar{0}|\hat{T}^\dagger_{1,k-1}\hat{u}^\dagger_{k,p}\hat{T}^\dagger_{k,j-1}\hat{u}_{j,q}\hat{T}_{1,j-1}|\bar{0}\rangle \Big) \\
=&\frac{1}{2}\Big(e^{-i\phi}\langle\bar{0}|\hat{\mathcal{U}}^\dagger_{j,q}\hat{\mathcal{U}}_{k,p}|\bar{0}\rangle + e^{i\phi}\langle\bar{0}|\hat{\mathcal{U}}^\dagger_{k,p}\hat{\mathcal{U}}_{j,q}|\bar{0}\rangle \Big) \\
=&\Re\Big(e^{i\phi}\langle\bar{0}|\hat{\mathcal{U}}^\dagger_{k,p}\hat{\mathcal{U}}_{j,q}|\bar{0}\rangle \Big),
\end{aligned}
\tag{A2}
$$

where we apply $\hat{T}^\dagger_{j,N_\mathrm{p}}\hat{T}_{j,N_\mathrm{p}} = I$ in the third line. By multiplying B by the expectation values of the measurement results, we obtain $\Re\Big(Be^{i\phi}\langle\bar{0}|\hat{\mathcal{U}}^\dagger_{k,p}\hat{\mathcal{U}}_{j,q}|\bar{0}\rangle \Big)$. Finally, by repeating the first and second step for every p and q, we obtain $M_{k,j}$ by summing up these results as

$$
M_{k,j} = \sum_{p,q} \Re\Big(Be^{i\phi}\langle\bar{0}|\hat{\mathcal{U}}^\dagger_{k,p}\hat{\mathcal{U}}_{j,q}|\bar{0}\rangle \Big). \tag{A3}
$$

We can also obtain C_k in the same way.

Appendix B: The effect of shot noise

In this section, we consider the effect of noise when we compute elements of M and $\vec{C}$ by utilizing quantum circuits shown in Fig. 1. It is worth mentioning that the effect of the decoherence and gate imperfections can be suppressed by quantum error mitigation technique [31, 66–75]. Also, for simplicity, we assume that we choose the perfect ansatz to simulate the imaginary time evolution:

$$
|\psi_{\boldsymbol{u}}\rangle = |\varphi(\vec{\theta}(\tau = 1/2))\rangle \tag{B1}
$$

$$
\vec{\theta}(\tau = 1/2) = \int_0^{1/2} d\tau M_0^{-1}\vec{C}_0 \tag{B2}
$$

where M_0 and $\vec{C}_0$ denote the ideal matrix of M and $\vec{C}$ without shot noise. In this case, we should mainly consider the shot noise that comes from the finite number of the repetitions of the measurements. So we investigate how many measurements are required per circuit to obtain the accurate expectation values.

The shot noise prevents us from performing a precise estimation of the M and $\vec{C}$, and this leads a deviation of our ansatz quantum state (generated from the quantum circuit) from the ideal one. We define the state $|\psi_\varepsilon\rangle$ with the shot noise as

$$
|\psi_\varepsilon\rangle = \sqrt{1-\varepsilon}\,|\psi_{\boldsymbol{u}}\rangle + \sqrt{\varepsilon}\,|\delta\psi_{\boldsymbol{u}}\rangle\,, \tag{B3}
$$

where $|\psi_{\boldsymbol{u}}\rangle$ is an ideal state without noise defined in Eq. (21) and $|\delta\psi_{\boldsymbol{u}}\rangle$ is a normalized state, $\| |\delta\psi_{\boldsymbol{u}}\rangle \| = 1$ and $\langle\psi_{\boldsymbol{u}}|\delta\psi_{\boldsymbol{u}}\rangle = 0$, which denotes the deviation of the ideal state due to the shot noise. Here, we assume ε is sufficiently small. Since we consider the pure states that are described by the wavefunctions $|\psi_{\boldsymbol{u}}\rangle$ and $|\psi_\varepsilon\rangle$, the

following relation holds:

$$D(|\psi_{\boldsymbol{u}}\rangle, |\psi_\varepsilon\rangle) = \sqrt{1 - F(|\psi_{\boldsymbol{u}}\rangle, |\psi_\varepsilon\rangle)} \tag{B4}$$

$$F(|\psi_{\boldsymbol{u}}\rangle, |\psi_\varepsilon\rangle) = |\langle\psi_{\boldsymbol{u}}|\psi_\varepsilon\rangle|^2 = 1 - \varepsilon, \tag{B5}$$

where $D(|\psi_{\boldsymbol{u}}\rangle, |\psi_\varepsilon\rangle)$ denotes the trace distance [44]. We can give the bound of KLD between the true distribution function $P(\boldsymbol{\sigma}|\boldsymbol{u})$ in Eq. (1) and the distribution function obtained by the variational imaginary time simulation:

$$P_\psi(\boldsymbol{\sigma}|\boldsymbol{u}) = \frac{1}{2^N}|\langle\boldsymbol{\sigma}|\psi_\varepsilon\rangle|^2 \tag{B6}$$

$$= P(\boldsymbol{\sigma}|\boldsymbol{u}) + 2\sqrt{\varepsilon P(\boldsymbol{\sigma}|\boldsymbol{u})}\Re(\langle\boldsymbol{\sigma}|\delta\psi_{\boldsymbol{u}}\rangle) + O(\varepsilon), \tag{B7}$$

where $|\boldsymbol{\sigma}\rangle$ denotes the computational basis corresponding to $\boldsymbol{\sigma}$ in $P(\boldsymbol{\sigma}|\boldsymbol{u})$. We can calculate the KLD as

$$\begin{aligned}\mathrm{KL}(P||P_\psi) &\simeq -2\sum_{\boldsymbol{\sigma}}\sqrt{\varepsilon P(\boldsymbol{\sigma}|\boldsymbol{u})}\Re(\langle\boldsymbol{\sigma}|\delta\psi_{\boldsymbol{u}}\rangle),\\ &\le 2\sqrt{\varepsilon}\sqrt{\sum_{\boldsymbol{\sigma}}|\Re(\langle\boldsymbol{\sigma}|\delta\psi_{\boldsymbol{u}}\rangle)|^2}\\ &\le 2\sqrt{\varepsilon}\sqrt{\sum_{\boldsymbol{\sigma}}|\langle\boldsymbol{\sigma}|\delta\psi_{\boldsymbol{u}}\rangle|^2}\\ &= 2\sqrt{\varepsilon}.\end{aligned} \tag{B8}$$

Therefore, we obtain

$$\mathrm{KL}(P||P_\psi) \le 2\sqrt{1 - F(|\psi_{\boldsymbol{u}}\rangle, |\psi_\varepsilon\rangle)} = 2D(|\psi_{\boldsymbol{u}}\rangle, |\psi_\varepsilon\rangle). \tag{B9}$$

Using this inequality, we will show the condition for the required number of measurements [34, 45, 46].

The solution of Eq. (11) can be rewritten as

$$\frac{\partial\vec{\theta}(\tau)}{\partial\tau} = (M_0 + \delta M)^{-1}(\vec{C}_0 + \delta\vec{C}), \tag{B10}$$

where δM and $\delta\vec{C}$ denote the deviation originating from shot noise. If we assume that δM is sufficiently small compared with M_0, Eq. (B10) can be rewritten as

$$\frac{\partial\vec{\theta}(\tau)}{\partial\tau} \simeq M_0^{-1}\vec{C}_0 + M_0^{-1}\delta\vec{C} - M_0^{-2}\delta M\vec{C}_0, \tag{B11}$$

where we employ the linear approximation $(M_0 + \delta M)^{-1} \simeq M_0^{-1} - M_0^{-2}\delta M$, neglecting the terms of higher order in δM and $\delta\vec{C}$. In Eq. (B11), the first term of the RHS denotes the noise free solution and the second and the third terms represent the effect of the shot noise, and then we can evaluate the noise terms as

$$\left|\delta\dot{\vec{\theta}}\right| = \left|\frac{\partial\vec{\theta}(\tau)}{\partial\tau} - M_0^{-1}\vec{C}_0\right| \tag{B12}$$

$$\simeq \left|M_0^{-1}\delta\vec{C} - M_0^{-2}\delta M\vec{C}_0\right|, \tag{B13}$$

$$\le \left\|M_0^{-1}\right\||\delta\vec{C}| + \left\|M_0^{-1}\right\|^2\|\delta M\||\vec{C}_0|, \tag{B14}$$

where $\|\cdots\|$ denotes the Frobenius norm. From the central limit theorem, the noise terms behave as

$$\|\delta M\| \simeq \frac{\Delta_M}{\sqrt{N_m}}, \tag{B15}$$

$$|\delta\vec{C}| \simeq \frac{\Delta_{\vec{C}}}{\sqrt{N_m}}, \tag{B16}$$

where N_m denotes the number of measurements per quantum circuit. Here, Δ_M and $\Delta_{\vec{C}}$ can be evaluated as

$$\Delta_M \simeq 2\sqrt{\sum_{j,k}\left(\sum_{p,q}|a^*_{k,p}a_{j,q}|\right)^2}, \tag{B17}$$

$$\Delta_{\vec{C}} \simeq 2\sqrt{\sum_{j}\left(\sum_{p,q}|a^*_{k,p}f_j|\right)^2}. \tag{B18}$$

Using these quantities, we can rewrite Eq. (B14) as

$$\left|\delta\dot{\vec{\theta}}\right| \le \frac{\left\|M_0^{-1}\right\|\Delta_{\vec{C}} + \left\|M_0^{-1}\right\|^2|\vec{C}_0|\Delta_M}{\sqrt{N_m}} \tag{B19}$$

$$= \frac{\Delta}{\sqrt{N_m}}, \tag{B20}$$

where we define Δ by

$$\Delta = \left\|M_0^{-1}\right\|\Delta_{\vec{C}} + \left\|M_0^{-1}\right\|^2|\vec{C}_0|\Delta_M. \tag{B21}$$

By using the trace distance, the previous study [34] expressed the degree of the error between the ideal wavefunction and the wavefunction obtained from the variational algorithms when we update from $\vec{\theta}(\tau)$ to $\vec{\theta}(\tau+\delta\tau)$ as follows:

$$\begin{aligned}&D(|\varphi^{(0)}(\vec{\theta}_0(\tau+\delta\tau))\rangle, |\varphi(\vec{\theta}(\tau+\delta\tau))\rangle)\\ &= \sqrt{\left[\delta\dot{\vec{\theta}}^T A(\tau)\delta\dot{\vec{\theta}}\right](\delta\tau)^2 + O((\delta\tau)^3)},\end{aligned} \tag{B22}$$

where $|\varphi^{(0)}(\vec{\theta}_0(\tau+\delta\tau))\rangle$ denotes the ideal wavefunction at time $\tau+\delta\tau$. Here, the matrix $A(\tau)$ is defined by

$$\begin{aligned}&[A(\tau)]_{i,j}\\ &= M_{i,j} - \frac{\partial\langle\varphi(\vec{\theta}(\tau))|}{\partial\theta_j}|\varphi(\vec{\theta}(\tau))\rangle\langle\varphi(\vec{\theta}(\tau))|\frac{\partial|\varphi(\vec{\theta}(\tau))\rangle}{\partial\theta_i}.\end{aligned} \tag{B23}$$

From these calculations, the total error D_I from $\tau = 0$ to $\tau = T$ is given by

$$D_I = D(|\varphi^{(0)}(\vec{\theta}_0(T))\rangle, |\varphi(\vec{\theta}(T))\rangle) \tag{B24}$$

$$\le \sum_n D(|\varphi^{(0)}(\vec{\theta}_0(n\delta\tau))\rangle, |\varphi(\vec{\theta}(n\delta\tau))\rangle) \tag{B25}$$

$$\lesssim \sqrt{\|A\|_{\max}}\|\delta\dot{\vec{\theta}}\|_{\max}T \tag{B26}$$

$$= \sqrt{\|A\|_{\max}}\frac{\Delta_{\max}T}{\sqrt{N_m}}, \tag{B27}$$

where $\|\cdots\|_{\max}$ is the maximum value on the time interval $0 \le \tau \le T$. In order to obtain the expectation values within the accuracy ε_s, we need

$$\varepsilon_s \ge \sqrt{\|A\|_{\max}} \frac{\Delta_{\max} T}{\sqrt{N_m}}, \tag{B28}$$

and we can obtain the condition on the required number of measurements as

$$N_m \ge \|A\|_{\max} \Delta_{\max}^2 T^2 / \varepsilon_s^2. \tag{B29}$$

If this condition is satisfied by taking a large number of measurements, we obtain both the KLD and the trace distance within the accuracy ε_s except for a constant factor, according to Eq. (B9).

Appendix C: Comparison with contrastive divergence learning

In this Appendix, we compare our scheme with contrastive divergence (CD) learning, which is a standard technique to train the RBM [19–21]. Although the CD learning is mainly used for the RBM, the learning algorithm can be also applied to the fully visible BM [20], such as our model in the main text. Our calculation is restricted for the case of a few qubits because no actual NISQ device with relatively large qubits is currently available, as described in Section VI.

We apply the k-step CD learning (CD-k) for calculating the thermal average in Eq. (6) to update the parameters in the BM. To calculate the thermal average by using CD-k, we generate D samples from the Markov chain with length k and approximate the thermal average with the sample average. An initial distribution of the Markov chain is the uniform distribution of the training data set $\{\boldsymbol{\sigma}^{(d)}\}_{d=1}^{D}$. For simplicity, we assume that each data in the training data set appears once in the sampling from the initial distribution when generating the D samples. In the Markov process, we use the heat-bath algorithm to update the state, where the parameters are sequentially updated from σ_1 to σ_N.

Figure 11 shows the average of the KLD between the true distribution $P(\boldsymbol{\sigma}|\boldsymbol{u}_l^*)$ and the estimated probabilities (our scheme, the CD-1 learning, and the CD-50 learning) at the s-th step over all the true parameter sets $\boldsymbol{u}_l^*$ ($l = 1, 2, \ldots, L$). We apply the common parameters of $N = 4, L = 30, D = 10^4, N_{\text{step}} = 100$, and $\eta = 0.1$. These results show that while the values of the KLD in our scheme converge more rapidly than the ones in the CD-1 learning, the performance of our scheme is comparable with that of the CD-50. This is because both our scheme and the CD-50 learning achieve thermal average values with high accuracy compared with the CD-1 learning, and therefore both of the learning parameters are almost the same at each step s. The rapid convergence compared with the CD-1 learning probably depends on problems such as a data set and the number of qubits. Therefore,

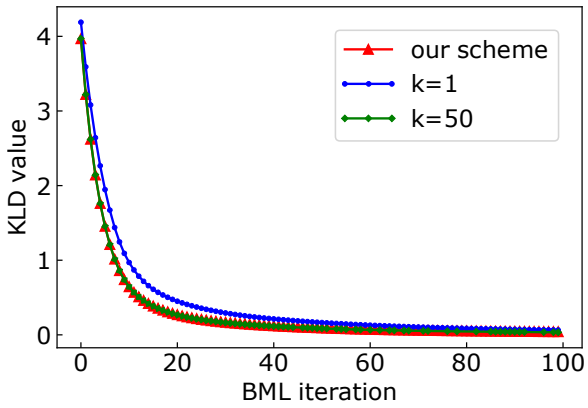

FIG. 11: The KLD between the true distribution and the estimated distributions which are obtained from three different methods; our scheme, the CD-1 learning ($k = 1$), and the CD-50 learning ($k = 50$) at each iteration step. We take averages of the KLD over all group of training data at each step.

in order to prove quantum advantage of our scheme over the classical schemes, we need a careful benchmark of the performance using an actual NISQ device with relatively larger qubits. We leave this for a future work.

[1] S. E. Fahlman, G. E. Hinton, and T. J. Sejnowski, in *National Conference on Artificial Intelligence, AAAI* (1983).

[2] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, Cognitive science **9**, 147 (1985).

[3] P. Smolensky, *Information processing in dynamical systems: Foundations of harmony theory*, Tech. Rep. (Colorado Univ at Boulder Dept of Computer Science, 1986).

[4] V. Nair and G. E. Hinton, in *Proceedings of the 27th international conference on machine learning (ICML-10)* (2010) pp. 807–814.

[5] G. E. Hinton, in *Neural networks: Tricks of the trade* (Springer, 2012) pp. 599–619.

[6] N. Le Roux and Y. Bengio, Neural computation **20**, 1631 (2008).

[7] G. E. Hinton and R. R. Salakhutdinov, science **313**, 504 (2006).

[8] R. Salakhutdinov, A. Mnih, and G. Hinton, in *Proceedings of the 24th international conference on Machine learning* (2007) pp. 791–798.

[9] H. Larochelle and Y. Bengio, in *Proceedings of the 25th international conference on Machine learning* (2008) pp.

536–543.
[10] R. G. Melko, G. Carleo, J. Carrasquilla, and J. I. Cirac, Nature Physics **15**, 887 (2019).
[11] Z.-A. Jia, B. Yi, R. Zhai, Y.-C. Wu, G.-C. Guo, and G.-P. Guo, Advanced Quantum Technologies **2**, 1800077 (2019).
[12] C.-y. Hsieh, Q. Sun, S. Zhang, and C. K. Lee, arXiv preprint arXiv:1912.02988 (2019).
[13] G. E. Hinton and R. R. Salakhutdinov, in *Advances in neural information processing systems* (2009) pp. 1607–1614.
[14] A. Coates, A. Ng, and H. Lee, in *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (2011) pp. 215–223.
[15] G. E. Dahl, D. Yu, L. Deng, and A. Acero, IEEE Transactions on audio, speech, and language processing **20**, 30 (2011).
[16] G. E. Hinton, S. Osindero, and Y.-W. Teh, Neural computation **18**, 1527 (2006).
[17] R. Salakhutdinov and G. Hinton, in *Artificial intelligence and statistics* (PMLR, 2009) pp. 448–455.
[18] R. Salakhutdinov and G. Hinton, Neural computation **24**, 1967 (2012).
[19] G. E. Hinton, Neural computation **14**, 1771 (2002).
[20] M. A. Carreira-Perpinan and G. E. Hinton, in *Aistats*, Vol. 10 (Citeseer, 2005) pp. 33–40.
[21] Y. Bengio and O. Delalleau, Neural computation **21**, 1601 (2009).
[22] M. Gabrié, E. W. Tramel, and F. Krzakala, in *Advances in neural information processing systems* (2015) pp. 640–648.
[23] B. Marlin, K. Swersky, B. Chen, and N. Freitas, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (2010) pp. 509–516.
[24] M. Yasuda, S. Kataoka, Y. Waizumi, and K. Tanaka, in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)* (IEEE, 2012) pp. 2234–2237.
[25] M. Yasuda, T. Sakurai, and K. Tanaka, Nonlinear Theory and Its Applications, IEICE **2**, 153 (2011).
[26] P. Rebentrost, M. Mohseni, and S. Lloyd, Physical review letters **113**, 130503 (2014).
[27] M. Schuld, I. Sinayskiy, and F. Petruccione, Physical Review A **94**, 022342 (2016).
[28] N. Wiebe, D. Braun, and S. Lloyd, Physical review letters **109**, 050505 (2012).
[29] S. Lloyd, M. Mohseni, and P. Rebentrost, Nature Physics **10**, 631 (2014).
[30] J. Preskill, Quantum **2**, 79 (2018).
[31] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, Journal of the Physical Society of Japan **90**, 032001 (2021).
[32] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, *et al.*, arXiv preprint arXiv:2012.09265 (2020).
[33] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. Obrien, Nature communications **5**, 4213 (2014).
[34] Y. Li and S. C. Benjamin, Physical Review X **7**, 021050 (2017).
[35] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, New Journal of Physics **18**, 023023 (2016).
[36] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, Quantum **3**, 191 (2019).
[37] S. Endo, J. Sun, Y. Li, S. C. Benjamin, and X. Yuan, Phys. Rev. Lett. **125**, 010501 (2020).
[38] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Physical Review A **98**, 032309 (2018).
[39] M. Benedetti, D. Garcia-Pintos, O. Perdomo, V. Leyton-Ortega, Y. Nam, and A. Perdomo-Ortiz, npj Quantum Information **5**, 1 (2019).
[40] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, Physical Review X **8**, 021050 (2018).
[41] S. McArdle, T. Jones, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, npj Quantum Information **5**, 1 (2019).
[42] A. McLachlan, Molecular Physics **8**, 39 (1964).
[43] J. Broeckhove, L. Lathouwers, E. Kesteloot, and P. Van Leuven, Chemical physics letters **149**, 547 (1988).
[44] M. A. Nielsen and I. Chuang, "Quantum computation and quantum information," (2002).
[45] S. Endo, *Hybrid quantum-classical algorithms and error mitigation*, Ph.D. thesis, University of Oxford (2019).
[46] S. Endo, I. Kurata, and Y. O. Nakagawa, Phys. Rev. Research **2**, 033281 (2020).
[47] K. Mitarai and K. Fujii, Physical Review Research **1**, 013006 (2019).
[48] R. Y. Li, T. Albash, and D. A. Lidar, Quantum Science and Technology **5**, 045010 (2020).
[49] D. P. Landau and K. Binder, *A guide to Monte Carlo simulations in statistical physics* (Cambridge University Press, 2014).
[50] I. Sutskever and T. Tieleman, in *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (JMLR Workshop and Conference Proceedings, 2010) pp. 789–795.
[51] K. Cho, T. Raiko, and A. Ilin, in *The 2010 international joint conference on neural networks (ijcnn)* (IEEE, 2010) pp. 1–8.
[52] G. Desjardins, A. Courville, Y. Bengio, P. Vincent, and O. Delalleau, in *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (JMLR Workshop and Conference Proceedings, 2010) pp. 145–152.
[53] R. D. Somma, C. D. Batista, and G. Ortiz, Physical review letters **99**, 030603 (2007).
[54] M. Yamamoto, M. Ohzeki, and K. Tanaka, Journal of the Physical Society of Japan **89**, 025002 (2020).
[55] J. Wu and T. H. Hsieh, Physical Review Letters **123**, 220502 (2019).
[56] D. Zhu, S. Johri, N. Linke, K. Landsman, N. Nguyen, C. Alderete, A. Matsuura, T. Hsieh, and C. Monroe, arXiv preprint arXiv:1906.02699 (2019).
[57] G. Verdon, J. Marks, S. Nanda, S. Leichenauer, and J. Hidary, arXiv preprint arXiv:1910.02071 (2019).
[58] A. N. Chowdhury, G. H. Low, and N. Wiebe, arXiv preprint arXiv:2002.00055 (2020).
[59] Y. Wang, G. Li, and X. Wang, arXiv preprint arXiv:2005.08797 (2020).
[60] G. Verdon, M. Broughton, and J. Biamonte, arXiv preprint arXiv:1712.05304 (2017).
[61] A. Broadbent, J. Fitzsimons, and E. Kashefi, in *2009 50th Annual IEEE Symposium on Foundations of Computer Science* (IEEE, 2009) pp. 517–526.
[62] T. Morimae and K. Fujii, Physical Review A **87**, 050301 (2013).
[63] E. Kashefi, D. Leichtle, L. Music, and H. Ollivier, arXiv preprint arXiv:2011.10005 (2020).
[64] H. A. et al., "Qiskit: An open-source framework for quantum computing," (2019).

[65] C. Zoufal, A. Lucchi, and S. Woerner, arXiv preprint arXiv:2006.06004 (2020).
[66] K. Temme, S. Bravyi, and J. M. Gambetta, Physical review letters **119**, 180509 (2017).
[67] S. Endo, S. C. Benjamin, and Y. Li, Physical Review X **8**, 031027 (2018).
[68] X. Bonet-Monroig, R. Sagastizabal, M. Singh, and T. O'Brien, Physical Review A **98**, 062339 (2018).
[69] S. Endo, Q. Zhao, Y. Li, S. Benjamin, and X. Yuan, Physical Review A **99**, 012334 (2019).
[70] S. McArdle, X. Yuan, and S. Benjamin, Physical review letters **122**, 180501 (2019).
[71] C. Song, J. Cui, H. Wang, J. Hao, H. Feng, and Y. Li, Science advances **5**, eaaw5686 (2019).
[72] R. Sagastizabal, X. Bonet-Monroig, M. Singh, M. A. Rol, C. Bultink, X. Fu, C. Price, V. Ostroukh, N. Muthusubramanian, A. Bruno, *et al.*, Physical Review A **100**, 010302 (2019).
[73] A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, Nature **567**, 491 (2019).
[74] S. Zhang, Y. Lu, K. Zhang, W. Chen, Y. Li, J.-N. Zhang, and K. Kim, Nature communications **11**, 1 (2020).
[75] H. Hakoshima, Y. Matsuzaki, and S. Endo, arXiv preprint arXiv:2009.12759 (2020).