Quantum Boltzmann machine algorithm with dimension-expanded equivalent Hamiltonian

Tailong Xiao,[1] Dongyun Bai,[1] Jianping Fan,[2] and Guihua Zeng[1,*]

[1] *State Key Laboratory of Advanced Optical Communication Systems and Networks and Center of Quantum Information Sensing and Processing, Shanghai Jiao Tong University, Shanghai 200240, China*

[2] *Department of Computer Science, University of North Carolina-Charlotte, Charlotte, North Carolina 28223, USA*

(Received 26 August 2019; revised manuscript received 5 November 2019; accepted 13 February 2020; published 5 March 2020)

Quantum machine learning explores the devising and implementing of quantum software to enable the speedup for a classical machine learning counterpart, which has attracted extensive attention. The quantum Boltzmann machine is a promising quantum machine learning model and can be executed on near-term quantum devices to demonstrate quantum supremacy. We investigate an efficient algorithm for training a quantum Boltzmann machine based on a continuous-time quantum cluster Monte Carlo method realized with a dimension-expanded equivalent Hamiltonian. The proposed algorithm is capable of efficiently training various types of quantum Boltzmann machines, including a two-dimensional random transverse Ising Hamiltonian and other stochastic κ-local ($\kappa \geqslant 2$) Hamiltonians. We further apply a quantum mean-field method to speed up the training process through discarding the quantum fluctuations of visible-hidden interactions. The experiment results show that our algorithm-trained quantum Boltzmann machine is more efficient on a small and large data set in terms of network convergence and time complexity.

DOI: 10.1103/PhysRevA.101.032304

I. INTRODUCTION

Machine learning has achieved tremendous success in various fields, including computer vision [1,2], collaborative filtering [3], quantum many-body physics [4,5], quantum metrology [6], and quantum tomography [7]. With the advancement of the integration between machine learning and quantum mechanics, numerous algorithms have been proposed to efficiently solve the existing problems originating from quantum physics or computer science, which has spawned a new research subject called quantum machine learning (QML) [8,9].

QML algorithms alleviate the dilemma of the inefficiency of classical machine learning algorithms in the era of big data. Numerous QML algorithms have been proposed, such as a quantum support vector machine [10], a quantum-data-fitting algorithm [11], and a quantum generative adversarial network [12,13]. These QML algorithms achieve exponential speedup compared with their classical counterparts and are executed on a universal quantum computer (UQC). However, it is a challenge to build UQC in the short term because of the fast decoherence of the qubits system and unstable nonlinear quantum gates. Thus, quantities of quantum schemes are proposed aiming to present the quantum supremacy in noisy intermediate-scale quantum (NISQ) devices [14] such as Boson sampling machines [15] and quantum Boltzmann machines (QBMs) [16].

The QBM is a quantum generalization of the classical Boltzmann machine (CBM) by replacing the energy function of a CBM with a quantum Hamiltonian. Training QBMs can be divided into two steps: (a) improving the positive phase, referring to an increase of the probability of training data, and (b) improving the negative phase, referring to a decrease of the probability of samples generated by QBMs [17]. The problem of the positive phase can be mitigated by a *bound-based* method [16]. In Refs. [18,19], a *state-based* training method is proposed based on quantum relative entropy. The *state-based* training requires quantum empirical data distribution. Thus, the classical data needs to be transformed into superposition states by calling quantum random access memory such as to be impractical in NISQ devices. Currently, a new quantum training method via generalized Gibbs ensembles overcomes the difficulties of projection training. The quantum average over the projector by means of generalized Gibbs ensembles can be efficiently sampled [20]. The negative phase computation is generally NP-hard for conventional exact algorithms [21]. In Ref. [22], a sampling algorithm for negative phase computation based on the eigenstate thermalization hypothesis (ETH) is proposed to train different types of QBMs. However, there is no known analytical derivation of the ETH for general interacting systems [23]. Moreover, the training algorithm that requires more hyperparameter optimizations and lacks flexibility renders it uneasily implemented.

In this regard, we attempt to obtain a general efficient training algorithm for various QBMs. First, we consider quantum-Merlin-Arthur–complete (QMA-complete) Hamiltonians such as ZZXX and ZX local models [24] and derive the respective equation for their positive phase learning based on a *bound-based* approximation. Second, motivated by a highly impressive algorithm for an efficient training quantum variational autoencoder [25] in which QBM is regarded as a component, we make use of a path integral (PI) formulation to present an efficient continuous-time quantum cluster Monte

[*] ghzeng@sjtu.edu.cn

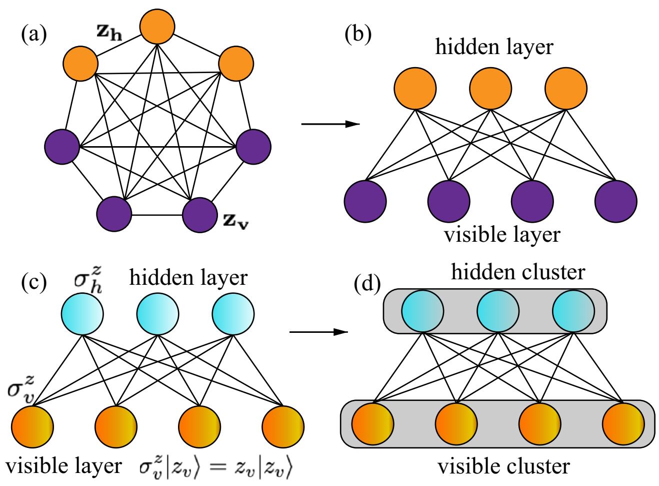


FIG. 1. Illustration of classical and quantum Boltzmann machine: (a) fully connected Boltzmann machine, (b) restricted Boltzmann machine, (c) quantum restricted Boltzmann machine, and (d) quantum mean-field approximation when regarding visible or hidden layer as a cluster.

Carlo (CT-QCMC) algorithm to learn the negative phase without systematic error. In order to train QBMs in a large-scale data set, we further apply the quantum mean-field (QMF) [26] approximation to reduce the complexity of the computation such as to accelerate the training process. Finally, to demonstrate the learning ability of various QBMs, the experiments on a small toy data set are conducted where the negative phase is updated through CT-QCMC and QMF approximation algorithms. In addition, we validate the reconstruction performance of QBMs on a large-scale data set via CT-QCMC and its QMF approximation. Through a large number of experiments, we demonstrate that both algorithms are able to efficiently train QBMs. In particular, QMF-algorithm-trained QBMs show a competitive performance compared to CT-QCMC-algorithm-trained QBMs.

The remainder of this paper is organized as follows. Section II introduces the basics of CBM and QBMs from a novel perspective. The efficient training algorithm is presented in detail in Sec. III. Section IV provides the simulation results of the proposed algorithm. Conclusions are drawn in Sec. V.

II. PRELIMINARIES

A. Classical Boltzmann machine

CBM is commonly viewed as a generative unsupervised machine learning model which involves a probability distribution from an original data set and using it to make inferences about never before seen data.

1. Fully connected Boltzmann machine

A fully connected Boltzmann machine consists of a visible layer and a hidden layer as Fig. 1(a) shows. Neurons on the visible layer and hidden layer are called visible and hidden neurons, respectively. Each neuron is connected with rest of neurons within the network. Suppose a CBM over an N-dimensional binary random vector $\mathbf{z} \in \{-1, 1\}^N$ is defined. Since CBM is an energy-based model, the joint probability distribution using an Ising energy function given by

$$\mathcal{P}(\mathbf{z}) = \frac{\exp\left(-E(\mathbf{z})\right)}{Z}, \tag{1}$$

where $E(\mathbf{z})$ is the energy function and Z is the partition function calculated by $Z = \sum_{\mathbf{z}} \exp\left(-E(\mathbf{z})\right)$. The energy function of the fully connected Boltzmann distribution is given by

$$E_f(\mathbf{z}) = -\sum_i \sum_j J_{ij} z_i z_j - \sum_i b_i z_i, \tag{2}$$

where J_{ij} is the coupling strength between z_i and z_j, b_i is the bias parameter for each z_i, representing the auxiliary field for each spin. The dimensionless parameters b_i and J_{ij} are tuned during the network training process [3,17]. Note that neurons can be divided into visible and hidden neurons, denoted by $\mathbf{z} = (\mathbf{z_v}, \mathbf{z_h})$. In thermal equilibrium, the probability of observing a visible state $\mathbf{v}$ is given by the Boltzmann distribution summed over the hidden neurons given by

$$\mathcal{P}_f(\mathbf{z_v}) = Z_f^{-1} \sum_{\mathbf{z_h}} \exp\left(-E_f(\mathbf{z})\right) = Z_{fv}^{-1} \exp\left(-\mathcal{F}_f(\mathbf{z_v})\right), \tag{3}$$

where we have defined the free energy and the corresponding partition function of the visible neurons given by

$$\mathcal{F}_f(\mathbf{z_v}) = -\ln \sum_{\mathbf{z_h}} \exp\left(-E_f(\mathbf{z})\right), \tag{4}$$

$$Z_{fv} = \sum_{\mathbf{z_v}} \exp\left(-\mathcal{F}_f(\mathbf{z_v})\right). \tag{5}$$

The goal of training CBM is to learn the optimal coupling strength J_{vh} and bias parameter b_i such as to minimize the Kullback-Leibeler (KL) divergence between the model distribution $\mathcal{P}_f(\mathbf{z_v})$ and the empirical data distribution $\mathcal{Q}(\mathbf{z_v})$. To train CBM, a simpler method is to maximize the average log-likelihood function (negative cross-entropy), which is equivalent to minimize the KL divergence defined by

$$\mathcal{L}_f = \sum_{\mathbf{z_v}} \mathcal{Q}(\mathbf{z_v}) \ln \mathcal{P}_f(\mathbf{z_v}). \tag{6}$$

The minimization can be achieved using the gradient ascent method. In each iteration, the parameter set $\theta = \{J_{ij}, b_i\}$ is adjusted by a small step in the direction of the gradient, namely, $\delta\theta = \alpha \partial_\theta \mathcal{L}_f$, where α is called the learning rate. The derivative of the average log-likelihood function by using Eq. (6) is given by

$$\partial_\theta \mathcal{L}_f = \sum_{\mathbf{z_v}} \mathcal{Q}(\mathbf{z_v}) \partial_\theta \mathcal{F}_f(\mathbf{z_v}) - \sum_{\mathbf{z}'_\mathbf{v}} \mathcal{P}(\mathbf{z}'_\mathbf{v}) \partial_\theta \mathcal{F}_f(\mathbf{z}'_\mathbf{v}), \tag{7}$$

where $\mathbf{z}'_\mathbf{v}$ denotes full configurations of visible neurons in the Boltzmann distribution. The first term called positive phase increases the probability of training data (by reducing the corresponding free energy), while the second term called negative phase decreases the probability of samples generated by the model. In reality, the free energy requires summing over full configurations of hidden neurons. The positive phase can be calculated efficiently by obtaining the expectation of the free energy over data distribution $\mathcal{Q}(\mathbf{z_v})$. However, the negative phase requires calculating the expectation of the free energy over a model distribution $\mathcal{P}(\mathbf{z}'_\mathbf{v})$ in which full visible configurations are required. This problem can be alleviated by using sampling methods such as Monte Carlo Markov chains (MCMCs) to estimate the expectations of the free energy of the model [27]. However, the high cost of burning in the Markov chains renders this procedure computationally infeasible. Therefore, practically speaking, the fully connected Boltzmann machine is difficult to train.

2. Restricted Boltzmann machine

A restricted Boltzmann machine (RBM) is developed by restricting the connections of neurons only between the visible and hidden layers, as Fig. 1(b) shows. Thus, the energy function of the RBM can be written as

$$E_r(\mathbf{z}) = -\sum_v \sum_h J_{vh} z_v z_h - \sum_v a_v z_v - \sum_h b_h z_h. \tag{8}$$

The corresponding Boltzmann distribution $\mathcal{P}_r(\mathbf{z})$ can be calculated by Eq. (1). The free energy of the visible neurons is given by

$$\mathcal{F}_r(\mathbf{z_v}) = -\sum_v a_v z_v - \sum_h \ln\left[\cosh\left(\sum_v J_{vh} z_v + b_h\right)\right]. \tag{9}$$

Thus, the gradient with respect to J_{vh}, a_v, b_h can be calculated by Eq. (7) with replaced free energy $\mathcal{F}_r(\mathbf{z_v})$ given by

$$\partial_{J_{vh}} \mathcal{L} = \sum_{\mathbf{z_v}} \mathcal{Q}(\mathbf{z_v}) z_v \tanh\left(\sum_v J_{vh} z_v + b_h\right) - \mathbb{E}_{\mathcal{P}_r(\mathbf{z})}[z_v z_h], \tag{10a}$$

$$\partial_{a_v} \mathcal{L} = \sum_{\mathbf{z_v}} \mathcal{Q}(\mathbf{z_v}) z_v - \mathbb{E}_{\mathcal{P}_r(\mathbf{z})}[z_v], \tag{10b}$$

$$\partial_{b_h} \mathcal{L} = \sum_{\mathbf{z_v}} \mathcal{Q}(\mathbf{z_v}) \tanh\left(\sum_v J_{vh} z_v + b_h\right) - \mathbb{E}_{\mathcal{P}_r(\mathbf{z})}[z_h]. \tag{10c}$$

Furthermore, although the negative phase involves evaluating the partition function, it can be estimated by Gibbs sampling from the conditional distribution [28]. In addition, the contrastive divergence (CD) or persistent CD (PCD) method developed for RBM is more efficient in practical training [29].

B. Quantum Boltzmann machine

The QBM is a highly promising quantum machine learning model exploited based on quantum simulation. Given the quantum Hamiltonian of QBM denoted with $H_{\rm QBM}$ and the inverse temperature β (here assume Boltzmann factor $k_B = 1$), the quantum thermal Boltzmann distribution for visible qubits [16] is given by

$$\mathcal{P}_Q(\mathbf{z_v}) = \frac{\mathrm{Tr}[\Xi_{\mathbf{v}} \exp(-\beta H_{\rm QBM})]}{\mathrm{Tr}[\exp(-\beta H_{\rm QBM})]}, \tag{11}$$

where $\Xi_{\mathbf{v}}$ is a projector of the visible qubits $|\mathbf{z_v}\rangle$ in QBM onto classical visible neurons $\mathbf{z_v}$. Equation (11) is presented by partial tracing out the hidden qubits in the quantum Boltzmann distribution, in which the density matrix of QBM is defined by $\rho = Z_Q^{-1}\exp(-\beta H_{\rm QBM})$, with $Z_Q = \mathrm{Tr}[\exp(-\beta H_{\rm QBM})]$ to ensure $\mathrm{Tr}[\rho] = 1$. The free energy for visible qubits $|\mathbf{z_v}\rangle$ is given by

$$\mathcal{F}_Q = -\beta^{-1} \ln \mathrm{Tr}[\Xi_{\mathbf{v}} \exp(-\beta H_{\rm QBM})]. \tag{12}$$

To train QBM, one can maximize the average log-likelihood function of quantum thermal distribution similar to Eq. (6):

$$\mathcal{L}_Q = \sum_{\mathbf{z_v}} \mathcal{Q}(\mathbf{z_v}) \ln\left(\frac{\exp(-\beta\mathcal{F}_Q)}{Z_{Qv}}\right). \tag{13}$$

We adopt the *bound-based* projection training method [16,25]. More specifically, the lower bound of $\mathcal{L}_Q$ denoted by $\widehat{\mathcal{L}}_Q$ is maximized by taking its gradient

$$\frac{\partial\widehat{\mathcal{L}}_Q}{\beta\partial\theta} = \sum_{\mathbf{z_v}} \mathcal{Q}(\mathbf{z_v}) \frac{\mathrm{Tr}[e^{-\beta H_{\mathbf{v}}}\partial_\theta H_{\mathbf{v}}]}{\mathrm{Tr}[e^{-\beta H_{\mathbf{v}}}]} - \frac{\mathrm{Tr}[\partial_\theta H_{\rm QBM} e^{-\beta H_{\rm QBM}}]}{\mathrm{Tr}[e^{-\beta H_{\rm QBM}}]}, \tag{14}$$

where $H_{\mathbf{v}}$ is the visible-clamped Hamiltonian. More details of derivations of *bound-based* training is presented in Appendix A. Two terms of Eq. (14) can be sampled efficiently. Hence, the parameter can be updated through the gradient ascent method, namely, $\theta^{(t+1)} = \theta^{(t)} + \alpha\partial_\theta\widehat{\mathcal{L}}_Q$. Due to the clamped Hamiltonian $H_{\mathbf{v}}$, the positive phase is efficiently computable on a classical Turing machine [30], except for the transverse field parameter Γ_i, whose gradient vanishes under the clamp operation. However, the negative phase is not believed to be efficiently classically computable in general. The exact calculation of it, such as exact diagonalization, would need exponential complexity. In this work, we mainly focus on efficient and approximate sampling on the negative phase via the path integral Monte Carlo (PIMC) algorithm.

III. EFFICIENT TRAINING VIA QUANTUM MONTE CARLO METHOD

A. General approach

We make use of the PI formulation to describe the quantum thermal distribution. The fundamental sampling algorithm is based on the PI formalism, which transforms the quantum Ising Hamiltonian into a classical one by adding an extra dimension [31,32]. Sampling from the paths with the corresponding probability can provide the approximation expectation of observable. For convenience, we adopt the quantum restricted Boltzmann machine (QRBM) in Appendix B as an example, whose illustration is displayed in Fig. 1(c). The Hamiltonian of QRBM is presented as follows:

$$H^r_{\rm QBM} = -\sum_i \Gamma_i \sigma_i^x - \sum_{v,h} J_{vh}\sigma_v^z\sigma_h^z - \sum_v a_v\sigma_v^z - \sum_h b_h\sigma_h^z. \tag{15}$$

In the context of PI formulation, the partition function of quantum thermal distribution is given by

$$\begin{aligned} Z_Q &= \mathrm{Tr}[e^{-\beta H^r_{\rm QBM}}] = \lim_{P\to\infty} \mathrm{Tr}[e^{(-\beta H^r_{\rm QBM}/P)^P}] \\ &= \lim_{P\to\infty} \sum_{\mathbf{z}^1} \cdots \sum_{\mathbf{z}^P} \langle \mathbf{z}^1 | e^{-\frac{\beta}{P}(H_{\text{diag}} + H_{\text{off-diag}})} | \mathbf{z}^2\rangle \langle \mathbf{z}^2 | \\ &\quad \times e^{-\frac{\beta}{P}(H_{\text{diag}} + H_{\text{off-diag}})} \ldots |\mathbf{z}^P\rangle\langle\mathbf{z}^P| e^{-\frac{\beta}{P}(H_{\text{diag}} + H_{\text{off-diag}})} |\mathbf{z}^1\rangle, \end{aligned} \tag{16}$$

where P is the number of Trotter slices, $\mathbf{Z} = \{\mathbf{z}^k, k = 1, \dots, P\}$ is the full configuration of all $N \times P$ qubits, and $\mathbf{z}^k = \{\mathbf{Z}_l^k, l = 1, \dots, N\}$ denotes the configuration of N qubits in the kth Trotter slices. The restricted quantum Hamiltonian is decomposed into the diagonal term H_{diag} and the nondiagonal term $H_{\text{off-diag}}$. In order to approximate the partition function, we apply Suzuki-Trotter breakup [33],

$$e^{-\beta H^r_{\text{QBM}}/P} = e^{-\beta H_{\text{diag}}/P} e^{-\beta H_{\text{off-diag}}/P} + O((\beta/P)^2), \qquad (17)$$

and obtain the approximate partition function Z_P whose error is proportional to the square of the Trotter breakup time $\Delta\tau = \beta/P$. Hence, the approximate partition function when the second order of $\Delta\tau$ is neglected is given by

$$Z_Q \approx Z_P = \prod_{i,k} \mathcal{C}_i \sum_{\mathbf{Z}} e^{-\beta H^r_{\text{eff}}/P}, \qquad (18)$$

where $\mathcal{C}_i = \sqrt{\frac{1}{2}\sinh(2\beta\Gamma_i/P)}$, and $\mathbf{Z} = \{\mathbf{z}^1, \mathbf{z}^2, \dots, \mathbf{z}^P\}$ denotes the full configurations of qubits in the QRBM. H^r_{eff} represents the effective quantum Hamiltonian given by

$$H^r_{\text{eff}} = -\sum_{k=1}^{P} \left(\left(\sum_i \tilde{\mathcal{B}}_i \mathbf{Z}_i^k \mathbf{Z}_i^{k+1} + \sum_{v,h} J_{vh} \mathbf{Z}_v^k \mathbf{Z}_h^k \right.\right.$$
$$\left.\left. + \sum_v a_v \mathbf{Z}_v^k + \sum_h b_h \mathbf{Z}_h^k \right) \right), \qquad (19)$$

with $\tilde{\mathcal{B}}_i = -\frac{P}{2\beta} \ln \tanh(\beta\Gamma_i/P), \mathbf{Z}_i^j \in \{-1, +1\}$. The effective Hamiltonian is classically presented, since the qubit is collapsed to the classical value. In the context of *bound-based* training, Γ_i cannot be trained, and we treat it as a hyperparameter Γ [16], i.e., each σ^x will be attached with the same transverse field. We introduce a classical probability measure for the visible and hidden qubits,

$$\mu(\mathbf{Z}) = \frac{1}{Z'_P} e^{-\beta H^r_{\text{eff}}/P}, \qquad (20)$$

where $Z'_P = \mathcal{C}^{NP}/Z_P$. Thus, the expectation of observable O over the quantum thermal distribution ρ can be obtained as given by

$$\mathbb{E}_\rho[O] = \sum_{\mathbf{Z}} \mu(\mathbf{Z}) \frac{\langle \mathbf{z}^k | O e^{-\beta H^r_{\text{QBM}}/P} | \mathbf{z}^{k+1} \rangle}{\langle \mathbf{z}^k | e^{-\beta H^r_{\text{QBM}}/P} | \mathbf{z}^{k+1} \rangle}$$
$$= \sum_{\mathbf{Z}} \mu(\mathbf{Z}) \frac{1}{P} \sum_{k=1}^{P} O(\mathbf{z}^k). \qquad (21)$$

B. Efficient sampling based on continuous-time quantum cluster Monte Carlo

A conventional spin-flipping algorithm such as local and single-spin updating requires large MC sweeps to converge, having strong autocorrelation between the successive configurations. However, it is intolerable for the gradient training process with a low convergent rate for each parameter updating. In addition, the discrete-time quantum MC has suffered form the systematic error caused by the Suzuki-Trotter approximation. Thus, we apply the state-of-the-art continuous-time quantum cluster Monte Carlo (CT-QCMC) method to update the configurations nonlocally, which has been proven to be efficient in simulating the quantum bosonic models [34–36].

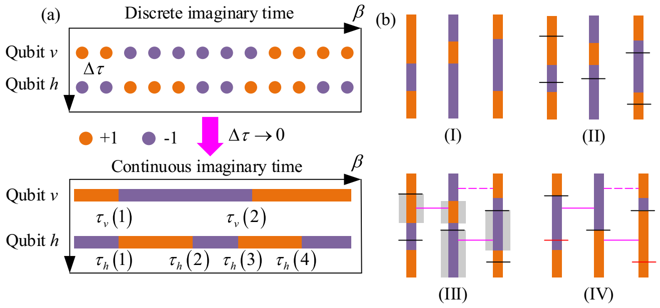


FIG. 2. The schematic for continuous-time quantum cluster updating algorithm. (a) Illustration of continuous-time quantum cluster algorithm by taking the time limit $\Delta\tau \to 0$ for its discrete version. (b) The cluster configuration updating process consists of four steps: (1) initial configuration before the update, (2) insertion cuts randomly via Poissonian statistics, and (3) placing bonds between neighboring qubits through Poissonian statistics, shaded area denotes clusters and dotted bonds will be rejected. (4) Flipping clusters to $+1$ or -1 with equal probability. The redundant cuts (red horizontal line) should be removed before the next updating.

CT-QCMC consists of two core steps: (1) updating the configuration with the standard cluster (Swendsen-Wang) algorithm, and (2) applying the Metropolis algorithm to determine whether the updated configuration is accepted or rejected. The schematic of CT-cluster updating algorithm is illustrated in Fig. 2. The derivation of CT-QCMC in QRBM is presented in Appendix C. The discrete-time spin configurations $\mathbf{Z}$ are inaccessible to be stored in the CT limit. Instead, we specify for every visible or hidden neuron only the imaginary time locations of the kinks and the spin values just after these kinks, denoted by

$$\tilde{\mathbf{Z}} = \{[\tau_1(1), z_1(1)], \dots, [\tau_1(n_1), z_1(n_1)],$$
$$[\tau_N(1), z_N(1)], \dots, [\tau_N(n_N), z_N(n_N)]\},$$

where $\tau_i(n_j)$ denotes the n_jth kink of one imaginary time line of site i, and n is the number of cuts on an imaginary time line. Therefore, the memory requirement for this CT-QCMC algorithm is proportional to the number of cuts, namely, $O(\beta\Gamma N^2)$ compared with the discrete-time representation of $O(\beta N^2/\Delta\tau)$. The memory requirement is reduced by at least $O(P/(\beta\Gamma))$. In order to sample from the partition function in the continuous-time limit, the discrete PI formulation is represented as the continuous one. When $\Delta\tau \to 0$, the partition function becomes exact, given by

$$Z_Q = \int_{\tilde{\mathbf{Z}}} \mathcal{D}z(\tau) \exp\left(-\int_0^\beta d\tau \mathcal{E}(z(\tau)) \right), \qquad (22)$$

where the classical energy function is given by

$$\mathcal{E}(z(\tau)) = \sum_{v,h} J_{vh} z_v(\tau) z_h(\tau) + \sum_v a_v z_v(\tau) + \sum_h b_h z_h(\tau), \qquad (23)$$

where $\mathcal{D}z(\tau)$ denotes a Poisson measure on the space of spin-1/2 imaginary time lines. The neighboring interaction of spins along with the imaginary time line is vanished in the continuous-time limit, since the integrand is a τ-independent item (i.e., $z_i(\tau)^2 = 1$, $\lim_{P\to\infty} \tilde{\mathcal{B}} = \ln \Gamma/2$) such that the integral becomes a constant. The classical probability measure now is given by

$$\mu(\tilde{\mathbf{Z}}) = \frac{1}{Z_Q} \exp\left(-\int_0^\beta d\tau \mathcal{E}(z(\tau))\right). \tag{24}$$

Thus, Eq. (21) is rewritten as

$$\mathbb{E}_\rho[O] = \int_{\tilde{\mathbf{Z}}} \mathcal{D}z(\tau)\mu(\tilde{\mathbf{Z}})\frac{1}{\beta}\int_0^\beta O(z(\tau))d\tau. \tag{25}$$

More specifically, the gradient of the quantum Hamiltonian with respect to three different parameters is given respectively by

$$\mathbb{E}_\rho\left[\sigma_v^z\sigma_h^z\right] = \int_{\tilde{\mathbf{Z}}} \mathcal{D}z(\tau)\mu(\tilde{\mathbf{Z}})\frac{1}{\beta}\int_0^\beta d\tau z_v(\tau)z_h(\tau), \tag{26a}$$

$$\mathbb{E}_\rho\left[\sigma_v^z\right] = \int_{\tilde{\mathbf{Z}}} \mathcal{D}z(\tau)\mu(\tilde{\mathbf{Z}})\frac{1}{\beta}\int_0^\beta d\tau z_v(\tau), \tag{26b}$$

$$\mathbb{E}_\rho\left[\sigma_h^z\right] = \int_{\tilde{\mathbf{Z}}} \mathcal{D}z(\tau)\mu(\tilde{\mathbf{Z}})\frac{1}{\beta}\int_0^\beta d\tau z_h(\tau). \tag{26c}$$

Integrals with quantum Boltzmann distribution such as Eqs. (26a)–(26c) are best sampled by generating a configuration using the MCMC method. The details of the CT-QCMC algorithm and the variance analysis are demonstrated in Appendix C.

C. Quantum mean-field approximation

In order to further accelerate the training process, we introduce the QMF approximation into the partition function in the PI formulation. More specifically, we group each layer into a cluster, and the intercluster interactions can be replaced by the average value of the whole Trotter slice in imaginary time, as Fig. 1(d) shows. The classical mean-field theory, however, discards all the correlations between each cluster (layer) and preserves only the correlations in the cluster; therefore it cannot be applied in the QRBM, since it will lead to the weight failing to be learned. Conversely, the QMF approach discards only the quantum fluctuation between each layer and retains the thermal effect (averaged spins) to learn the parameter (weight) [26]. More formally, Eq. (19) in the context of QMF is rewritten as

$$H_{\mathrm{QMF}}^r = -\sum_{k=1}^P \left(\sum_i \tilde{\mathcal{B}}_i \mathbf{Z}_i^k \mathbf{Z}_i^{k+1} + \sum_{v,h} J_{vh}\bar{\mathbf{Z}}_v\bar{\mathbf{Z}}_h + \sum_v a_v \mathbf{Z}_v^k + \sum_h b_h \mathbf{Z}_h^k\right), \tag{27}$$

where $\bar{\mathbf{Z}}_i = \frac{1}{P}\sum_{k=1}^P \mathbf{Z}_i^k$ is the averaged spin value of neuron i along with the imaginary time line. It is important to remark that the QMF approximation affects only the spacelike couplings J_{vh}. In the QMF approximation, the partition function is rewritten as

$$Z_{\mathrm{QMF}} = \int_{\tilde{\mathbf{Z}}} \mathcal{D}z(\tau) \exp\left(-\int_0^\beta d\tau \mathcal{E}_{\mathrm{QMF}}(z(\tau))\right), \tag{28}$$

where the energy in the QMF approximation is given by

$$\mathcal{E}_{\mathrm{QMF}}(z(\tau)) = \sum_{v,h} J_{vh}\langle z_v(\tau)\rangle\langle z_h(\tau)\rangle + \sum_v a_v z_v(\tau) + \sum_h b_h z_h(\tau), \tag{29}$$

with $\langle z_i(\tau)\rangle = \beta^{-1}\int_0^\beta z_i(\tau)d\tau$. Similarly, the classical probability measure over the QMF approximation is given by

$$\mu_{\mathrm{QMF}}(\tilde{\mathbf{Z}}) = \frac{1}{Z_{\mathrm{QMF}}} \exp\left(-\int_0^\beta t d\tau \mathcal{E}_{\mathrm{QMF}}(z(\tau))\right). \tag{30}$$

It is clear that the expectation of observables involving the bias parameters a_v, b_h is the same with the nonapproximation case. The expectation of an observable involving the interaction parameter J_{vh}, i.e., Eq. (26a), in the context of the QMF approximation, is rewritten as

$$\mathbb{E}_\rho\left[\sigma_v^z\sigma_h^z\right] = \int_{\tilde{\mathbf{Z}}} \mathcal{D}z(\tau)\mu_{\mathrm{QMF}}(\tilde{\mathbf{Z}})\langle z_v(\tau)\rangle\langle z_h(\tau)\rangle. \tag{31}$$

It is important to remark that the CT-QCMC framework still works under the situation of QMF approximation. The classical energy $\mathcal{E}(z(\tau))$ is modified to a simplified expression under the situation of QMF approximation when taking the spin value of one imaginary time line into the averaged spin value. Thus, accept the probability $\mathcal{R}$ as simplified correspondingly from Eq. (C1). The Markov process will eventually converge to the stationary probability $\mu_{\mathrm{QMF}}(\tilde{\mathbf{Z}}) \cdot \tilde{\mathbf{Z}}$ when making use of the QMF approximation. The mean value of Eq. (31) will be obtained by averaging the observable samples, and the statistical error of the estimate also shrinks linearly with the MC sweeps R.

In theory, the QMF approximation is able to reach thermal equilibrium faster compared to the exact evolution in spin networks by choosing an appropriate cluster parameter [26]. As for the QRBM's training, there is a possibility of rendering the network converged to the stable and optimal state by using less training time. In addition, when using QMF approximation, the complexity of calculating the energy function can be reduced in continuous-time world lines. More specifically, calculating the energy function over the imaginary time β requires the integration over a piece-wise function for each pair of visible-hidden sites because the sign of two neighboring spins is opposite for each time cut τ. Suppose each world line contains n time cuts on average. One could use the *MergeSort* oracle to render the two time-cut lists into a sorted one, and it would then cost $2n$ multiplications and $(n-1)$ additions on average to obtain the integration value for two visible-hidden pairs of slices. *MergeSort* costs $O(n \ln n)$ time complexity in general. After QMF approximation, the energy function can be calculated, requiring only one multiplication and $2(n-1)$ additions to complete the integration along with the slice for each visible-hidden pair and the *MergeSort* is not required. In fact, the number of time cuts in each slice is $O(\beta\Gamma)$. Therefore, for each interaction pair, there is approximately polynomial speedup $O[(\beta\Gamma)^2 \ln(\beta\Gamma)]$ in terms of multiplication times. The computational speedup will lead to a larger size of neural networks compared to the CT-QCMC algorithm.

D. Generative algorithm for efficient training QBM

For simplicity, we first consider the QRBM as the exemplar. Since the restricted connections of the QRBM lie between the visible and hidden layers, the gradient of the upper bound of the exact log-likelihood function can be deduced from Eq. (14), given by

$$\beta^{-1}\partial_\theta\widehat{\mathcal{L}} = \sum_{\mathbf{z_v}} \mathcal{Q}(\mathbf{z_v})\mathbb{E}_{\text{clamp}}\left[\partial_\theta H_{\mathbf{v}}^r\right] - \mathbb{E}_\rho\left[\partial_\theta H_{\text{QBM}}^r\right], \tag{32}$$

where $\mathbb{E}_{\text{clamp}}[\cdot]$ denotes the expectation over the clamped quantum Boltzmann distribution, $H_{\mathbf{v}}^r$ represents the clamped Hamiltonian of the QRBM. Furthermore, the positive phase, i.e., the data expectation over the clamped quantum Boltzmann distribution for θ combined with $\mathcal{Q}(\mathbf{z_v})$, can be deduced explicitly, which is given by

$$\mathbb{E}_+\left[\sigma_v^z\sigma_h^z\right] = \sum_{\mathbf{z_v}} \mathcal{Q}(\mathbf{z_v})\frac{z_v\tilde{b}_h(\mathbf{z_v})}{\Phi}\tanh(\beta\Phi) \tag{33a}$$

$$\mathbb{E}_+\left[\sigma_v^z\right] = \sum_{\mathbf{z_v}} \mathcal{Q}(\mathbf{z_v})z_v \tag{33b}$$

$$\mathbb{E}_+\left[\sigma_h^z\right] = \sum_{\mathbf{z_v}} \mathcal{Q}(\mathbf{z_v})\frac{\tilde{b}_h(\mathbf{z_v})}{\Phi}\tanh(\beta\Phi), \tag{33c}$$

where $\Phi = \sqrt{\Gamma^2 + \tilde{b}_h^2(\mathbf{z_v})}$, and $\mathbb{E}_+[\cdot]$ denotes the positive phase expectation. It is found that when $\Gamma \to 0$, the positive phase is the same as the classical RBM, as Eqs. (10a)–(10c) show. The negative phase, i.e., the model expectation over the quantum Boltzmann distribution for θ, can be obtained by sampling from Eqs. (26a)–(26c) by using the CT-QCMC algorithm. A more efficient expectation estimation can be obtained by using the QMF approximation. The detailed training process is presented below:

Algorithm 1. Efficient algorithm for QRBM training.

```
Input: Training set: $\hat{\mathbf{z}}^{(i)}, i = 1, \ldots, m$; learning rate $\alpha$; traverse
    field hyperparameter $\Gamma$
Initialization: $\boldsymbol{J} \leftarrow 0, \boldsymbol{a} \leftarrow 0, \boldsymbol{b} \leftarrow 0$
while not converged do
    for $i = 1, \ldots m$ do
        for all visible and hidden qubits do
            Choose a sample $\hat{\mathbf{z}}^{(i)}$, calculate the positive phase
            for $\theta$ by using Eqs. (33a)–(33c)
            Obtain the negative phase for $\theta$ by sampling from
            Eqs. (26a)–(26c) or Eq. (31) with an efficient method
            Calculate the gradient of Hamiltonian parameters
            $\partial_{J_{vh}}\widehat{\mathcal{L}}, \partial_{a_v}\widehat{\mathcal{L}}, \partial_{b_h}\widehat{\mathcal{L}}$ by Eq. (32)
            $J_{vh} \leftarrow J_v h - \alpha\partial_{J_{vh}}\widehat{\mathcal{L}}$        ▷ Updating $\theta$
            $a_v \leftarrow a_v - \alpha\partial_{a_v}\widehat{\mathcal{L}}$
            $b_h \leftarrow b_h - \alpha\partial_{b_h}\widehat{\mathcal{L}}$
        end for
    end for
end while
Output: $\boldsymbol{J}, \boldsymbol{a}, \boldsymbol{b}$
```

It is necessary to remark that the positive phase formulation for semirestricted QBM has a similar form. The negative phase can also be calculated by invoking Algorithm 1. The difference between these two QBMs is the Hamiltonian parameter replaced with $\theta = \{J_{vi}, b_i\}$. For a fully connected QBM, the positive phase cannot be calculated explicitly. However, the positive phase can be obtained efficiently, since the training set $\{\hat{\mathbf{z}}\}$ scales with polynomial size. The negative phase of full connected QBM is efficiently estimated by the CT-QCMC algorithm . As a result, Algorithm 1 is also available for these types of QBMs. As for restricted ZZXX QBM, when using the Golden-Thompson inequality, the clamped operation i.e., $\langle\mathbf{z_v}|H_{\text{QBM}}^{\text{ZZXX}}|\mathbf{z_v}\rangle$, will eliminate the term $\sigma_v^x\sigma_h^x$ and the rest of the terms in the Hamiltonian are the same as the case of H_{QBM}^r. A similar mathematical operation can be provided for a restricted ZX QBM. Therefore, the positive phase for both more complex QBMs can be carried out explicitly. More calculation details are presented in Appendix B. However, the negative phase updating is not the same thing as the case of QRBM, because restricted *ZX* and ZZXX quantum Hamiltonians belong to the κ-local ($\kappa \geqslant 2$) Hamiltonian, which is proven to be QMA complete [24]. The state-of-the-art quantum Monte Carlo method still requires exponential time to converge, which thus renders the training process inefficient or even inaccessible. For small-scale QBM, the negative phase can be solved by exact diagonalization, which is an exponential complexity algorithm.

IV. EXPERIMENTAL ANALYSIS

A. Small-scale data set

We first apply the standard test in Refs. [16,22] as the benchmark. A multimode multivariate Bernoulli distribution is capable of representing the distribution of binary images [37], which is given by

$$\mathcal{Q}(\mathbf{z_v}) = \frac{1}{M}\sum_{i=1}^{M}\mathcal{P}_B(\mathbf{z_v}; p_i, \boldsymbol{s}_i), \tag{34}$$

where

$$\mathcal{P}_B(\mathbf{z_v}; p_i, \boldsymbol{s}_i) = p_i^{N_v - |(\mathbf{z_v} - \boldsymbol{s}_i)/2|}(1 - p_i)^{|(\mathbf{z_v} - \boldsymbol{s}_i)/2|} \tag{35}$$

is the *i*th mode of the Bernoulli distribution centered at $\boldsymbol{s}_i = \{s_i^1, s_i^2, \ldots, s_i^{N_v}\}$. Note that $|\boldsymbol{x}|$ denotes the number of components equal to -1 of $\boldsymbol{x}$. The modes' center points $\boldsymbol{s}_i^v \in \{+1, -1\}$ are randomly and independently generated from a uniform distribution for each training data. N_v is the number of visible qubits for QBM, p_i is the probability of qubit $|\mathbf{z}_v\rangle$ being aligned with $\boldsymbol{s}_i^v$. Each single-mode Bernoulli distribution is in fact a factorizable multivariate Bernoulli distribution. Here, we adopt M modes and equal weight to compose the ultimate equal-weighted artificial data set. In the experiments, p_i is set to be 0.9 identically for each mode. Each data set contained 1000 training examples. Generally, larger M indicates a more complex data distribution. The artificial data generated by multi-mode Bernoulli distribution is typical for illustrating the learning ability of the machine learning model in a small-scale data set.

To measure the learning ability of the machine learning model, we adopt the KL divergence as the metric. The KL divergence is given by

$$\mathrm{KL} = \mathcal{L} - \mathcal{L}_{\min} = \sum_{\mathbf{z_v}} \mathcal{Q}(\mathbf{z_v}) \ln \frac{\mathcal{Q}(\mathbf{z_v})}{\mathcal{P}(\mathbf{z_v})}. \tag{36}$$

The KL value is non-negative and reaches the minimum value 0 when two distributions are identical.

In the experiment, the restricted ZX, ZZXX and semi-restricted QBMs and QRBM are tested by using the artificial data set compared to RBM. The number of visible and hidden neurons is set to be $N_v = 8$ and $N_h = 2$, respectively. The transverse field parameter is treated as a hyperparameter and is set to be $\Gamma = 2$ [16]. The QRBM is trained via Algorithm 1. The positive phase of ZX and ZZXX QBMs adopts the equations in Appendix B. The negative phase updating is also executed by the CT-QCMC method where the respective Hamiltonian is changed but the concrete algorithm framework is aligned with Algorithm 1. The other model parameters such as J_{vh}, a_v, b_h are tunable and optimized in the training process. Figure 3 shows the performance of QBMs and CBM.

It is found that QRBM presents the best performance compared to other machines and reaches the minimum KL divergence, KL ≈ 0.30. Moreover, all QBMs are superior to CBM under the same training data, which demonstrates that QBMs preserve a more powerful learning ability. Restricted ZX (RZX) and ZZXX (RZZXX) QBM nearly have the same performance, probably because they fall into the same complexity class. In principle, RZX and RZZXX QBMs should have a smaller KL value than QRBM, since the latter is easier to be simulated by a classical Turing machine. However, in the context of a *bound-based* training example, their performance is weakened, perhaps due to the clamp operation. The converged KL value of a semirestricted QBM (QSBM) is only larger than the QRBM but less than other models, which illustrates that restrictions are not necessarily required in BM when using *bound-based* training. Here, QRBM is trained by QMF-approximated and CT-QCMC algorithms, respectively. Both algorithms reach the optimal KL distance indicated by Fig. 3(a) by using the appropriate MC sweeps. In particular, we find that QMF-approximated algorithm exhibits slightly faster convergence in the early stage of training. This advantage is gradually decreased compared to the CT-QCMC training algorithm. Nevertheless, the QMF-approximated training algorithm still improves the KL distance with the increase of iterations. Moreover, we calculate the difference of logarithm of partition function between QMF, CT-QCMC, and their exact value (i.e., $\ln \hat{Z} - \ln Z$) to demonstrate their convergence rate, as Figs. 3(b) and 3(c) shows. Note that the exact value is available for only a small-scale data set. Here, the estimated partition function of QMF and CT-QCMC trained QRBM are calculated by annealed importance sampling (AIS) [38,39], in which we used 10^3 β_k spaced uniformly from 0 to 1. In Fig. 3(b), the difference and their variance approaches 0 when the AIS runs increase, which implies that more runs are required to estimate the exact value. It is found that 100 AIS runs are enough to estimate the partition function in a small-scale data set. In Fig. 3(c), we find that the estimated value approaches the exact value, implying that two algorithms are improving the QRBM's performance with the increase of iterations. In particular, the QMF-approximated algorithm has a smaller difference when iterations are less than 10, which indicates that the QMF algorithm perseveres at a faster convergence rate.

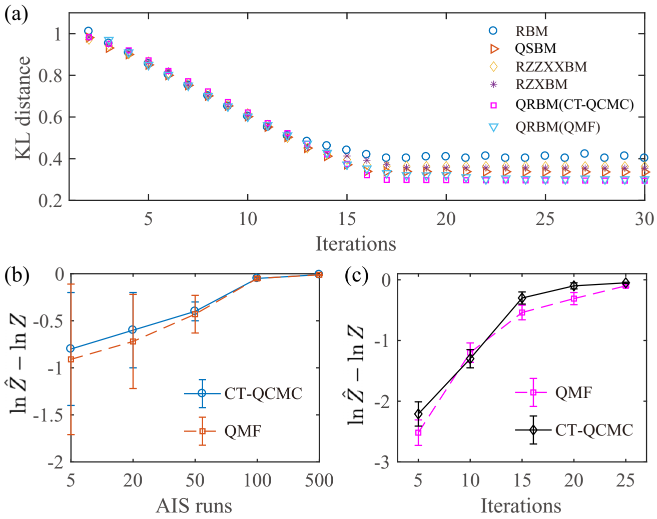


FIG. 3. Performance illustration of various QBMs compared to RBM. The parameters are set as follows: $\Gamma = 2.5$, $M = 8$, $N_v = 8$, $N_h = 2$. The MC sweeps is set to be 10^3 of each iteration and the learning rate $\alpha = 10^{-3}$. (a) KL distance is the average value of 100 repeated experiments. (b) Comparison of QMF approximation and CT-QCMC algorithm in terms of $\ln \hat{Z} - \ln Z$ when given different AIS runs. $\ln Z$ is calculated by a brute-force algorithm, and the variance is $\pm 3\sigma$ [the same with (c)]. (c) Tracking of partition function during training process with QMF and CT-QCMC algorithm. Each record is estimated over 100 AIS runs.

As a result, the QMF approximation is more efficient and it requires fewer iterations to reach a suboptimal KL distance, although a larger number of iterations is required to reach the optimal KL distance. In addition, although this superiority is not significant in a small data set, the experimental results still imply that the QMF approximation is feasible to train QBMs.

B. Large-scale data set

In order to further illustrate the efficiency of QMF algorithm, we train QRBM via algorithm 1 and its QMF approximation in the MNIST data set. MNIST is a well-known data set consisting of 0–9 handwritten digital images. Each image is composed of $28 \times 28 = 784$ pixels, and each pixel can be represented by a binary neuron of BM. The data set consists of 6×10^4 training and 10^4 testing images and their corresponding labels. However, the original data set requires 784 visible qubits to encode the data, which is expensive for NISQ devices. Moreover, although the proposed algorithm is able to improve the training efficiency, it is still computationally expensive, preventing us from training 784 qubits. Therefore, before training the large-scale images set, the data dimension requires it to be reduced to adapt to our experimental settings. We first employ the principal component analysis (PCA) to analyze the data redundancy, which is achieved through finding a lower-dimensional representation of the data by using an orthogonal transformation to project the data onto uncorrelated directions [40]. It is found

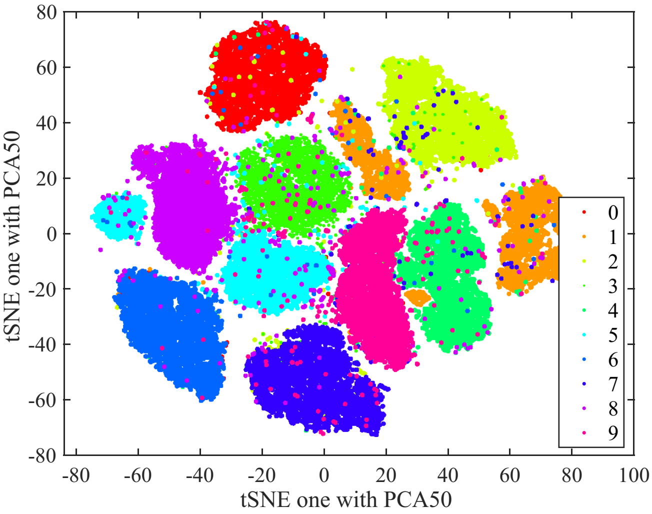
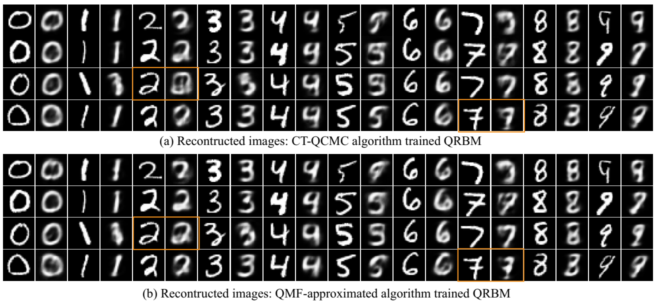


FIG. 4. t-SNE visualization of image data with 50 dimensions reduced by PCA. The learning rate of t-SNE is set to be 500 and the perplexity is 30. Ten colors map ten different categories of handwritten digital images.



(a) Recontructed images: CT-QCMC algorithm trained QRBM

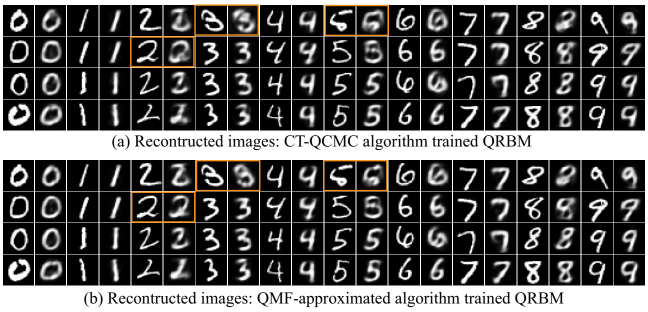
(b) Recontructed images: QMF-approximated algorithm trained QRBM

FIG. 5. Reconstructed MNIST samples on the test set. The size of QRBM is 36 × 36. Total iterations are 150 and $\Gamma = 2$. In experiments (a) and (b), the number of population replicas is 2000. Ten and five MCMC sweeps are considered for (a) and (b) per gradient update. The total reconstruction error for (a) and (b) is 0.075 and 0.052, respectively. Yellow boxes highlight the obvious better reconstruction.

that nearly 85% variation of the data information is retained when the dimension reduces to 50, which demonstrates that a reduced image set will not decrease network performance dramatically. We visualize the dimension reduced training data by means of the t-distributed stochastic neighboring entities (t-SNE) algorithm [41], as Fig. 4 shows. The image data is well clustered into 10 categories, which indicates that nearly 50 dimensions are a feasible choice for our model training. We adopt the static binarization to process the image set. Here, we utilize PCA only to illustrate the feature redundancy and not to change the basis before feeding the data to the network. Our dimension-reduced data is achieved by using a variational auto-encoder which will not change the data basis and is proven to possess better performance in terms of dimension reduction [42]. The dimension-reduced image will be regarded as input to train our quantum model. The final reconstructed images by QRBM are reshaped into the original size through cubic interpolation for convenient comparison with the original images.

We adopt different N_v to demonstrate the reconstruction performance of QRBM, where QRBM is trained by CT-QCMC and QMF-approximated algorithms, respectively. We do not analyze the performance of RBM and leave the comparison in the classification task. To measure the performance of BM quantitatively, we calculate the estimation value of the logarithm of the quantum partition function by population annealing (PA) [25,43], and the ultimate exact value of QRBM is not calculated due to expensive computation costs [44]. We also do not estimate the log-likelihood function, since it requires calculating the quantum probabilities for a large number of latent configurations, even for a medium number of hidden qubits, which is also computationally expensive. The PA method is employed to accelerate CT-QCMC and QMF algorithms. We have used the number momentum method with momentum equal to 0.95 and batch normalization [45] to increase the speed of learning. The batch size is chosen as 100. Note that other batch sizes present similar results, and small batch sizes only result in less stable learning curves. Figures 5 and 6 display a proportion of reconstructed MNIST samples from 0 to 9 handwritten digital images by QRBM (even and odd columns are original and reconstructed images, respectively). To analyze the convergence rate, the partition function of both algorithms is estimated under different MC sweeps, as Fig. 7 shows. We do not report the estimated partition function during the training process due to the expensive computation cost.

It is found that two algorithms are capable of efficiently training QRBM in large-scale data sets, and the QMF-approximated algorithm shows better performance in terms of reconstruction error in Figs. 5 and 6. In particular, the advanced performance of the QMF approximation is magnified in large-size QRBM and it requires fewer MC sweeps to reach the optimal partition function from Fig. 7(b). The variance of the QMF approximation is also smaller compared to the CT-QCMC algorithm, although large enough MC sweeps will render both algorithms converged to the ultimate partition function. In a relatively small network, the partition function of QMF still shows a slightly faster convergence rate than



(a) Recontructed images: CT-QCMC algorithm trained QRBM

(b) Recontructed images: QMF-approximated algorithm trained QRBM

FIG. 6. Reconstructed MNIST samples on the test set. The size of QRBM is 81 × 81. Total iterations are 100 and $\Gamma = 2$. In experiment (a), 2500 population replicas and eight MCMC sweeps are considered for each gradient evaluation. In experiment (b), the number of population replicas is 1000, and MCMC sweeps are four per gradient evaluation. The total reconstruction error for (a) and (b) is 0.0312 and 0.0213, respectively.

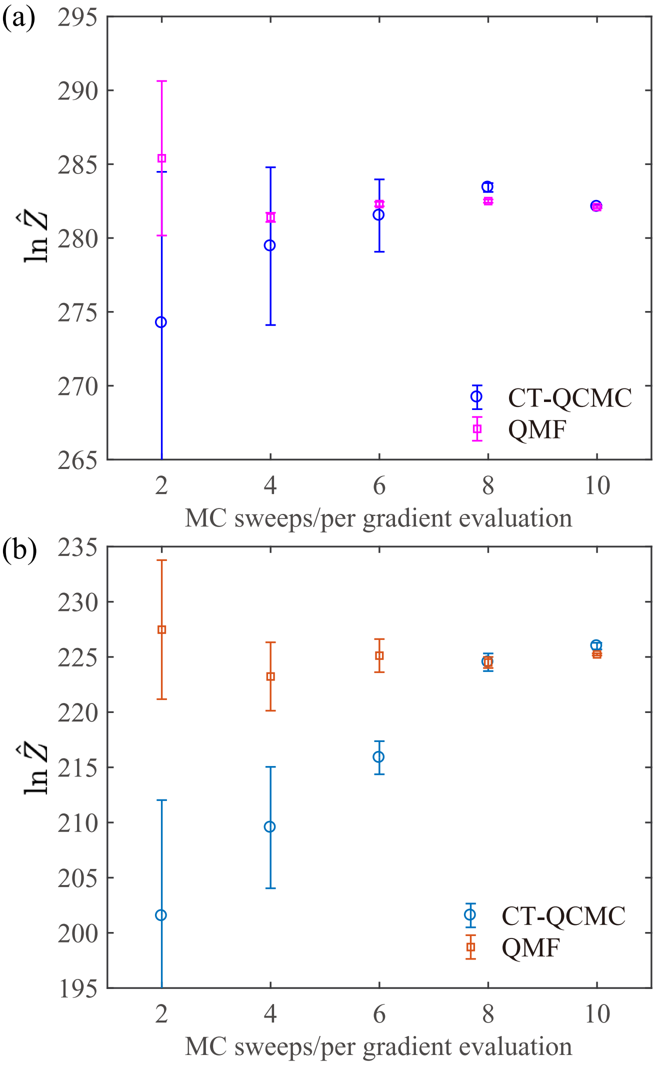


FIG. 7. Logarithm of partition function comparison between CT-QCMC and QMF approximation algorithm under different MC sweeps. Each estimation value uses 14 500 annealed β_k spaced uniformly in [0,1] and 3000, 1500 population replicas are considered in (a) and (b). The variance is $\pm 3\sigma$.

CT-QCMC in Fig. 7(a). As a result, the QMF approximation is able to accelerate the QRBM's training and improves the quality of generated images.

To achieve the classification by using QRBM, it is required that a top supervised learning layer be added to complete the task, including the following: (1) The QRBM first learns the optimal model parameters via Algorithm 1 by using training data and adopts the one-hot encoding method to encode the training labels. (2) Feed the training and test images into the visible qubits, and obtain encoded data by hidden qubits. (This step is equivalent to a dimensions reduction.) (3) Feed the QRBM-encoded data and the one-hot encoded labels into the softmax learner (both for training and testing data). (4) Train the softmax learner to obtain prediction (classification) values of training data and test data. Note that steps (3) and (4) are the standard method of using BM to achieve the classification task [2]. A trained BM is an encoder to complete the feature extraction. This task is different from the discriminative learning using QRBM in which the conditional distribution is required and is inaccessible when using *bound-based* training [16].

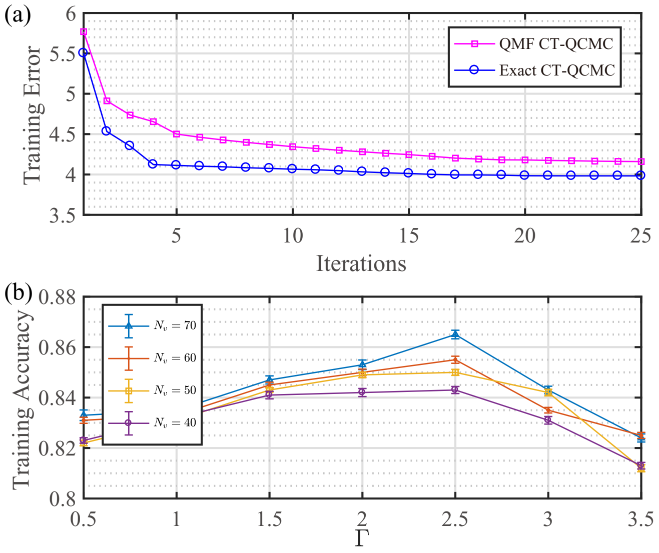


FIG. 8. QRBM performance test under different parameters. 3000 replicas and eight MCMC sweeps per gradient evaluation are used. Momentum is 0.9, and $\alpha = 10^{-3}$. (a) Error during training QRBM with CT-QCMC compared to training with QMF-approximated algorithm. $N_v = 50, N_h = 10$. (b) Accuracy of training set of QRBM with different visible qubits $N_v = 70, 60, 50, 40$ and the same hidden qubits $N_h = 10$. The error bars are the variance of 100 repeated experiments with QMF-approximated Algorithm 1.

We depict the ℓ_2-norm error of two algorithms during training QRBM in Fig. 8(a). As can be seen, although the error of the QMF-approximated algorithm is larger than the exact one, it can be neglected at the final stage of training. Therefore, we are convinced that QMF-approximated Algorithm 1 is feasible and efficient for the classification task. Furthermore, in order to find the optimal hyperparameter of Γ in the classification task, we test accuracy on the training set with different Γ in Fig. 8(b) for different visible qubits ($N_v = 70, 60, 50, 40$) via a QMF-approximated training algorithm, and it is found that better performance is displayed at different Γ values. In general terms, $\Gamma \in [1, 3]$ exhibits better performance for different visible qubits. In addition, a larger number of visible qubits shows better classification accuracy, which is capable of a more powerful representation ability. Note that $N_h = 10$ is chosen because a large number of hidden units will lead to overfitting. In practice, the accuracy of the independent test set, however, represents the generalization ability of the machine learning model. Thus, we further test both the training and test accuracy with the QMF-approximated algorithm on the training and test sets, respectively. RBM has the same number of neurons as a comparison. The training accuracy and test accuracy for QRBM and RBM are presented in Table I.

It is found that QRBM has a higher training and test accuracy than RBM. However, we can see that when the number of visible neurons increases, the test accuracy is not increased accordingly and the training accuracy is nearly the same, which implies that less qubits have more powerful generalization ability. Therefore, we can conclude that when

TABLE I. Training and test accuracy for RBM and QRBM.

	RBM		QRBM	
	Training accuracy	Test accuracy	Training accuracy	Test accuracy
$N_v = 70$	0.828	0.865	**0.856**	**0.878**
$N_v = 50$	0.825	0.881	**0.850**	**0.895**
$N_v = 40$	0.822	0.892	**0.848**	**0.906**

$N_v = 70, 50$, the model is overfitting. By decreasing the number of visible qubits, the overfitting problem can be reduced efficiently. In particular, the QRBM trained by Algorithm 1 in large-scale data set presents better performance than RBM, which indicates that our efficient training algorithm is feasible to QRBM. More significantly, based on general PI formulation, the negative phase is efficiently sampled via a state-of-the-art QMC method. From this point of view, Algorithm 1 provides a general scheme for efficiently training varieties of QBMs. Although the most advanced machine learning model can reach 98% accuracy by using a convolution neural network [46], our algorithm still presents insight into efficiently solving the problem of QML training by means of quantum statistical mechanics. In addition, there may be a more efficient approach to the positive phase updating process than just training an approximate upper bound $\widehat{\mathcal{L}}$. Therefore, mathematically, further approximation or approaching of this upper bound may improve the performance of the QRBM. Moreover, our training algorithm is capable to be extended into a fully quantum situation where the training data is accessed by a quantum memory, which will bring a quadratic speedup for our algorithm [8]. In particular, QBMs considered in this work are known to be universal for full quantum computing: with the appropriate parameter adjustments, these machines can execute any algorithm that a general-purpose quantum computer can perform.

V. CONCLUSIONS

In this paper, we have proposed an efficient algorithm for training QBMs based on the CT-QCMC method and its QMF approximation. We have considered various types of QBMs and provided the analytical expressions for their positive phase when applying the *bound-based* approximation. The proposed algorithm is capable of efficiently training various QBMs by integrating their analytical expressions of the positive phase. In particular, we have made use of the QMF approximation to further accelerate the training process of the CT-QCMC algorithm. We have tested the performance of QBMs on a small-scale artificial data set. Compared with RBM, QRBM trained by CT-QCMC and QMF-approximated algorithms presents the most advanced learning ability in terms of KL divergence. By calculating the partition function, the QMF approximation shows a slightly faster convergence compared to the CT-QCMC algorithm. To validate the efficiency of our algorithm on a large-scale data set, we have conducted the experiments on the MNIST data set and analyzed the reconstructed images of both algorithms. Moreover, we have estimated their partition function under different MC sweeps. It is found that the QMF approximation is able to train QRBM more efficiently and have a better reconstruction quality. Furthermore, the classification experiment is conducted to demonstrate the application of QRBM on supervised learning. The training and test accuracy imply that QMF-trained QRBM has a higher learning and generalization ability compared to RBM. Based on our training algorithm, varieties of QBMs can be implemented on quantum annealers as generative models to learn the MNIST data or other more complex tasks, such as to show quantum supremacy. Our work provides insight into the practical application of QML algorithms by making use of the quantum statistical method.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (Grants No. 61701302 and No. 61631014).

APPENDIX A: BOUND-BASED TRAINING FOR QBMs

The consistency of QBMs with CBMs is achieved by treating the quantum Hamiltonian H_{QBM} as a diagonal matrix. Therefore, the energy function $E(\mathbf{z})$ is obtained by measuring the quantum Hamiltonian in the basis of visible and hidden qubits, namely, $E(\mathbf{z}) = \langle \mathbf{z}|H_{\mathrm{QBM}}|\mathbf{z}\rangle$. However, when the quantum Hamiltonian has the off-diagonal terms, the ground states are not easily obtained. Neurons are in a superposition state, and their tensor products compose the ground state. From this point, the diagonal quantum Hamiltonian is a special case of nondiagonal quantum Hamiltonian. More generally, the quantum Hamiltonian of a QBM can be written as

$$H_{\mathrm{QBM}} = -H_{\mathrm{off}} - \sum_{i,j} J_{ij}\sigma_i^z\sigma_j^z - \sum_i b_i\sigma_i^z, \tag{A1}$$

where H_{off} is composed of terms that are not diagonal in the computational basis, σ^z is the Pauli-z matrix. It is the noncommutation effect, i.e., $[H_{\mathrm{QBM}}, \partial_\theta H_{\mathrm{QBM}}] \neq 0$, that leads to the quantum fluctuations in a QBM. In fact, there are different nondiagonal terms in H_{QBM}, and we present some general cases in Appendix B.

The quantum thermal distribution of visible qubits can be rewritten in terms of free energy with Eq. (12) as

$$\mathcal{P}_Q(\mathbf{z_v}) = Z_{Qv}^{-1}\exp(-\beta\mathcal{F}_Q), \tag{A2}$$

where $Z_{Qv} = \mathrm{Tr}[\exp(-\beta\mathcal{F}_Q)]$. The goal of QBM is to adjust the Hamiltonian parameters $\theta = \{J_{vh}, \Gamma_i, a_v, b_h\}$ to render the quantum thermal distribution $\mathcal{P}_Q(\mathbf{z_v})$ indistinguishable from $\mathcal{Q}(\mathbf{z_v})$.

The gradient of $\mathcal{L}_Q$ with respect to Hamiltonian parameters θ can be calculated by substituting $\mathcal{F}_Q$ into Eq. (7)

TABLE II. The various quantum Hamiltonians to construct the QBM.

Quantum Boltzmann machine
$H_{\mathrm{QBM}} = -\sum_i \Gamma_i \sigma_i^x - \sum_{i,j} J_{ij}\sigma_i^z\sigma_j^z - \sum_i b_i\sigma_i^z$
$H_{\mathrm{QBM}}^s = -\sum_i \Gamma_i \sigma_i^x - \sum_{v,i} J_{vi}\sigma_v^z\sigma_j^z - \sum_i b_i\sigma_i^z$
$H_{\mathrm{QBM}}^r = -\sum_i \Gamma_i \sigma_i^x - \sum_{v,h} J_{vh}\sigma_v^z\sigma_h^z - \sum_v a_v\sigma_v^z - \sum_v b_h\sigma_h^z$
$H_{\mathrm{QBM}}^{\mathrm{ZZXX}} = -\sum_i \Gamma_i \sigma_i^x - \sum_i b_i\sigma_i^z - \sum_{v,h} J_{vh}\big(\sigma_v^z\sigma_h^z + \sigma_v^x\sigma_h^x\big)$
$H_{\mathrm{QBM}}^{\mathrm{ZX}} = -\sum_i \Gamma_i \sigma_i^x - \sum_i b_i\sigma_i^z - \sum_{v,h} J_{vh}\big(\sigma_v^z\sigma_h^x + \sigma_v^x\sigma_h^z\big)$

given by

$$\frac{\partial \mathcal{L}_Q}{\beta\partial\theta} = \sum_{\mathbf{z_v}} \mathcal{Q}(\mathbf{z_v})\frac{\mathrm{Tr}[\Xi_{\mathbf{v}}\partial_\theta e^{-\beta H_{\mathrm{QBM}}}]}{\mathrm{Tr}[\Xi_{\mathbf{v}} e^{-\beta H_{\mathrm{QBM}}}]} - \frac{\mathrm{Tr}[\partial_\theta H_{\mathrm{QBM}} e^{-\beta H_{\mathrm{QBM}}}]}{\mathrm{Tr}[e^{-\beta H_{\mathrm{QBM}}}]}, \tag{A3}$$

where $\mathrm{Tr}[\partial_\theta e^{-\beta H_{\mathrm{QBM}}}] = -\beta \mathrm{Tr}[\partial_\theta H_{\mathrm{QBM}} e^{-\beta H_{\mathrm{QBM}}}]$ is applied. However, the exact negative log-likelihood minimization is impractical, since the positive phase of Eq. (A3) (i.e., the gradient of free energy for visible qubits) requires efficient sampling from a quantum average of a projector using Gibbs ensembles. This challenge can be mitigated by replacing the projection measurement with a positive operator value measurement (POVM) [18], with which it is possible to train these off-diagonal elements, though the prior knowledge about data distribution is necessarily provided to derive a POVM and thus is generally difficult in practice [22]. Here, we adopt the *bound-based* projection training method [16,25], by which the gradient of free energy for visible qubits can be efficiently sampled. More formally, the visible-clamped free energy $\mathcal{F}_Q^{\mathbf{v}}$ is given by

$$\mathcal{F}_Q \leqslant \mathcal{F}_Q^{\mathbf{v}} = -\beta^{-1}\ln \mathrm{Tr}[\exp{(-\beta H_{\mathbf{v}})}], \tag{A4}$$

where $H_{\mathbf{v}} = \langle \mathbf{z_v}|H_{\mathrm{QBM}}|\mathbf{z_v}\rangle$, and we have applied the inequality $\mathrm{Tr}[\Xi_{\mathbf{v}}\exp(-\beta H_{\mathrm{QBM}})] \geqslant \mathrm{Tr}[\exp{(-\beta H_{\mathbf{v}})}]$ derived with the Golden-Thompson inequality [47]. Thus, the lower bound of $\mathcal{L}_Q$ is given by

$$\mathcal{L}_Q \geqslant \widehat{\mathcal{L}}_Q = \sum_{\mathbf{z_v}} \mathcal{Q}(\mathbf{z_v}) \ln\left(\frac{\exp{(-\beta\mathcal{F}_Q^{\mathbf{v}})}}{Z_{Qv}}\right). \tag{A5}$$

We maximize the lower bound $\widehat{\mathcal{L}}_Q$ by using the gradient over θ, which is exactly Eq. (14).

APPENDIX B: GRADIENT OF FREE ENERGY IN POSITIVE PHASE

The quantum Boltzmann machine can be constructed by different types of quantum Hamiltonians. The off-diagonal component in a quantum Hamiltonian can enhance the expressive power of the QBM. For instance, taking H_{off} to be composed of tunable σ_i^x and $\sigma_i^x\sigma_j^x$ terms renders the ground state of Hamiltonian QMA complete (i.e., universal for adiabatic quantum computation). In Table II, we list the representative Hamiltonians among which the restricted ZZXX model and the restricted ZX model are believed to be QMA complete.

The restriction means that the $\sigma_v^z\sigma_h^z$ and $\sigma_v^x\sigma_h^x$ terms in the ZZXX model and the $\sigma_v^z\sigma_h^x$ and $\sigma_v^x\sigma_h^z$ terms in the ZX model share the same interaction strength J_{vh}. Note that the ZX model cannot be rewritten as the form of Eq. (A1). Here we do not consider the XY model, which will lead to the vanishing of gradient in the positive phase.

For a restricted transverse Ising Hamiltonian, let $H_{\mathrm{off}} = \sum_i \Gamma_i\sigma_i^x$. Moreover, we restrict the interactions of qubits in the visible-hidden layer. The clamped quantum restricted Boltzmann distribution with Ising model H_{QBM}^r is given by

$$\begin{aligned} H_{\mathbf{v}}^r &= -\sum_h \Gamma_h\sigma_h^x - \sum_{v,h} J_{vh} z_v \sigma_h^z - \sum_v a_v z_v - \sum_h b_h\sigma_h^z \\ &= -\sum_h \left[\Gamma_h\sigma_h^x + \left(\sum_v J_{vh} z_v + b_h\right)\sigma_h^z\right] - \sum_v a_v z_v \\ &= -\sum_h \big(\Gamma_h\sigma_h^x + \tilde{b}_h(\mathbf{z_v})\sigma_h^z\big) - \sum_v a_v z_v, \end{aligned} \tag{B1}$$

where the matrix elements of the clamped Hamiltonian with respect to the Pauli-x matrix σ^x for visible qubits (nondiagonal terms) are zeros since the nondiagonal terms satisfy $\langle z_v|\sigma_v^x|z_v\rangle = 0$. The function of Hamiltonian clamping is to render the visible qubits $|\mathbf{z_v}\rangle$ clamps to its classical data $\mathbf{z_v}$, namely, $\langle z_v|\sigma_v^z|z_v\rangle = z_v$. In the previous, we have known the eigenstate of the quantum Ising Hamiltonian H_{QBM} is the superposition of $|\mathbf{z_v}\mathbf{z_h}\rangle$. The eigenspace of the clamped Hamiltonian $H_{\mathbf{v}}$ is thus tensored by the hidden eigenstate $|\mathbf{z_h}\rangle$. As a result, the operation of Hamiltonian clamping is semi-quantum mechanical, since it only leaves the hidden qubits quantum mechanical. Due to the restricted connections between the hidden and visible qubits, the density matrix of the clamped Boltzmann distribution for QRBM is simplified and can be presented explicitly as follows:

$$\rho_{\mathrm{clamped}} = \frac{e^{-\beta H_{\mathbf{v}}^r}}{\mathrm{Tr}[e^{-\beta H_{\mathbf{v}}^r}]}, \tag{B2}$$

where

$$\begin{aligned} e^{-\beta H_{\mathbf{v}}^r} &= \exp\left\{\beta\sum_h \big[\Gamma_h\sigma_h^x + \tilde{b}_h(\mathbf{z_v})\sigma_h^z\big] + \beta\sum_v a_v\mathbf{z_v}\right\} \\ &= \exp(\beta\mathbf{a}^T\mathbf{z})\prod_h \exp\big[\beta\big(\Gamma_h\sigma_h^x + \tilde{b}_h(\mathbf{z_v})\sigma_h^z\big)\big]. \end{aligned} \tag{B3}$$

Since the hidden qubits are independent from each other, the clamped density matrix for a single hidden qubit $|h\rangle$ is thus the multiplicative of the single quantum Boltzmann distribution. Thus, the clamped Boltzmann expectation over one single hidden qubit is given by

$$\begin{aligned} \mathbb{E}_{\mathrm{clamp}}\big[\sigma_h^z\big] &= \mathrm{Tr}\big[\rho_{\mathrm{clamped}}(|h\rangle)\sigma_h^z\big] \\ &= \frac{\exp(\beta\mathbf{a}^T\mathbf{z})\mathrm{Tr}\big\{\exp\big[\beta\big(\Gamma_h\sigma_h^x + \tilde{b}_h(\mathbf{z_v})\sigma_h^z\big)\big]\sigma_h^z\big\}}{\exp(\beta\mathbf{a}^T\mathbf{z})\mathrm{Tr}\big\{\exp\big[\beta\big(\Gamma_h\sigma_h^x + \tilde{b}_h(\mathbf{z_v})\sigma_h^z\big)\big]\big\}} \\ &= \frac{2\tilde{b}_h(\mathbf{z_v})\sinh\big(\beta\sqrt{\Gamma_h^2 + \tilde{b}_h^2(\mathbf{z_v})}\big)/\sqrt{\Gamma_h^2 + \tilde{b}_h^2(\mathbf{z_v})}}{2\cosh\big(\beta\sqrt{\Gamma_h^2 + \tilde{b}_h^2(\mathbf{z_v})}\big)} \\ &= \frac{\tilde{b}_h(\mathbf{z_v})}{\sqrt{\Gamma_h^2 + \tilde{b}_h^2(\mathbf{z_v})}}\tanh\big(\beta\sqrt{\Gamma_h^2 + \tilde{b}_h^2(\mathbf{z_v})}\big). \end{aligned} \tag{B4}$$

The gradient over b_h can be calculated by Eq. (14) combined with the data distribution $\mathcal{Q}(\mathbf{z_v})$. The main difference of Eq. (B4) compared with the classical RBM in terms of the gradient updating for the bias item is the traverse field magnitude Γ_h in the QRBM. The visible qubits now are reduced to the classical case, since the clamp operation projects the qubits to the classical data. And we also assume the visible qubits are independent from each other. Thus, the expectation over one single visible qubits is $\mathbf{z_v}$, and the gradient of a_v can also be calculated exactly by Eq. (14), which is the same as with the classical case, i.e., the first term of Eq. (10b). Similarly, the expectation of $\sigma_v^z\sigma_h^z$ over the clamped Boltzmann distribution with respect to one visible and hidden qubit is given by

$$\begin{aligned}\mathbb{E}_{\text{clamp}}\big[\sigma_v^z\sigma_h^z\big] &= \text{Tr}\big[\rho_{\text{clamped}} z_v \sigma_h^z\big]\\ &= \frac{z_v \tilde{b}_h(\mathbf{z_v})}{\sqrt{\Gamma_h^2+\tilde{b}_h^2(\mathbf{z_v})}}\tanh\Big(\beta\sqrt{\Gamma_h^2+\tilde{b}_h^2(\mathbf{z_v})}\Big).\end{aligned} \tag{B5}$$

Therefore, we have finished the expectation calculation over the clamped Boltzmann distribution with respect to three different parameters for QRBM. Combined with the empirical data distribution $\mathcal{Q}(\mathbf{z_v})$, we can deduce the equation of the positive phase explicitly.

For restricted ZZXX QBM, the positive phase involves calculating the gradient of the visible-clamped free energy. The visible-clamped quantum ZZXX Hamiltonian is given by

$$H_{\mathbf{v}}^{\text{ZZXX}} = \langle \mathbf{z_v}|H_{\text{QBM}}^{\text{ZZXX}}|\mathbf{z_v}\rangle = H_{\mathbf{v}}^{r}. \tag{B6}$$

In Eq. (B6), we decompose the parameter b_i into two bias terms a_v, b_h with respect to visible and hidden qubits. The clamp operation transforms the restricted ZZXX Hamiltonian into the semiquantum one similar as with the restricted transverse Hamiltonian. Thus, the deduction for the gradient of the free energy over a clamped Boltzmann distribution is the same with the case of QRBM. That is, the expectation of $\sigma_h^z, \sigma_v^z\sigma_h^z, \sigma_v^z$ over the clamped Boltzmann distribution can be represented by Eqs. (B4) and (B5) and z_v.

As for the restricted ZX model, the clamped quantum Hamiltonian is given by

$$\begin{aligned}H_{\mathbf{v}}^{\text{ZX}} &= \langle \mathbf{z_v}|H_{\text{QBM}}^{\text{ZX}}|\mathbf{z_v}\rangle\\ &= -\sum_h \Gamma_h\sigma_h^x - \sum_v a_v z_v - \sum_h b_h\sigma_h^z - \sum_{v,h} J_{vh} z_v \sigma_h^x\\ &= -\sum_h\left[\left(\Gamma_h + \sum_v J_{vh}z_v\right)\sigma_h^x + b_h\sigma_h^z\right] - \sum_v a_v z_v\\ &= -\sum_h\big(\tilde{\gamma}_h(\mathbf{z_v})\sigma_h^x + b_h\sigma_h^z\big) - \sum_v a_v z_v.\end{aligned} \tag{B7}$$

Thus, the clamped quantum Boltzmann distribution can be represented as the density matrix

$$\rho_{\text{clamped}}^{\text{ZX}} = \frac{e^{-\beta H_{\mathbf{v}}^{\text{ZX}}}}{\text{Tr}[e^{-\beta H_{\mathbf{v}}^{\text{ZX}}}]}. \tag{B8}$$

It is necessary to remark that the independence of visible and hidden qubits which leads to the density matrix be decomposed into the multiplicative of a single hidden qubit's density matrix. The expectation of σ_h^z over the clamped ZX Boltzmann distribution is given by

$$\begin{aligned}\mathbb{E}_{\rho_{\text{clamped}}^{\text{ZX}}}\big[\sigma_h^z\big] &= \text{Tr}\big[\rho_{\text{clamped}}^{\text{ZX}}\sigma_h^z\big]\\ &= \frac{\tilde{\gamma}_h(\mathbf{z_v})}{\sqrt{b_h^2+\tilde{\gamma}_h^2(\mathbf{z_v})}}\tanh\Big(\beta\sqrt{b_h^2+\tilde{\gamma}_h^2(\mathbf{z_v})}\Big).\end{aligned} \tag{B9}$$

Similarly, the expectation of $\sigma_v^z\sigma_h^z$ over the clamped ZX quantum Boltzmann distribution is given by

$$\begin{aligned}\mathbb{E}_{\rho_{\text{clamped}}^{\text{ZX}}} &= \text{Tr}\big[\rho_{\text{clamped}}^{\text{ZX}} z_v \sigma_h^z\big]\\ &= \frac{z_v\tilde{\gamma}_h(\mathbf{z_v})}{\sqrt{b_h^2+\tilde{\gamma}_h^2(\mathbf{z_v})}}\tanh\Big(\beta\sqrt{b_h^2+\tilde{\gamma}_h^2(\mathbf{z_v})}\Big).\end{aligned} \tag{B10}$$

The expectation of σ_v^z is the same with the classical case, i.e., $\mathbb{E}_{\rho_{\text{clamped}}^{\text{ZX}}}[\sigma_v^z] = z_v$. It turns out that the positive phase of restricted ZX QBM is similar to the QRBM. Next, by combining the empirical data distribution, the positive phase of restricted ZX QBM can be obtained explicitly and efficiently. It is important to remark that the negative phase of these models, especially for the ZX and ZZXX models, is not similar to that with the QRBM. The former is proven to be QMA complete. This is a complexity class defined by a quantum computation machine, which is widely believed to be harder than NP problems. Although there is a high efficiency of CT-QCMC for Ising models, it is still hard to converge in polynomial time. By using an adiabatic algorithm in a quantum computer, these models may be solved efficiently.

APPENDIX C: CT-QCMC ALGORITHM DETAILS

Here we demonstrate the details of CT-QCMC for QRBM training. Taking $\Delta\tau \to 0$ in the context of CT-QCMC means $P \to \infty$, the consecutive spins with the same value along with the imaginary-time direction (world line), e.g., $\boldsymbol{Z}_i^k = \boldsymbol{Z}_i^{k+1} = \cdots = \boldsymbol{Z}_i^{k+S}$, form continuous *segments* $\bar{\boldsymbol{Z}}_i^{k+t}$ of length $t = S\Delta\tau$. Since we take these time limits implicitly, we will consider these world line segments as the dynamical objects in a MC algorithm and not the individual spins at discrete imaginary times. The spin-flips between two segments with different spin value is called *kinks or cuts*. As the next step we apply the scheme of the Swendsen-Wang update method [48]. In this scheme, neighboring spins in the world line of one site i, for instance, $\boldsymbol{Z}_i^k$ and $\boldsymbol{Z}_i^{k+1}$, are connected with a certain probability $p_i = 1 - \exp(-2\Delta\tau\tilde{\mathcal{B}}) = 1 - \frac{1}{2}\Delta\tau\Gamma + O(\Delta^2\tau)$. Connected neurons of visible-hidden qubits (neighboring neurons) in the space direction, for instance, $\boldsymbol{Z}_v^k$ and $\boldsymbol{Z}_h^k$ are placed with a *bond* with a probability $p_{vh} = 1 - \exp(-2\Delta\tau J_{vh}) = \frac{1}{2}\Delta\tau J_{vh} + O(\Delta^2\tau)$. These spin connection probabilities are now translated into probabilities for cutting and bonding segments.

In the context of a continuous-time limit, the probability to place cuts (*not connecting*) between neighboring pairs of one site i along with the imaginary time line is $\lim_{\Delta\tau\to 0}(1-p_i) = \lim_{\Delta\tau\to 0} 1/2\Delta\tau\Gamma$, which amounts to cutting the segments of constant spins into shorter ones by inserting cuts randomly distributed on the imaginary time line according to the uniform probability density $1/2\Gamma$. The probability to place a bond between two segments of neighboring neurons with a nonvanishing time overlap (e.g., $\tilde{\boldsymbol{Z}}_v^{t_1+t_2}$ and $\tilde{\boldsymbol{Z}}_h^{t_3+t_4}$ with overlapping time $[t_1,t_2]\cap[t_3,t_4]\neq 0$) is $\lim_{\Delta\tau\to 0}\frac{1}{2}\Delta\tau J_{vh}$, which means that these bonds bind two overlapping neighboring segments with density $1/2J_{vh}$ if they have the same spin value. The generation of the uniformly and randomly distributed cuts or bond times with a density forms a homogeneous Poisson process.

In Sec. III, we demonstrate the CT-QCMC algorithm for solving the expectation of observable. When using MCMC method, we need to compute accept probability between the updated configuration and the original configuration such as to determine whether the updated configuration is accepted or rejected according to the Metropolis rule, which is given by

$$\mathcal{R} = \exp\left(-\int_0^\beta \Delta\mathcal{E}(z(\tau))d\tau\right), \tag{C1}$$

where $\Delta\mathcal{E}$ is the energy difference between the original and updated configuration. Starting form an arbitrary distribution, the Markov chain will converge exponentially to a stationary distribution $\tilde{\boldsymbol{Z}}\cdot\mu(\tilde{\boldsymbol{Z}})$ after quantities of MC sweeps according to the Metropolis updating rule. Finally, the expectation of observable Eq. (25) will be estimated by the average of R MC sweeps,

$$\mathbb{E}_\rho[O] \approx \frac{1}{R}\sum_{r=1}^{R}\frac{1}{\beta}\int_0^\beta O^{(r)}(z(\tau))d\tau, \tag{C2}$$

where $O^{(r)}(z(\tau))$ denotes the rth observable estimation by using the rth sample. According to the central limit theorem, as long as the number of MC sweeps is large enough the estimate of Eq. (C2) will be normally distributed around the exact value of Eq. (25) with statistical error (variance) given by

$$\Delta O = \sqrt{\frac{\text{Var}[O](2\tau_a+1)}{R}}, \tag{C3}$$

where Var$[O]$ is the variance of observable O, and the integrated autocorrelation time τ_a is a measure of the autocorrelations of the sample sequences $O^{(r)}(z(\tau))$. In practice, the autocorrelation time will be burnt out in CT-QCMC to assure high independence of the successive samples. In general, the MC methods can solve the observable expectation with polynomial time as long as τ_a increases not faster than polynomially with the number of qubits N. The detailed CT-QCMC algorithm is demonstrated in Appendix C.

Here we present the detailed realization framework of this algorithm as follows:

Algorithm 2. CT-QCMC: Continuous-time quantum cluster Monte Carlo algorithm for QBM.

```
Input: $N, \beta, \Gamma, n$
Output: Expectation estimation of observables $\sigma_v^z\sigma_h^z, \sigma_v^z, \sigma_h^z$.
function METRO $\tilde{\boldsymbol{Z}}_{\rm old}, \tilde{\boldsymbol{Z}}_{\rm upd}, \beta$
    Calculate $\Delta\mathcal{E}$ by Eq. (23) or $\Delta\mathcal{E}_{\rm QMF}$ by Eq. (29)
    $u \leftarrow$ uniform random sampling from [0,1]
    if $\mathcal{R} \geqslant u$ then                ▷ accept probability satisfied
        return$\tilde{\boldsymbol{Z}}_{\rm upd}$
    else
        return $\tilde{\boldsymbol{Z}}_{\rm old}$
    end if
end function

function CT-QC $\tilde{\boldsymbol{Z}}_{\rm old}, \beta, n$
    for every qubit do
        Remove all redundant cuts
        Generate new cuts by selecting the cut times $\tau\in[0,\beta)$ uniformly and randomly with density $1/2\Gamma$
    end for
    for every pair of visible-hidden qubits do
        Generate potential bond times $\tau\in[0,\beta)$ uniformly and randomly with density $1/2J_{vh}$
        for each bond time $\tau$ do
            if they have the same spin value at $\tau$ then
                Insert a bond for the two segments at $\tau$
            end if
        end for
    end for
    Identify clusters of bonded segments
    for every cluster do
        Flip the spin on all the segments in the cluster with probability $1/2$
    end for
return the updated configuration $\tilde{\boldsymbol{Z}}_{\rm upd}$
end function

Randomly initialize configuration: $\tilde{\boldsymbol{Z}}$
$\tilde{\boldsymbol{Z}}_{\rm old} \leftarrow \tilde{\boldsymbol{Z}}$
for $r$ from 1 to $R$ do
    Configuration updating: $\tilde{\boldsymbol{Z}}_{\rm upd} \leftarrow$ CT-QC($\tilde{\boldsymbol{Z}}_{\rm old}, \beta, n$)
    Metropolis updating: $\tilde{\boldsymbol{Z}}_{\rm upd}^{(r)} \leftarrow$ METRO($\tilde{\boldsymbol{Z}}_{\rm old}, \tilde{\boldsymbol{Z}}_{\rm upd}, \beta$)
    $\tilde{\boldsymbol{Z}}_{\rm old} \leftarrow \tilde{\boldsymbol{Z}}_{\rm upd}^{(r)}$
end for
Observable Expectation approximated by using Eq. (C2)
```

[1] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, Deep learning for computer vision: A brief review, Comput. Intell. Neurosci. **2018**, 7068349 (2018).

[2] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).

[3] R. Salakhutdinov, A. Mnih, and G. Hinton, Restricted Boltzmann machines for collaborative filtering, in *Proceedings of*

the 24th International Conference on Machine Learning (ACM, New York, 2007), pp. 791–798.

[4] M. J. Hartmann and G. Carleo, Neural-Network Approach to Dissipative Quantum Many-Body Dynamics, Phys. Rev. Lett. **122**, 250502 (2019).

[5] F. Vicentini, A. Biella, N. Regnault, and C. Ciuti, Variational Neural-Network Ansatz for Steady States in Open Quantum Systems, Phys. Rev. Lett. **122**, 250503 (2019).

[6] T. Xiao, J. Huang, J. Fan, and G. Zeng, Continuous-variable quantum phase estimation based on machine learning, Sci. Rep. **9**, 1 (2019).

[7] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, Nat. Phys. **14**, 447 (2018).

[8] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature (London) **549**, 195 (2017).

[9] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, Quantum machine learning: A classical perspective, Proc. R. Soc. London A **474**, 20170551 (2018).

[10] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum Support Vector Machine for Big Data Classification, Phys. Rev. Lett. **113**, 130503 (2014).

[11] N. Wiebe, D. Braun, and S. Lloyd, Quantum Algorithm for Data Fitting, Phys. Rev. Lett. **109**, 050505 (2012).

[12] S. Lloyd and C. Weedbrook, Quantum Generative Adversarial Learning, Phys. Rev. Lett. **121**, 040502 (2018).

[13] P.-L. Dallaire-Demers and N. Killoran, Quantum generative adversarial networks, Phys. Rev. A **98**, 012324 (2018).

[14] M. W. Johnson, M. H. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk *et al.*, Quantum annealing with manufactured spins, Nature (London) **473**, 194 (2011).

[15] S. Aaronson and A. Arkhipov, The computational complexity of linear optics, in *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing* (ACM, New York, 2011), pp. 333–342.

[16] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, Quantum Boltzmann Machine, Phys. Rev. X **8**, 021050 (2018).

[17] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, A learning algorithm for Boltzmann machines, Cognit. Sci. **9**, 147 (1985).

[18] M. Kieferová and N. Wiebe, Tomography and generative training with quantum Boltzmann machines, Phys. Rev. A **96**, 062327 (2017).

[19] N. Wiebe and L. Wossnig, Generative training of quantum Boltzmann machines with hidden units, arXiv:1905.09902.

[20] T. Puškarov and A. C. Cubero, Machine learning algorithms based on generalized Gibbs ensembles, J. Stat. Mech. (2018) 103102.

[21] M. Denil and N. de Freitas, Toward the implementation of a quantum RBM, in *Proceedings of the NIPS 2011 Workshop on Deep Learning and Unsupervised Feature Learning* (2011).

[22] E. R. Anschuetz and Y. Cao, Realizing quantum Boltzmann machines through eigenstate thermalization, arXiv:1903.01359 (2019).

[23] M. Rigol, V. Dunjko, and M. Olshanii, Thermalization and its mechanism for generic isolated quantum systems, Nature (London) **452**, 854 (2008).

[24] J. D. Biamonte and P. J. Love, Realizable Hamiltonians for universal adiabatic quantum computers, Phys. Rev. A **78**, 012352 (2008).

[25] A. Khoshaman, W. Vinci, B. Denis, E. Andriyash, and M. H. Amin, Quantum variational autoencoder, Quantum Sci. Technol. **4**, 014001 (2018).

[26] D. Malpetti and T. Roscilde, Quantum mean-field approximation for lattice quantum models: Truncating quantum correlations and retaining classical ones, Phys. Rev. B **95**, 075112 (2017).

[27] H. Larochelle, M. Mandel, R. Pascanu, and Y. Bengio, Learning algorithms for the classification restricted Boltzmann machine, J. Mach. Learn. Res. **13**, 643 (2012).

[28] W. Byrne, Alternating minimization and Boltzmann machine learning, IEEE Trans. Neural Networks **3**, 612 (1992).

[29] G. E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. **14**, 1771 (2002).

[30] Here, "efficiently computable" refers to the program can be executed in polynomial time on a classical Turing machine, namely **P** class. **NP-hard** means the program requires exponential complexity to be executed.

[31] F. Krzakala, A. Rosso, G. Semerjian, and F. Zamponi, Path-integral representation for quantum spin models: Application to the quantum cavity method and Monte Carlo simulations, Phys. Rev. B **78**, 134428 (2008).

[32] T. H. Hsieh, From d-dimensional quantum to d+1-dimensional classical systems, Student review, (2) 1 (2016).

[33] R. M. Fye, New results on Trotter-like approximations, Phys. Rev. B **33**, 6271 (1986).

[34] E. Gull, A. J. Millis, A. I. Lichtenstein, A. N. Rubtsov, M. Troyer, and P. Werner, Continuous-time Monte Carlo methods for quantum impurity models, Rev. Mod. Phys. **83**, 349 (2011).

[35] A. Winter, H. Rieger, M. Vojta, and R. Bulla, Quantum Phase Transition in the Sub-Ohmic Spin-Boson Model: Quantum Monte Carlo Study with a Continuous Imaginary Time Cluster Algorithm, Phys. Rev. Lett. **102**, 030601 (2009).

[36] H. Rieger and N. Kawashima, Application of a continuous time cluster algorithm to the two-dimensional random quantum Ising ferromagnet, Eur. Phys. J. B **9**, 233 (1999).

[37] A. Juan and E. Vidal, Bernoulli mixture models for binary images, in *Proceedings of the 17th International Conference on Pattern Recognition 2004, ICPR 2004* (IEEE, New York, 2004), Vol. 3, pp. 367–370.

[38] R. M. Neal, Annealed importance sampling, Stat. Comput. **11**, 125 (2001).

[39] R. Salakhutdinov and I. Murray, On the quantitative analysis of deep belief networks, in *Proceedings of the 25th International Conference on Machine Learning* (ACM, New York, 2008), pp. 872–879.

[40] S. Wold, K. Esbensen, and P. Geladi, Principal component analysis, Chemom. Intell. Lab. Syst. **2**, 37 (1987).

[41] L. v. d. Maaten and G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. **9**, 2579 (2008).

[42] J. T. Rolfe, Discrete variational autoencoders, arXiv:1609.02200.

[43] M. Weigel, L. Y. Barash, M. Borovský, W. Janke, and L. N. Shchur, Population annealing: Massively parallel simulations in statistical physics, J. Phys.: Conf. Ser. **921**, 012017 (2017).

[44] G. Desjardins, Y. Bengio, and A. C. Courville, On tracking the partition function, in *Advances in Neural Information Processing Systems, NIPS 2011, Granada, Spain*, edited by J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger (Neural Information Processing Systems Foundation, Inc., 2011), pp. 2501–2509.

[45] S. Ioffe and C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv:1502.03167.

[46] X.-X. Niu and C. Y. Suen, A novel hybrid CNN–SVM classifier for recognizing handwritten digits, Pattern Recognit. **45**, 1318 (2012).

[47] T. Ando and F. Hiai, Log majorization and complementary Golden-Thompson type inequalities, Linear Algebra Appl. **197**, 113 (1994).

[48] R. H. Swendsen and J.-S. Wang, Nonuniversal Critical Dynamics in Monte Carlo Simulations, Phys. Rev. Lett. **58**, 86 (1987).