Quantum Bayes classifiers and their application in image classification

Ming-Ming Wang[*] and Xiao-Ying Zhang

Shaanxi Key Laboratory of Clothing Intelligence, School of Computer Science, Xi'an Polytechnic University, Xi'an 710048, China

(Received 29 March 2024; revised 7 June 2024; accepted 24 June 2024; published 12 July 2024)

Bayesian networks are powerful tools for probabilistic analysis and have been widely used in machine learning and data science. Unlike the time-consuming parameter training process of neural networks, Bayes classifiers constructed on Bayesian networks can make decisions based solely on statistical data from samples. In this paper we focus on constructing quantum Bayes classifiers (QBCs). We design both a naïve QBC and three seminaïve QBCs (SN-QBCs). These QBCs are then applied to image classification tasks. To reduce computational complexity, we design a local feature sampling method to extract a limited number of feature attributes from an image. These attributes serve as nodes of the Bayesian networks to generate the QBCs. We simulate these QBCs on the MindQuantum platform and evaluate their performance on the MNIST and Fashion-MNIST data sets. Our results demonstrate that these QBCs achieve good classification accuracies even with a limited number of attributes. The classification accuracies of QBCs on the MNIST data set surpass those of classical Bayesian networks and quantum neural networks that utilize all available feature attributes. Additionally, we simulate these QBCs in a quantum noise environment.

DOI: 10.1103/PhysRevA.110.012433

I. INTRODUCTION

As a powerful tool for studying causal relationships between variables and inferring the impact of variable states on outcomes, Bayesian networks are widely used in machine learning and data science, including Monte Carlo analysis [1], reliability and risk analysis [2], health monitoring [3], health care [4], biomedical systems [5], etc. The size of a Bayesian network depends on the number of nodes and their dependencies [6]. Learning and inference in complex networks can be challenging, especially when dealing with large-scale Bayesian networks, which have been proven to be an NP-hard problem [7]. The emergence of quantum computing offers a new solution to this challenge.

In recent years, many quantum algorithms have demonstrated quantum supremacy for achieving certain accelerations over their classical counterparts. For example, Shor's algorithm [8] achieves exponential acceleration in solving the large number factorization problem. Grover's algorithm [9] achieves quadratic acceleration in searching the unstructured data. In addition, quantum algorithms based on classical machine learning, such as quantum support vector machine [10] and quantum K-nearest neighbor [11], have also demonstrated quantum accelerations. Image classification is a fundamental problem in computer vision. With the advancement of quantum machine learning, several quantum classifiers have been developed for image classification, including quantum convolutional neural networks [12–15], quantum K-nearest-neighbor algorithm [11], quantum ensemble methods [16–20], quantum capsule networks [21], quantum decision trees [22,23], etc. Recent progress of quantum classifiers could be found in [24].

Quantum Bayesian networks (QBNs) were introduced in 1995 as a simulation of classical ones [25,26]. In 2013, Ozols *et al.* proposed a quantum version of the rejection sampling algorithm called quantum rejection sampling for Bayesian inference [27]. In 2014 Low *et al.* studied quantum circuit representations of Bayesian networks with discrete nodes and designed the circuit for implementing quantum rejection sampling [28]. In 2015 Wiebe and Granade addressed a semiclassical algorithm of learning for quantum Bayesian inference and showed that the algorithm is polynomially faster than its classical anolog [29]. In 2016 Moreira and Wichert proposed a quantum-like Bayesian network that uses amplitudes to represent marginal and conditional probabilities [30]. In 2019 Woerner and Egger developed a quantum algorithm [31] for risk analysis using the principles of amplitude amplification and estimation. Their algorithm can provide a quadratic speed-up compared to classical Monte Carlo methods. In 2023 Escrig *et al.* proposed a quantum algorithm based on quantum walks for quantum Bayesian estimation of gravitational waves parameters from black holes [32]. Recently, Borujeni *et al.* proposed a quantum circuit representation of Bayesian networks [33]. They designed quantum Bayesian networks for specific problems, such as stock prediction and liquidity risk assessment. Fathallah *et al.* further proposed an optimized version of quantum circuit for improving the quantum representation of Bayesian networks [34].

Currently, there is a lack of research on quantum Bayes classifiers (QBCs) building on Bayesian networks for solving image classification problems. Different from the parameters learning mode of neural networks, a Bayes classifier makes classification decisions based only on sample features, without the tedious training process, resulting in lower computational complexity, faster speed, and less resource consumption.

[*] Contact author: bluess1982@126.com

In this paper we study the construction of QBCs for solving image classification problems. Specifically, we design a naïve [35,36] QBC and three seminaïve QBCs (SN-QBCs), i.e., the SN-QBC based on an SPODE network [37,38] with the attribute node in the center of an image as the superfather, the SN-QBC based on TAN network [37,38], and the SN-QBC based on symmetric relationship of attribute nodes in an image. These QBCs are then applied to image classification tasks. We simulate these QBCs on the MindQuantum platform [39] and evaluate their performance on the MNIST [40] and Fashion-MNIST [41] data sets.

The main contributions of this paper are as follows:

(1) Designing a naïve-QBC and three SN-QBCs for solving image classification problems.

(2) Introducing a local sample method to extract a small number of binary attribute nodes, simplifying the construction of quantum circuits for QBCs.

(3) Conducting noise analysis on QBCs for image classification, considering four common types of noise encountered in practice.

This paper is organized as follows. Section 2 introduces the basic concepts of Bayes classifiers. Section 3 discusses the constructions of QBCs. Section 4 presents the image classification algorithm based on QBCs. Section 5 demonstrates the simulation results of QBCs for image classification on the MNIST and Fashion-MNIST data sets. Section 6 analyzes the computation complex of QBCs, explores the impact of sampling block size on QBCs' performance, and simulates QBCs under four common types of noise. Section 7 further discusses and concludes the paper.

II. BAYES CLASSIFIER

A Bayes classifier is a statistical classifier based on Bayes' theorem. It considers selecting the optimal category label based on probabilities and misclassification losses, assuming that all relevant probabilities are known. Suppose the feature of a sample data is $X = \{x_1, x_2, \ldots, x_i, \ldots, x_n\}$ where x_i is an *attribute* of X, and the set of class labels is $Y = \{y_1, y_2, \ldots, y_i, \ldots, y_N\}$. Based on the *posterior probability* $P(y_i|X)$, the expected loss of classifying the sample with feature X as y_i is defined as [38]

$$R(y_i|X) = \sum_{j=1}^{N} \lambda_{ij} P(y_j|X), \tag{1}$$

where λ_{ij} is the misclassification loss. The Bayes classifier attempts to correctly classify new samples with minimal misclassification loss based on the distribution pattern of existing samples. The optimal Bayes classifier can be denoted as

$$D(X) = \arg\min_{y\in Y} R(y|X). \tag{2}$$

For a specific problem to minimize the misclassification rate, λ_{ij} can be written as

$$\lambda_{ij} = \begin{cases} 0, & \text{if } i = j; \\ 1, & \text{otherwise.} \end{cases} \tag{3}$$

Therefore, the optimal Bayes classifier can be rewritten as

$$D(X) = \arg\max_{y\in Y} P(y|X). \tag{4}$$

That is, the optimal Bayes classifier selects the class that maximizes the posterior probability $P(y|X)$ given the sample with feature X.

Obtaining an accurate posterior probability $P(y|X)$ is critical for a Bayes classifier, but this is often challenging in reality. In the probability framework, the posterior probability $P(y|X)$ can be estimated based on a finite number of training sample. According to Bayes' theorem [38], the posterior probability $P(y|X)$ can be written as

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}, \tag{5}$$

where $P(X)$ is the evidence factor used for normalization, $P(y)$ is the *class-prior probability*, and $P(X|y)$ is the *class-conditional probability* of the sample X with respect to y. According to the law of large numbers [42], when there are a sufficient number of independently and identically distributed samples, $P(y)$ can be estimated by the frequency of each class that appears in the training set. As for the class-conditional probability $P(X|y)$, it involves combinations of all attributes in X. Assuming each attribute x_i has d possible values, $P(X|y)$ will have $N \times d^n$ possible values.

A. Naïve Bayes classifier

Clearly, it is difficult to obtain the *class-conditional probabilities* directly from a limited number of training sample, as it will result in the problem of combinatorial explosion in calculation, which becomes more severe with the increase of attributes. The naïve Bayes classifier [35,36] is based on the assumption of "independence," which assumes that each attribute independently affects the classification result, as shown in Fig. 1(a). In this case, Eq. (5) can be rewritten as

$$P(y|X) = \frac{P(y)\prod_i^n P(x_i|y)}{P(X)}, \tag{6}$$

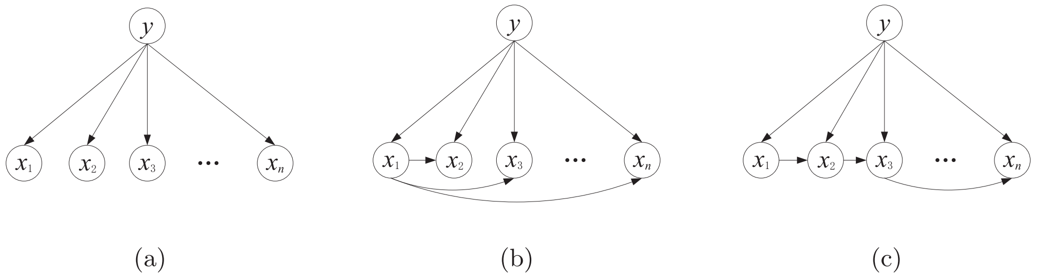
where x_i is the ith attribute of X. Since all $P(X)$ are the same, the naïve Bayes classifier can be represented as

$$D_{\text{nb}} = \arg\max_{y\in Y} P(y) \prod_i^n P(x_i|y). \tag{7}$$

B. Seminaïve Bayes classifier

The premise of the naïve Bayes classifier is that all attributes satisfy the assumption of independence, but this is often not the case in practical applications. Therefore, the learning method of the seminaïve Bayes classifier [37,38] has emerged, which considers stronger dependencies between attributes while it avoids the problem of combinatorial explosion caused by considering the joint probability distribution of all attributes. The *one-dependent estimator* (ODE) is the most common strategy for the seminaïve Bayes classifier, which assumes that each attribute depends on only at most one other attribute besides the label y,

$$P(y|X) \propto P(y) \prod_i^n P(x_i|y, pa_i), \tag{8}$$



(a) (b) (c)

FIG. 1. Bayesian networks with different attributes dependencies: (a) naïve, (b) SPODE, (c) TAN.

where pa_i is the dependent attribute (or parent attribute) of x_i. The key problem of the seminaïve Bayes classifier is how to determine the parent attribute of x_i.

A typical approach is to assume that all attributes depend on a single attribute, referred to as the Super-Parent ODE (SPODE), whose dependency relationship is illustrated in Fig. 1(b). An alternative method is the Tree Augmented Naïve Bayes (TAN), which calculates the conditional mutual information between any pair of attributes and constructs a maximum weighted tree [38,43] based on the attribute dependencies, as depicted in Fig. 1(c).

III. QUANTUM BAYES CLASSIFIERS

Unlike classical bits that can represent only either 0 or 1, a quantum bit (qubit) can represent both 0 and 1 simultaneously, i.e., quantum superposition. A single-qubit state can be represented as $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $|0\rangle$ and $|1\rangle$ are the basis states of the single qubit, and α and β are the amplitudes that satisfy the normalization condition $|\alpha|^2 + |\beta|^2 = 1$. When $|\varphi\rangle$ is measured in the computational basis $\{|0\rangle, |1\rangle\}$, the state will collapse into basis states $|0\rangle$ or $|1\rangle$ with the probability $|\alpha|^2$ or $|\beta|^2$, respectively. In the quantum gate computing model [44], quantum gates are used to represent unitary operations acting on qubits. Quantum gates can be divided into single-qubit gates and multiqubit gates, and any multiqubit gate can be decomposed into a set of universal quantum gates [44].

In this paper a quantum circuit for a QBC is constructed using single-qubit gates R_y and X, and multiqubit controlled gates C^nR_y. The X-gate is a flip gate that flips $|0\rangle$ to $|1\rangle$ or $|1\rangle$ to $|0\rangle$. Its matrix form is

$$X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{9}$$

R_y is a single-qubit rotation gate which has the form

$$R_y(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}, \tag{10}$$

where θ is the rotation angle. When acting on $|0\rangle$, the R_y gate generates the following superposition state:

$$R_y(\theta)|0\rangle = \cos\frac{\theta}{2}|0\rangle + \sin\frac{\theta}{2}|1\rangle.$$

C^nR_y is a multiqubit controlled rotation gate, where n represents the number of control qubits. When the control qubits are all in $|1\rangle$, the R_y rotation operation is performed on the target qubit. The two-qubit controlled rotation gate CR_y with $n = 1$ is represented as

$$CR_y(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ 0 & 0 & \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}. \tag{11}$$

The target qubit $|\varphi\rangle$ will undergo a $R_y(\theta)$ rotation if control qubits are all in $|1\rangle$,

$$C^nR_y(\theta)|1\rangle_c^{\otimes n}|\varphi\rangle_t = |1\rangle_c^{\otimes n}R_y(\theta)|\varphi\rangle_t,$$

where the subscript c represents the control qubits and t stands for the target qubit.

A. Naïve-QBC

A QBC uses Bayes' rule to perform classification tasks within the framework of quantum computing. A single-qubit can be used to represent a two-state node in a Bayesian network, and then a superposition quantum state can be used to represent the probability of different label values under various combinations of attributes in the Bayesian network, i.e., $P(y|X)$.

In a naïve Bayesian network, all attributes are independent of each other, and they depend only on the label node. Let n attribute nodes be $x_1, x_2, \ldots, x_i, \ldots, x_n$, and each attribute depends only on the label y, with $y, x_i \in \{0, 1\}$. The quantum circuit of the naïve-QBC is composed of $n+1$ qubits, corresponding to the label y and n attribute nodes x_1 to x_n, with each qubit represents a two-state node. The naïve-QBC is constructed as follows:

(1) All $n+1$ qubits are initialized to $|0\rangle$.

(2) For the label node y, the *class-prior probability* $P(y = 0)$ can be obtained by statistically counting the training set. After that, one can encode the class-prior probabilities by $R_y(\cdot)$ operation. Here arccos() is used for encoding. That is,

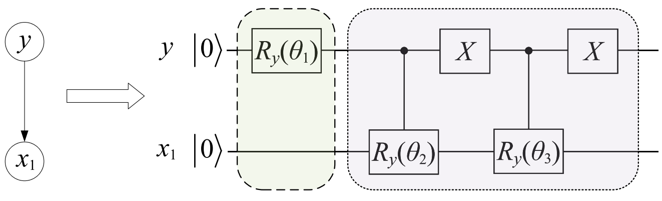


FIG. 2. Quantum circuit for $y \rightarrow x_1$, i.e., the naïve-QBC with only one attribute ($n = 1$).

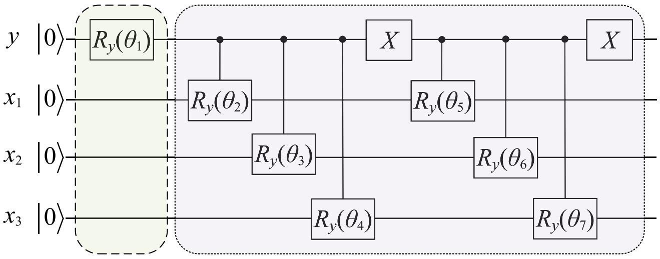


FIG. 3. Quantum circuit of the naïve-QBC with three attributes.

let

$$\cos^2\frac{\theta}{2} := P(y=0). \tag{12}$$

One can obtain

$$\theta = 2\arccos\sqrt{P(y=0)}. \tag{13}$$

This achieves quantum encoding of the *class-prior probabilities*. Note that the encoding function is equivalent to arctan() used in Ref. [33]. To simplify the representation in the following content, let

$$f(P) := 2\arccos\sqrt{P}. \tag{14}$$

(3) For each attribute node x_i, one can get the *class-conditional probabilities* by statistically calculating $P(x_i = 0|y=1)$ and $P(x_i=0|y=0)$. Then these probabilities are encoded by $CR_y(\cdot)$, where the controlled rotation angles are set as $\theta_{i+1} = f(P(x_i=0|y=1))$ and $\theta_{i+n+1} = f(P(x_i = 0|y = 0))$, respectively.

Take the naïve Bayesian network in Fig. 1(a) as an example. Suppose that these are one label node y and one attribute node x_1 ($n=1$), and each node has only two values, either 0 or 1. By statistically counting the *class-prior probability* $P(y=0)$ and the *class-conditional probabilities* $P(x_1 = 0|y=1)$ and $P(x_1=0|y=0)$, one can construct a quantum circuit as depicted in Fig. 2. In this case the output of the circuit is

$$\sum_{i,j=\{0,1\}}\sqrt{P(y=i)}\sqrt{P(x_1=j|y=i)}|ij\rangle. \tag{15}$$

Note that the probabilities of different label values and attribute values are encoded in the amplitude of the output state, which has the same form as Eq. (7). That is, the quantum circuit implements the naïve-QBC for features with only one attribute.

By continuously adding new attribute nodes to the quantum circuit and establishing controlled rotations between parent nodes and child nodes, one can construct QBCs based on different dependency relationships. Another example is a Bayesian network with multiple attributes. That is, the naïve QBC with $n=3$ attributes is shown in Fig. 3.

In the prediction stage, for a given feature value X^*, one needs only to obtain the probabilities of the basis state $|y=i, X=X^*\rangle$ by measuring the output state of the naïve QBC. For a binary classification problem, one obtains the values of $P(y=0, X=X^*)$ and $P(y=1, X=X^*)$, and chooses the y value with a higher probability as the classification outcome of the QBC.

B. SPODE-QBC

For the SPODE Bayesian network, n attribute nodes are $x_1, x_2, \ldots, x_n$. Without loss of generality, assume that x_1 is the super-parent node, as shown in Fig. 1(b). The quantum circuit of the SN-QBC based on the SPODE structure (SPODE-QBC) consists of $n+1$ qubits, corresponding to y and x_1 to x_n. The construction of SPODE-QBC is as follows:

(1) All $n+1$ qubits are initialized to $|0\rangle$.

(2) For the label node y, the *class-prior probability* $P(y=0)$ is counted and encoded by $R_y(\cdot)$, where the rotation angle is set as $\theta_1 = f(P(y=0))$.

(3) For the super-parent x_1, one needs to count the *class-conditional probabilities* $P(x_1=0|y=0)$ and $P(x_1=0|y=1)$ and encode these probabilities by using X and $CR_y(\cdot)$, with the controlled rotation angles set as $\theta_2 = f(P(x_1=0|y=1))$ and $\theta_3 = f(P(x_1=0|y=0))$, respectively.

(4) For the remaining attributes x_2 to x_n, since each node has two parent nodes, four $C^2R_y(\cdot)$ are used to encode the corresponding *class-conditional probabilities* $P(x_j=0|y,x_1)$, where the controlled rotation angles are set as $f(P(x_j = 0|y,x_1))$ given the values of y and x_1. That is, when the control bits are $yx_1 = 00, 01, 10, 11$, the corresponding controlled rotation angles are set as $f(P(x_j=0|y=0,x_1=0))$, $f(P(x_j=0|y=0,x_1=1))$, $f(P(x_j=0|y=1,x_1=0))$, and $f(P(x_j=0|y=1,x_1=1))$, respectively.

For example, for a SPODE-structured Bayesian network with three attributes shown in Fig. 1(b), the quantum circuit of the SPODE-QBC is depicted in Fig. 4. Specifically, the class-conditional probabilities of x_2, x_3, and x_4 are encoded as $\theta_4 = f(P(x_2=0|yx_1=11))$, $\theta_5 = f(P(x_3=0|yx_1=11))$, $\theta_6 = f(P(x_2=0|yx_1=10))$, $\theta_7 = f(P(x_3=0|yx_1=10))$, $\theta_8 = f(P(x_2=0|yx_1=00))$, $\theta_9 = f(P(x_3=0|yx_1=00))$, $\theta_{10} = f(P(x_2=0|yx_1=01))$, $\theta_{11} = f(P(x_3=0|yx_1=01))$. The output of the SPODE circuit is

$$\sum_{i,j,k,l=\{0,1\}}\sqrt{P(y=i)P(x_1=j|y=i)P(x_2=k|y=i,x_1=j)P(x_3=l|y=i,x_1=j)}|ijkl\rangle. \tag{16}$$

C. TAN-QBC

For a SN-QBC based on the TAN structure, one needs to obtain the TAN structure Bayesian network first. The TAN structure Bayes classifier is generated based on the maximum-weighted spanning tree algorithm [38,43], which includes the following steps [38]:

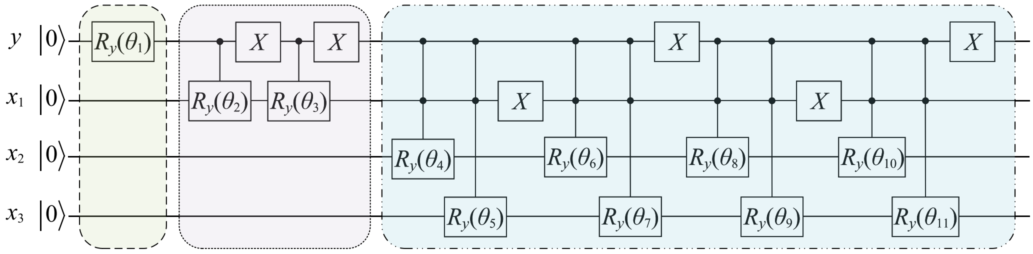


FIG. 4. Quantum circuit of the SPODE-QBC with three attributes and x_1 as the superparent node.

(1) Calculate the conditional mutual information between every two nodes x_i and x_j using the following equation:

$$I(x_i, x_j|y) := \sum_{x_i,x_j,c\in Y} P(x_i, x_j|c) \log_2 \frac{P(x_i, x_j|c)}{P(x_i|c)P(x_j|c)}. \quad (17)$$

(2) Build a complete graph among nodes and set $I(x_i, x_j|y)$ as the weight between x_i and x_j.

(3) Construct the maximum-weighted spanning tree of the complete graph and set the direction of each edge outward from the root.

(4) Add the label node y and directed edges from y to each node.

For a Bayesian network with a TAN structure, the quantum circuit of the TAN-QBC consists of $n + 1$ qubits, corresponding to y and x_1 to x_n. The construction of TAN-QBC is as follows:

(1) All qubits are initialized to $|0\rangle$.

(2) The *class-prior probability* $P(y = 0)$ is encoded by $R_y(f(P(y = 0)))$.

(3) Starting from the root node of the feature spanning tree, the *class-conditional probabilities* of each node are encoded layer by layer. For attribute x_j, it has at most two parent nodes, y and x_{parent_j}, where x_{parent_j} is the parent node of x_j on the upper layer (note that the root node of the feature spanning tree has only one parent y). The *class-conditional probabilities* $P(x_j|y, x_{\text{parent}_j})$ are encoded into the circuit as the rotation angles of four $C^2R_y(\cdot)$ gates.

A simple example of a Bayesian network with a TAN structure is given on the left side of Fig. 5. The circuit of the TAN-QBC based on the TAN structure is presented on the right side of Fig. 5. The rotation angles of the label node y and the root node x_1 of the spanning tree are set as the naïve-QBC shown in Sec. III A. The rotation angles of C^2R_y gates acting on x_3 are set as $\theta_4 = f(P(x_3 = 0|yx_1 = 11))$, $\theta_5 = f(P(x_3 = 0|yx_1 = 10))$, $\theta_6 = f(P(x_3 = 0|yx_1 = 00))$, and $\theta_7 = f(P(x_3 = 0|yx_1 = 01))$, respectively. The rotation angles of C^2R_y gates acting on x_2 are set as $\theta_8 = f(P(x_2 = 0|yx_3 = 01))$, $\theta_9 = f(P(x_2 = 0|yx_3 = 00))$, $\theta_{10} = f(P(x_2 = 0|yx_3 = 10))$, and $\theta_{11} = f(P(x_2 = 0|yx_3 = 11))$, respectively. The output of the TAN structure SN-QBC is

$$\sum_{i,j,k,l=\{0,1\}} \sqrt{P(y = i)P(x_1 = j|y = i)P(x_3 = k|y = i, x_1 = j)P(x_2 = l|y = i, x_3 = k)}|ijkl\rangle. \quad (18)$$

D. SN-QBC based on the symmetric relationship of image attributes

Both the naïve Bayes classifier and the seminaïve Bayes classifier consider only the dependencies between a small number of attribute nodes. Note that there are symmetric relationships among the sampled feature attributes in some images, such as the digits "6" and "9" shown in Fig. 6, where there exist symmetric relationships between attributes x_2 and x_5, and x_3 and x_4. These symmetric relationships of image attributes can be used to build Bayesian networks, which can then be used to establish corresponding SN-QBCs. The method for constructing a Bayesian network based on the symmetric relationships of image attributes is as follows:

(1) Establish a naïve Bayesian network as shown in Fig. 1(a).

(2) Consider the symmetric relationship of attributes in the sample images and add a directed edge between each pair of attributes that are symmetrical to each other.

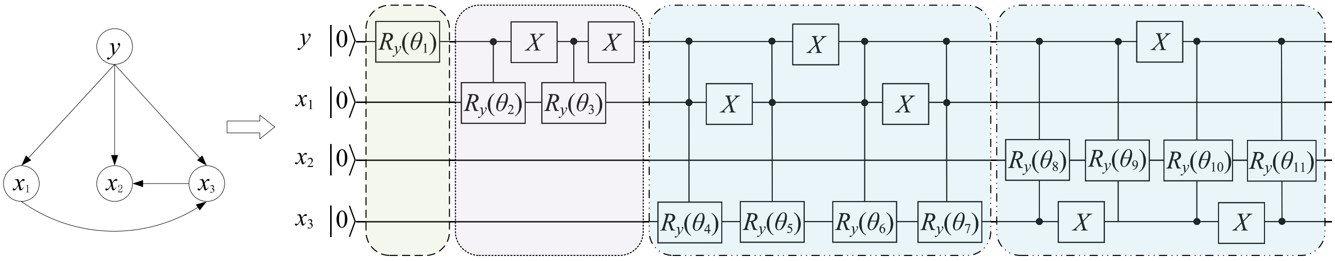


FIG. 5. Quantum circuit of the TAN-QBC based on the TAN structure Bayesian network.

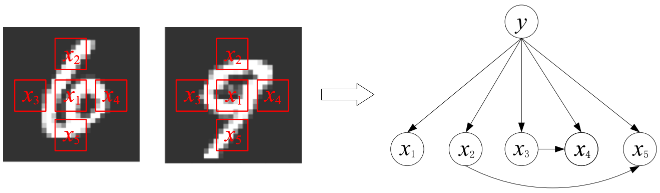


FIG. 6. Bayesian network based on symmetric relationships of feature's attributes.

For example, for an image data set of digits "6" and "9," a Bayesian network can be established as shown in Fig. 6.

For the Bayesian network with symmetric relationships between attributes, in the case of not considering the label node y, the network will consist of several independent trees. In this case, one can construct the corresponding symmetric-QBC using the method for the TAN-QBC in Sec. III C; that is,

(1) Each node is initialized to $|0\rangle$.

(2) The *class-prior probability* $P(y=0)$ is encoded by $R_y(\cdot)$.

(3) Each independent tree is considered separately. For each tree, starting from the root node, the *class-conditional probability* of each node is encoded layer by layer, respectively. That is, the *class-conditional probability* $p(x_{\mathrm{root}_i}|y)$ of the root node is encoded by $CR_y(\cdot)$, while the *class-conditional probability* $p(x_j|y, x_{\mathrm{parent}_j})$ of each child node is encoded by $C^2R_y(\cdot)$.

IV. IMAGE CLASSIFICATION BASED ON QBCs

The key to the accuracy of a Bayes classifier lies in feature selection and the structure of Bayesian networks. In this paper we propose an image classification framework based on QBCs and local feature sampling, as illustrated in Fig. 7. First, some local areas are selected in an image for feature sampling. Then the sampled pixels are pooled and binarized to obtain local binary attributes of features. Finally, a QBC is constructed based on a Bayesian network model with local attributes as nodes for image classification.

A. Local feature sampling

For image classifications using a naïve QBC, if each pixel of the image is taken as a node of a Bayesian network, the network will be very complex. For example, a 28 × 28 image requires 784 nodes to represent the network with each node having an integer value from 0 to 255. Although n qubits can form a feature space with a dimension of 2^n, the resources required for the quantum Bayesian networks are still relatively large. In addition, considering the complex relationship of interdependence among featuress attributes, the complexity of Bayesian networks will continue to increase, which hinders the implementation of QBCs on current noisy intermediate-scale quantum (NISQ) devices.

Since a single-qubit can represent a two-state node of a Bayesian network, the value of attribute nodes needs to be binarized. To reduce computational complexity and utilize features effectively for classification, we propose the local feature sampling method. This method aims to obtain a small number of binary local key attributes from an image. Besides, background pixels shared by some images might not provide useful information for accurate classification by Bayes classifiers, which means they can be ignored to further reduce the attribute number. The local feature sampling method can reduce the number of attribute nodes in Bayesian networks, decrease the scale and computational complexity of QBCs, and diminish the influence of quantum noise, which is beneficial for the experimental implementation of quantum circuits.

As a preprocessing of data, the local feature sampling method is illustrated in Fig. 7, which is performed as follows:

(1) Local sampling and average pooling. Selects n points on an image as centers of n sampling blocks. By using a convolution operation on each sampling block, an image is

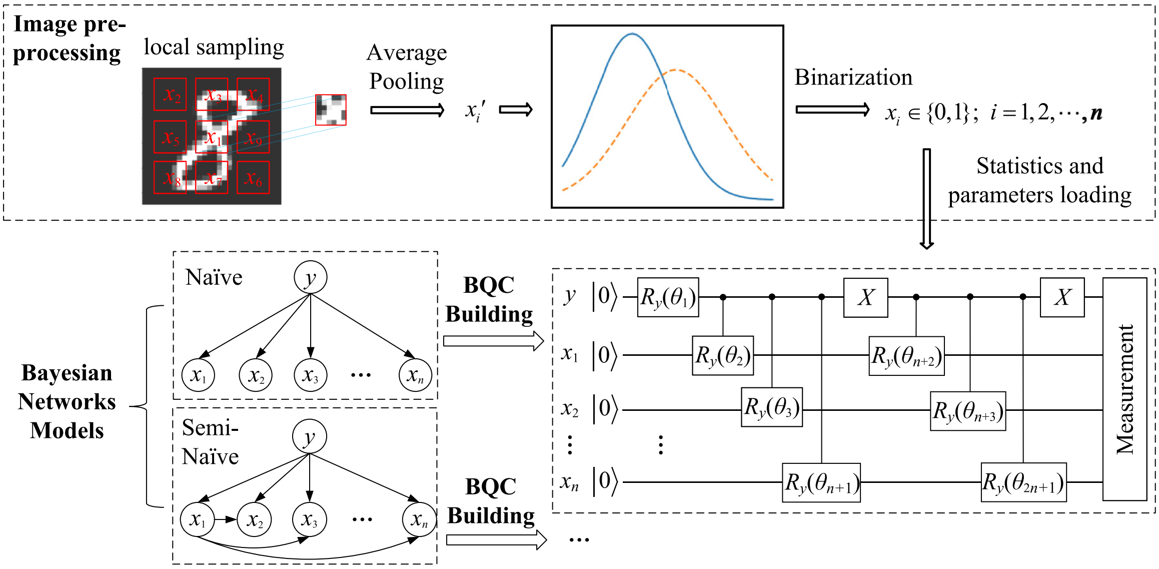


FIG. 7. Framework of image classifications based on QBCs.

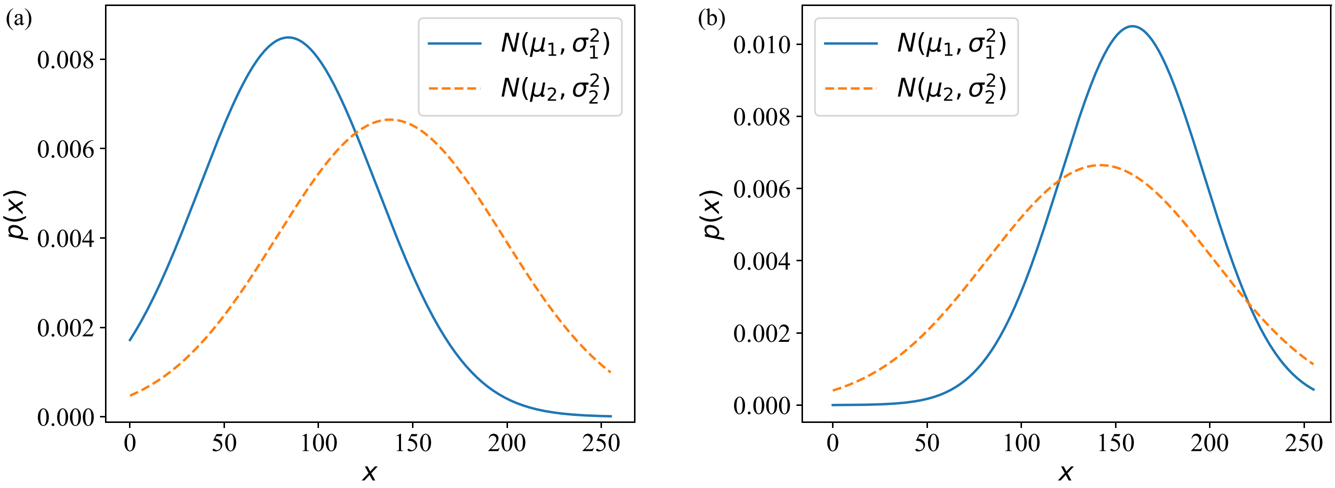


FIG. 8. Two Gaussian functions with intersections: (a) one intersection; (b) two intersections.

locally sampled according to a specified block size (convolution kernel size). Next, the average pooling [45] is applied on each block to obtain $x_i' \in [0, 255]$ of the attribute node x_i in the Bayesian network.

(2) Feature binarization. In a classical Bayes classifier, assume that the probability $p(x_i|y)$ is continuously distributed and satisfies $p(x_i|y) \sim N(\mu_{y,i}, \sigma_{y,i}^2)$, where $\mu_{y,i}$ and $\sigma_{y,i}^2$ are the mean and variance of the attribute x_i on the yth class. One can obtain the *class-prior probability* by calculating the probability density function. However, it is challenging to replicate this process for a QBC. In this case, binarization is used to transform x_i into either 0 or 1 such that it can be represented by a single qubit.

Here we use the maximum likelihood estimation (MLE) [46,47] method adopted in classical Bayes classifiers for obtaining *class-conditional probabilities* of continuous variables. For the binary classification problem, this method runs as follows. Assume that $p(x_i|y_0) \sim N(\mu_{y_0,i}, \sigma_{y_0,i}^2)$ and $p(x_i|y_1) \sim N(\mu_{y_1,i}, \sigma_{y_1,i}^2)$. When two Gaussian functions have only one intersection, which is denoted as x_{ins}, as shown in Fig. 8(a), the attribute value is set as

$$x_i = \begin{cases} 0, & \text{if } x_i' \leqslant x_{\text{ins}} \text{ and } \mu_{y_0,i} \leqslant \mu_{y_1,i}, \ \text{ or } x_i' > x_{\text{ins}} \text{ and } \mu_{y_0,i} > \mu_{y_1,i}, \\ 1, & \text{otherwise}, \end{cases} \tag{19}$$

where x_i' is generated by the average pooling process. If there are two intersections $x_{\text{ins}1}$ and $x_{\text{ins}2}$ with $x_{\text{ins}1} \leqslant x_{\text{ins}2}$, as shown in Fig. 8(b), in cases where $\mu_{y_0,i} \leqslant \mu_{y_1,i}$, the attribute value is set as

$$x_i = \begin{cases} 0, & \text{if } x_i' \leqslant x_{\text{ins}1}, \ \text{ or } x_{\text{ins}1} \leqslant x_i' \leqslant x_{\text{ins}2} \text{ and } |x_{\text{ins}1} - x_i'| \leqslant |x_{\text{ins}2} - x_i'|, \\ 1, & \text{otherwise}. \end{cases} \tag{20}$$

Conversely, when $\mu_{y_0,i} > \mu_{y_1,i}$, the attribute value is set as

$$x_i = \begin{cases} 1, & \text{if } x_i' \leqslant x_{\text{ins}1}, \ \text{ or } x_{\text{ins}1} \leqslant x_i' \leqslant x_{\text{ins}2} \text{ and } |x_{\text{ins}1} - x_i'| \leqslant |x_{\text{ins}2} - x_i'|; \\ 0, & \text{otherwise}. \end{cases} \tag{21}$$

By utilizing the aforementioned method, one can obtain a limited number of binary local attributes of features. These local attributes serve as nodes of Bayesian networks for the constructions of QBCs.

B. Image classification algorithm based on QBCs

The image classification algorithm based on QBCs and local feature sampling is shown in Fig. 7. The algorithm consists of two stages, namely, the image preprocessing stage and the Bayesian networks and QBCs construction stage. In the image preprocessing stage, n local areas are chosen for sampling. After that, the sampled attributes are pooled, and the Gaussian binarization method is applied. This preprocessing converts the value of sampled key attributes into either 0 or 1, allowing an attribute to be represented by a single-qubit in QBCs. In the second stage, a Bayesian network model is selected, and the corresponding QBC is constructed with local key attributes as nodes. The algorithm is shown in Algorithm 4, which runs as follows:

(1) A Bayesian network is selected.

(2) Local sampling and average pooling are performed on images from the training set, and the mean value μ_i and variance σ_i of the corresponding attributes are calculated. The Gaussian binarization method is executed to obtain the binarized attributes, as described in Algorithms 1, 2, and 3.

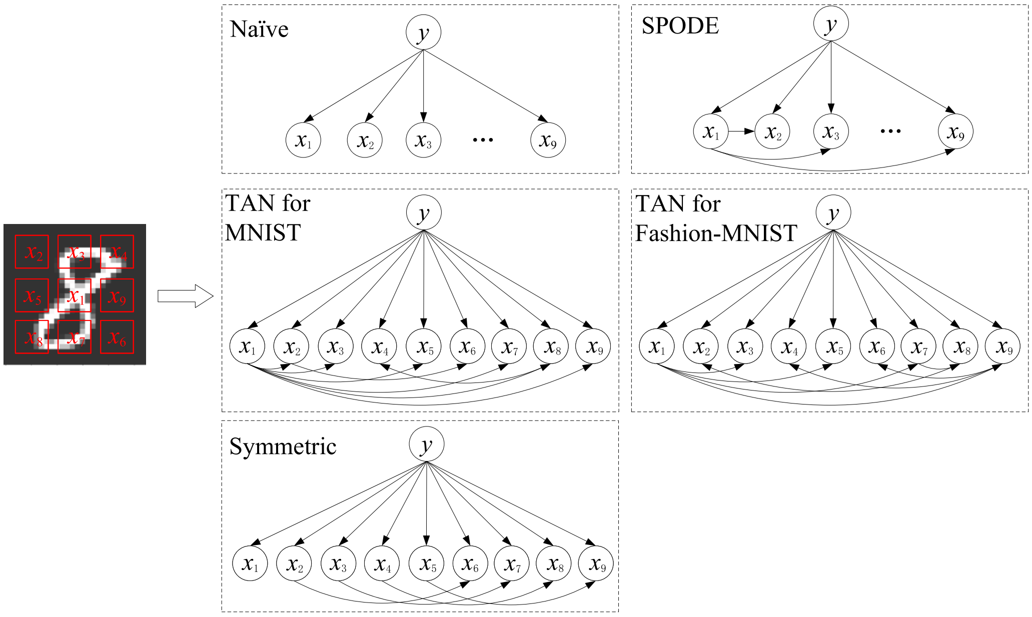


FIG. 9. Sampling blocks and four types of Bayesian networks used in the simulation: the naïve, SPODE, TAN, and symmetric Bayesian networks.

(3) The QBC circuit is constructed based on the chosen Bayesian network model using local binarized attributes as nodes. Also see Algorithm 4.

(4) The *class-prior probability* and *class-conditional probability* required for the Bayesian network are calculated statistically. These values are then loaded into the QBC circuit as controlled angles to complete the construction of the QBC. Also see Algorithm 4.

(5) To predict the class of a new image, one needs to repeat step (b) to obtain a new local key feature X' of the image, and then measure the probabilities of states $|0X'\rangle$ and $|1X'\rangle$ on the QBC circuit. The class with a higher probability will be the classification result of the image. Also see Algorithm 4.

V. SIMULATIONS

In this paper we simulate four QBCs on the MindQuantum [39] quantum simulation platform. The performances of these QBCs are tested using the MNIST [40] and Fashion-MNIST [41] data sets. Both the MNIST and Fashion-MNIST data sets consist of a training set of 60 000 images and a test set of 10 000 images, with the size of each image is 28×28. In the preprocessing stage, the local feature sampling method is used with the sampling block size set to 7×7, and the average pooling method is applied. All images are sampled according to the sampling blocks shown in Fig. 9 to obtain $n = 9$ local key attribute nodes with each node having only a binary value, i.e., either 0 or 1.

As is shown in Fig. 9, four Bayesian network models are used to construct four corresponding QBCs, i.e., the naïve QBC, the SN-QBC based on the SPODE structure (SPODE-QBC), the SN-QBC based on the TAN structure (TAN-QBC), and the SN-QBC based on the symmetric relationships of attributes (symmetric-QBC). For the TAN structures of the MNIST and Fashion-MNIST data sets, the spanning trees of the training data from two data sets are calculated according to the maximum weighted spanning tree algorithm [38,43] introduced in Sec. III C, where related TAN structure Bayesian networks are shown in Fig. 9.

A. The accuracy of binary classification

The MNIST and Fashion-MNIST data sets are used to verify the binary classification effects of four QBCs

ALGORITHM 1. samplingLocalFeatures()

```
Data: Training or test set: $Imgs = \{img_1, img_2, \cdots, img_{size}\}$, sampling points number: $n$;
Result: Local features: $Features = \{fea_1, fea_2, \cdots, fea_{size}\}$ with $fea_i = \{x'_1, x'_2, \cdots, x'_n\}$ and $x'_i \in [0, 255]$;
// sampling local feature on each image
for $j \leftarrow 1$ to $size$ do
    $feature \leftarrow 0$
    for $i \leftarrow 1$ to $n$ do
        $x'_i \leftarrow avg(getBlock(img_j, i))$ // sampling on point $i$
        $feature[i] \leftarrow x'_i$
    end
    $Features[j] \leftarrow feature$
end
```

ALGORITHM 2. getIntersections()

```
Data: Local features of training set: $Features = \{fea_1, fea_2, \cdots, fea_{size}\}$; $n$;
Result: Intersections on $n$ points: $intersections = \{intersec_1, intersec_2, \cdots, intersec_n\}$; means of the training set on
        attributes: $\mu_0, \mu_1$;
// get intersections of two Gaussian functions
for $i \leftarrow 1$ to $n$ do
    $\mu_{0,i}, \mu_{1,i} \leftarrow getMeans(Features)$
    $\sigma_{0,i}^2, \sigma_{1,i}^2 \leftarrow getStdDev(Features)$
    $intersection[i] \leftarrow getIntersection(\mu_{0,i}, \mu_{1,i}, \sigma_{0,i}^2, \sigma_{1,i}^2)$
end
```

ALGORITHM 3. getBinaryFeatures()

```
Data: Local features: $Features = \{fea_1, fea_2, \cdots, fea_{size}\}$; Intersections and means on training set:
      $intersections, \mu_0, \mu_1$; $n$;
Result: Binary features: $biFeatures = \{bifea_1, bifea_2, \cdots, bifea_{size}\}$ with $bifea_i = \{x_1, x_2, \cdots, x_n\}$ and $x_i \in \{0, 1\}$;
// get binary feature of each image
for $j \leftarrow 1$ to $size$ do
    $feature \leftarrow Features[j]$, $bifeature \leftarrow 0$
    for $i \leftarrow 1$ to $n$ do
        $bifeature \leftarrow getBinary(feature[i], intersection[i], \mu_{0,i}, \mu_{1,i})$
    end
    $biFeatures[j] \leftarrow bifeature$
end
```

ALGORITHM 4. The image classification algorithm based on QBCs.

```
Data: Training & test set: $Img_{\text{train}} = \{img_1, img_2, \cdots, img_{size_1}\}$, $Img_{\text{test}} = \{img_1, img_2, \cdots, img_{size_2}\}$; Training &
      test labels: $Labels_{\text{train}}, Labels_{\text{test}}$; QBCs type: $type \in$ {"naive", "SPODE", "TAN", "symmetric"}; Sampling
      points number: $n$.
Result: Classification accuracy: $acc \in [0, 1]$;
// call Algorithm1 to get local features of training set
$features_1 \leftarrow samplingLocalFeatures(Img_{\text{train}}, n)$
// call Algorithm2 to get intersections of training set
$intersections, \mu_0, \mu_1 \leftarrow getIntersections(features_1)$
// call Algorithm3 to get binary features of training set
$biFeatures_1 \leftarrow getBinaryFeatures(features_1, intersections, \mu_0, \mu_1, n)$
// Build the QBC circuit
$circuit \leftarrow Bayesian\_circuit(type, n)$
// Calculate the class-prior and class-conditional probabilities
$probs \leftarrow getProbabilities(biFeatures_1, type, Labels_{\text{train}})$
// simulate the quantum circuit
$final\_state \leftarrow Simulator(n).apply\_circuit(circuit, probs).get\_qs()$
// predict test set
$features_2 \leftarrow samplingLocalFeatures(Img_{\text{test}}, n)$ // call Algorithm1 to get features of test set
$biFeatures_2 \leftarrow getBinaryFeatures(features_2, intersections, \mu_0, \mu_1, n)$ // call Algorithm3 to get binary features
    of test set
for $j \leftarrow 1$ to $size_2$ do
    $p_0, p_1 \leftarrow searchFinalStates(final\_state, biFeatures_2[j])$ // search the probabilities of $|0X'\rangle$ and $|1X'\rangle$
    if $p_0 > p_1$ then
        $results[j] \leftarrow 0$ // labeled as class 0
    else
        $results[j] \leftarrow 1$ // labeled as class 1
    end
end
$acc \leftarrow calculateAcc(results, Labels_{\text{test}})$
```

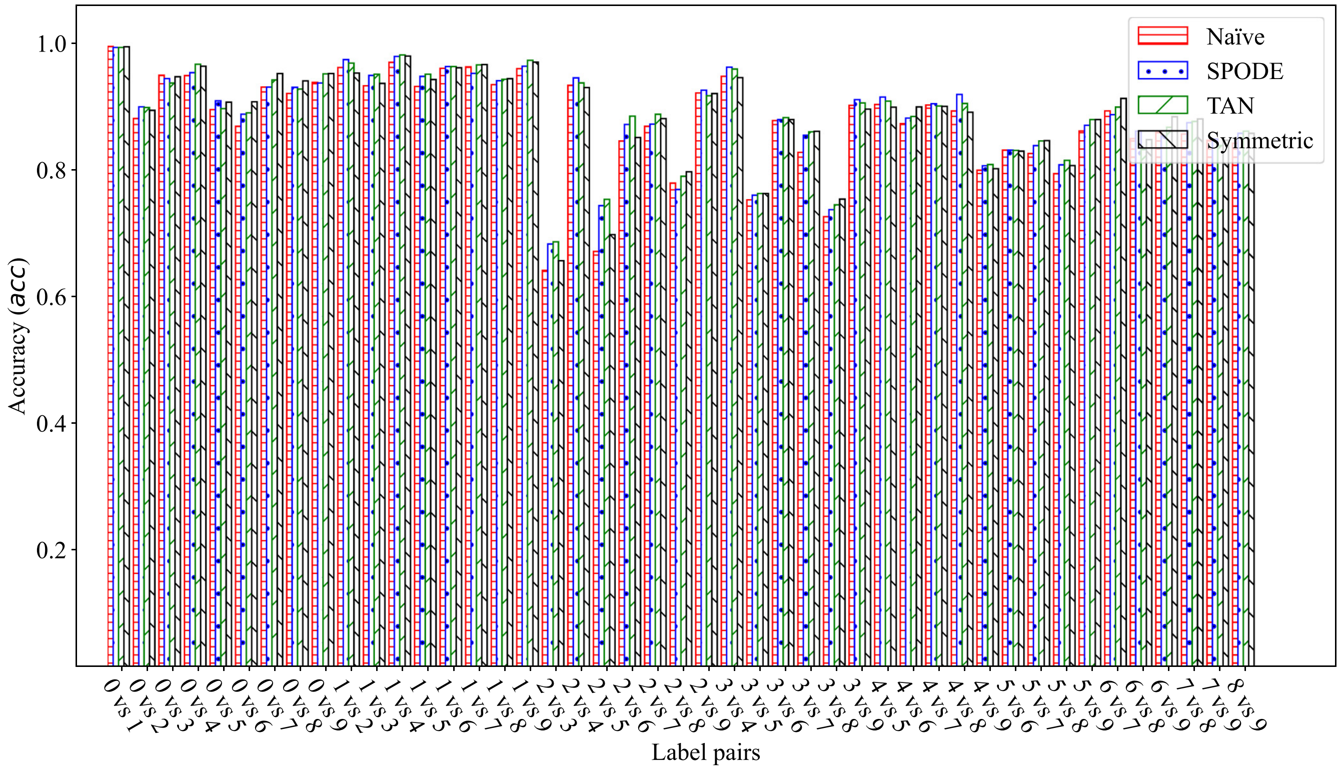


FIG. 10. Classification accuracies of every two classes in the MNIST data set on the naïve QBC, SPODE-QBC, TAN-QBC, and symmetric-QBC.

on every two image classes. Figures 10 and 11 show the classification accuracies of four QBCs on the MNIST and Fashion-MNIST data sets, respectively. It can be seen from Fig. 10 and Fig. 11 that the TAN-QBC shows a better

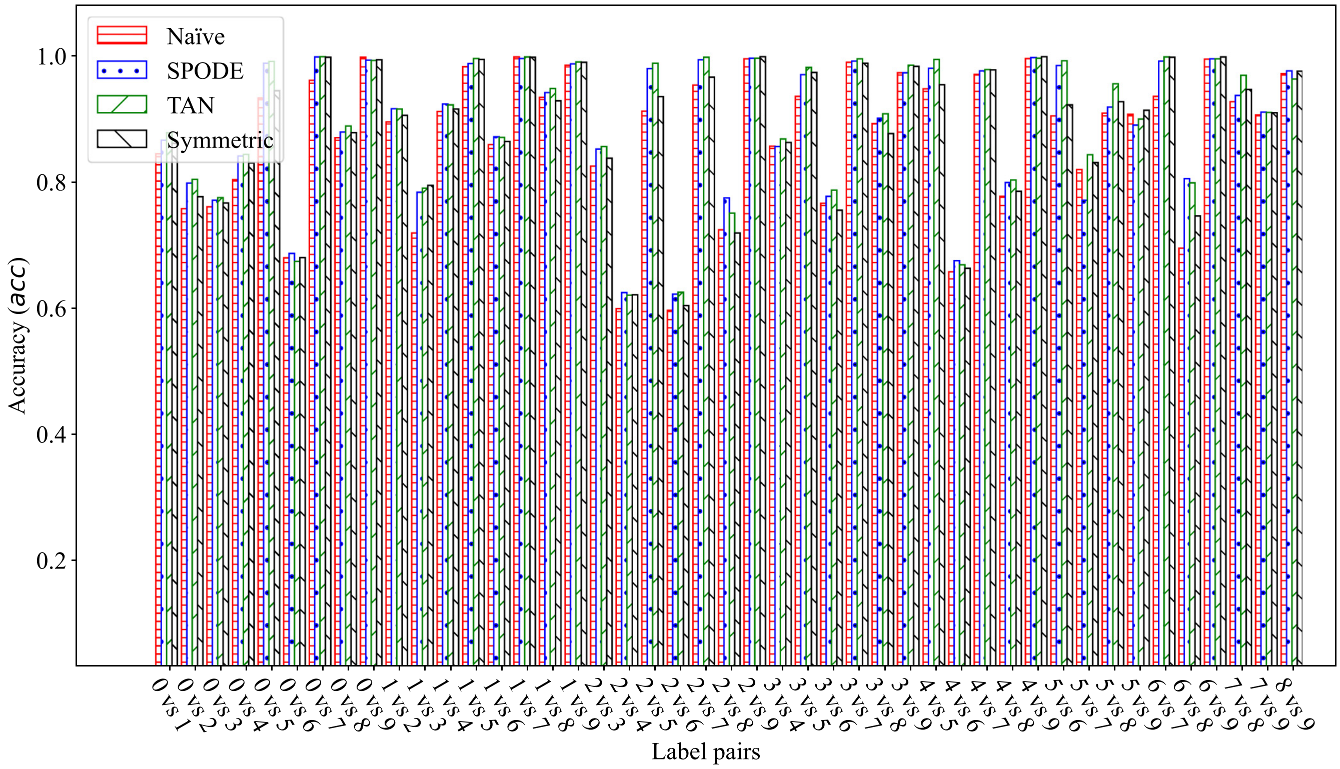


FIG. 11. Classification accuracies of every two classes in the Fashion-MNIST data set on the naïve QBC, SPODE-QBC, TAN-QBC, and symmetric-QBC.

TABLE I. Overall performances of the four QBCs in two data sets, where the boldface text indicates the optimal results.

Data set	Classifier	$\overline{\text{acc}}$	s_{N-1}^2	$\overline{\text{precision}}$	$\overline{\text{recall}}$	F_1
MNIST	Naïve-QBC	0.8767	0.00597	0.8844	0.8689	0.8762
	SPODE-QBC	0.8869	**0.00501**	0.8929	0.8808	0.8864
	TAN-QBC	**0.8891**	0.00482	**0.8930**	**0.8856**	0.8889
	Symmetric-QBC	0.8869	0.00537	**0.8930**	0.8813	0.8867
Fashion-MNIST	Naïve-QBC	0.8712	0.01273	0.8726	0.8748	0.8728
	SPODE-QBC	0.8916	**0.01167**	0.8863	0.9107	0.8962
	TAN-QBC	**0.8972**	0.01204	0.8885	**0.9182**	0.9019
	Symmetric-QBC	0.8835	0.01233	0.8807	0.8942	0.8859

classification performance in most cases, while the symmetric-QBC and the SPODE-QBC show similar classification effects to TAN-QBC.

In addition, the simulation results show that there exist a small number of classes pairs on which QBCs cannot achieve ideal classification results, for example, 2 vs 3 and 2 vs 5 in the MNIST data set, and 2 vs 4 and 2 vs 6 in the Fashion-MNIST data set, for the following reasons:

(1) The attribute number n is relatively small for containing less information to significantly distinguish each class from others. A bigger n needs to be considered.

(2) The Bayesian network used in the simulation may not be suitable for all classes of data. Different Bayesian networks should be considered for different data.

(3) The hyperparameters involved in the simulation, such as the sampling block size (convolution kernel size) and the pooling method, etc., have a significant impact on the results, which should be selected carefully.

(4) Some attributes extracted from the local sampling and binarization method cannot distinguish data from two classes very well. Other sampling and binarization methods should be considered.

B. Overall performance of QBCs

To evaluate the overall performance of each QBC, the average classification accuracy, variance, average precision, average recall, and average F_1 score [48,49] are calculated. The average classification accuracy of a binary classifier for two classes in the MNIST and Fashion-MNIST data sets is defined as

$$\overline{\text{acc}} = \frac{1}{45}\sum_{i=0}^{8}\sum_{j=i+1}^{9} acc_{ij}, \tag{22}$$

where acc_{ij} represents the accuracy of binary classification of classes i and j. The variance is defined as [50,51]

$$s_{N-1}^2 = \frac{1}{(45-1)}\sum_{i=0}^{8}\sum_{j=i+1}^{9} \times (\overline{\text{acc}} - acc_{ij})^2, \tag{23}$$

while the average precision, average recall, and average F_1 score are defined as

$$\overline{\text{prec}} = \frac{1}{45}\sum_{i=0}^{8}\sum_{j=i+1}^{9}\frac{TP_{ij}}{TP_{ij}+FP_{ij}},$$
$$\overline{\text{recall}} = \frac{1}{45}\sum_{i=0}^{8}\sum_{j=i+1}^{9}\frac{TP_{ij}}{TP_{ij}+FN_{ij}},$$
$$F_1 = \frac{2\times\overline{\text{prec}}\times\overline{\text{recall}}}{\overline{\text{prec}}+\overline{\text{recall}}}, \tag{24}$$

where TP means the number of true positive, FP means false positive, and FN means false negative. Table I shows the overall performance of the four QBCs mentioned above for all binary classification pairs in the MNIST and Fashion-MNIST data sets. As is shown in Table I, the TAN-QBC, the SPODE-QBC, and the symmetric-QBC demonstrate strong classification accuracies on both the MNIST and Fashion-MNIST data sets, while the naïve-QBC also shows relatively good classification performance. Overall, the TAN-QBC performs the best on both the MNIST and Fashion-MNIST data sets.

C. Comparison with other classifiers

Table II shows the comparison of classification performance of the classical Gaussian naïve Bayes classifier [52], the quantum convolutional neural network (QCNN) [14], and four QBCs for classes 0 and 1 in the MNIST and Fashion-MNIST data sets. The results show that QBCs performs better than the classical Bayes classifier and the QCNN on

TABLE II. Comparison of classification accuracy of classes 0 and 1 in two data sets by different classifiers, where the boldface text indicates the optimal results.

Classifier	MNIST 0 vs 1	Fashion-MNIST 0 (t-shirt) vs 1 (trouser)
Classical naïve Bayes [52]	0.985	0.897
QCNN [14]	0.987	**0.941**
Naïve-QBC	**0.994**	0.844
SPODE-QBC	0.993	0.866
TAN-QBC	0.993	0.878
Symmetric-QBC	**0.994**	0.861

TABLE III. Gate numbers used in our QBCs for image classification.

Classifier	$\#Ry(\cdot)$	$\#X$	$\#CRy(\cdot)$	$\#C^2Ry(\cdot)$
Naïve-QBC	1	2	18	0
SPODE-QBC	1	6	2	32
TAN-QBC	1	$12^{‡}$ $(16^{§})$[a]	2	32
Symmetric-QBC	1	12	10	16

[a] ‡ and § represent gate numbers for the MNIST and Fashion-MNIST data sets, respectively.

TABLE IV. Average classification accuracies (acc) of four QBCs on the MNIST data set by using different block sizes, where the boldface text indicates the optimal results.

	Block size					
Classifier	1 × 1	3 × 3	5 × 5	7 × 7	9 × 9	11 × 11
Naïve-QBC	0.8376	0.8651	**0.8820**	0.8767	0.8622	0.8250
SPODE-QBC	0.8399	0.8672	0.8840	**0.8869**	0.8729	0.8464
TAN-QBC	0.8439	0.8717	0.8874	**0.8891**	0.8799	0.8527
Symmetric-QBC	0.8431	0.8715	0.8868	**0.8869**	0.8740	0.8456

the MNIST data set, while the QCNN performs best on the Fashion-MNIST data set. It is important to note that the classical Bayes classifier and the QCNN use all 786 features, while the four QBCs in this paper use only nine binary features.

VI. ANALYSIS

A. Computation complexity

The numbers of quantum gates used in our four QBCs are summarized in Table III. The single-qubit $Ry(\cdot)$ gate is used to encode the *class-prior probability* $P(y=0)$. The two-qubit $CRy(\cdot)$ gate is used to encode the *class-conditional probabilities* $P(x_i=0|y=0)$ and $P(x_i=0|y=0)$, with x_i having only one parent node y. The three-qubit $C^2Ry(\cdot)$ gate is used to encode the *class-conditional probabilities* $P(x_i=0|y,x_j)$, with x_i having two parent nodes y and x_j. The X gate is used to set the state of control qubits y, or y and x_j. As shown in Fig. 9, each node in the Bayesian networks of our four QBCs contains at most two parents, which means the $C^2Ry(\cdot)$ is the most complex gate in our QBCs.

For a Bayesian network with n attribute nodes, suppose there are n_1 nodes, each of which has only one parent y, and $n-n_1$ nodes, each of which has two parents. The gate numbers required for the construction of QBCs are as follows:

$$\begin{aligned}\#Ry(\cdot)&=1,\\ \#CRy(\cdot)&=2n_1,\\ \#C^2Ry(\cdot)&=4(n-n_1),\\ \#X&=[2+4,2+2(n-n_1)+2].\end{aligned}\tag{25}$$

Note that the number of X gate varies depending on the structure of the Bayesian network. But in the worst case, $4+2(n-n_1)$ is enough for the construction of QBCs.

Besides, the number of the *class-prior probability* and *class-conditional probabilities* calculated from the binary features of the training set are equivalent to $\#Ry(\cdot)$ and $\#CRy(\cdot)+\#C^2Ry(\cdot)$.

It is interesting to compare the complexity of a QBC with a QCNN. The complexity of our QBCs contains two parts. The first one is the classical part containing the local feature sampling and the probabilities calculations. The second one is the quantum part executing the QBCs' circuits. The complexity of quantum part is denoted as $\mathbf{O}(Q_1)$, and the classical part is as $\mathbf{O}(C_1)$. The complexity of our QBCs algorithms is $\mathbf{O}(Q_1)+\mathbf{O}(C_1)$.

The training of a QCNN contains a series of iterative procedures, where each procedure includes a forward propagation performing on the quantum circuit and a backward propagation of the gradient descent algorithm running on classical computers. The complexity of the quantum part is denoted as $\mathbf{O}(Q_2)$, the gradient descent algorithm is $\mathbf{O}(C_2)$, and the epoch of QCNN is N. The complexity of the QCNN is $\mathbf{O}(N\times Q_2)+\mathbf{O}(N\times C_2)$. Note that $\mathbf{O}(Q_1)\approx\mathbf{O}(Q_2)$ and $\mathbf{O}(C_1)\ll\mathbf{O}(C_2)$, which means QBCs are at least N times faster than QCNN in both the quantum and classical parts.

B. Sample block size

In our image classification algorithm, the local feature sampling method (Sec. IV A) is used to extract a small number of binary key features from an image to reduce the computation complexity. Here a 7 × 7 size image block is applied with one of the nine sample points as its center. Similar to the kernel size of classical convolutional neural networks, the block size determines the receptive field of each sampling operation. Smaller block sizes capture more local information, while larger block sizes capture more global information.

Note that other sizes are also possible to choose. Since a pixel is used as the sampling center, the possible block size is $2r+1$ with $r\in\{0,1,2,\ldots\}$. We have tested our four QBCs on the MNIST data set by using six different block sizes: 1 × 1, 3 × 3, 5 × 5, 7 × 7, 9 × 9, and 11 × 11. As is shown in Table IV, the average classification accuracy of a QBC increases with the increase of block size started from 1 × 1, but the accuracy begins to decrease when the size exceeds a certain value. The best block size for the SPODE-QBC, the TAN-QBC, and the symmetric-QBC is 7 × 7, and for the naïve-QBC is 5 × 5.

C. Noise simulation

In the practical implementation of quantum computing, quantum noise emerges as a critical factor, especially in today's NISQ era. We analyze four common types of noise in practice, i.e., the bit-flip, the phase-flip, the amplitude damping, and the depolarizing noise [44], where their Kraus operators are shown in Table V. In our noise simulation, the noise model is set as follows. A noise channel with the error probability (or noise parameter) p is added after each quantum gate of the QBCs' circuit. For example, the circuit of the naïve-QBC in the bit-flip noise with $p=1/50$ is shown in Fig. 12.

Our simulation results of QBCs with four types of noise for binary classification of digits 0 and 1 in the MNIST data set are shown in Fig. 13. The noise parameter p is set from 0 to

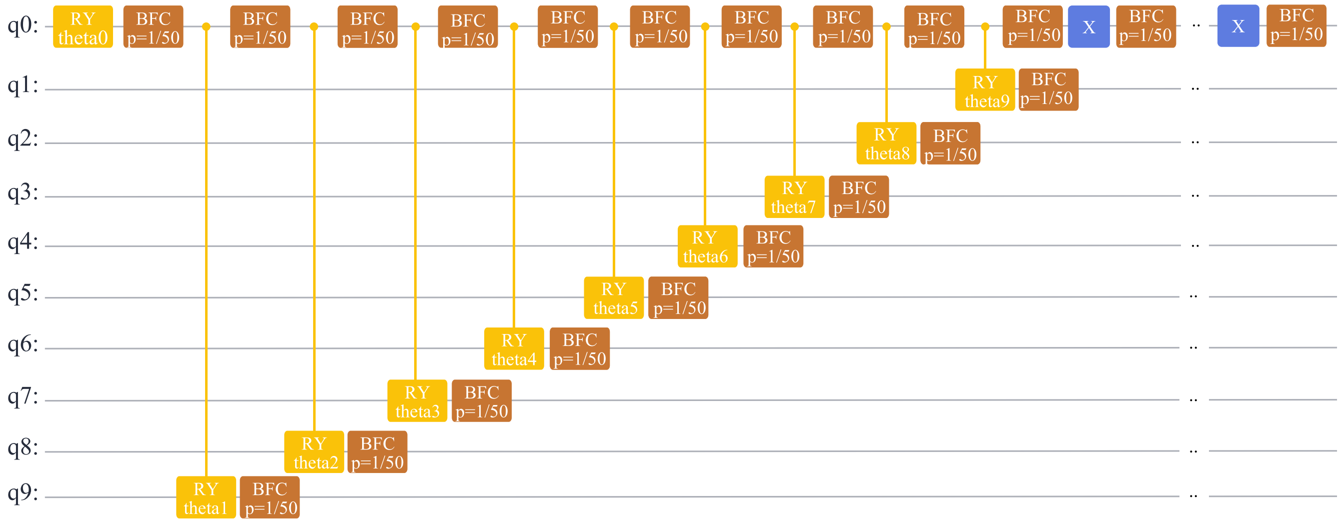
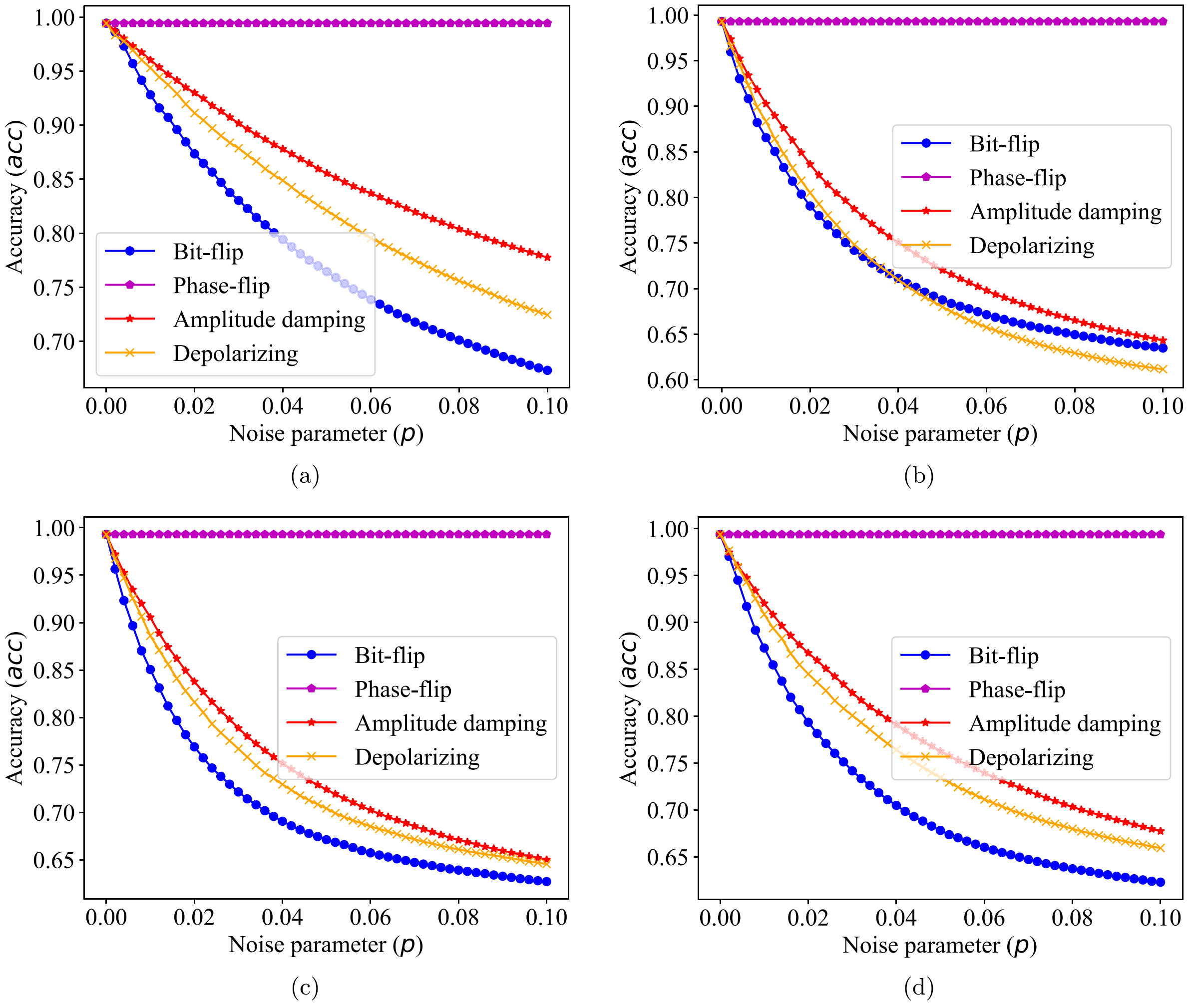


FIG. 12. Quantum circuit of the naïve-QBC in the bit-flip noise with $p = 1/50$, where q0 denotes y, q1 denotes x_9, q2 denotes x_8, . . ., and q9 denotes x_1.



(a)

(b)

(c)

(d)

FIG. 13. Binary classification accuracies of QBCs with four types of noise for classification of digits 0 and 1 in the MNIST data set: (a) the naïve-QBC; (b) SPODE-QBC; (c) TAN-QBC; (d) Symmetric-QBC.

TABLE V. Kraus operators of four types of noise, where $0 \leqslant p \leqslant 1$ is the error probability and I, σ_x, σ_y, and σ_z are the Pauli matrices.

Noise	Kraus operators
Bit-flip	$E_0 = \sqrt{1-p}\,I,\ E_1 = \sqrt{p}\,\sigma_x$
Phase-flip	$E_0 = \sqrt{1-p}\,I,\ E_1 = \sqrt{p}\,\sigma_z$
Amplitude damping	$E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{pmatrix}, E_1 = \begin{pmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{pmatrix}$
Depolarizing	$E_0 = \sqrt{1-p}I, E_1 = \sqrt{\frac{p}{3}}\sigma_x, E_2 = \sqrt{\frac{p}{3}}\sigma_z, E_3 = \sqrt{\frac{p}{3}}\sigma_y$

0.1 with step 0.002. As is shown, the classification accuracy *acc* decreases with the increases of the noise parameter p in the bit-flip, amplitude-damping, and depolarizing noise, while the phase-flip has no effect on the accuracies of all four QBCs.

Besides, even in the presence of noise, the QBCs can still obtain relatively high classification accuracy as long as the noise parameter p is lower than a certain threshold. For example, the naïve-QBC can achieve a classification accuracy higher than 0.9 as long as the noise parameter satisfies $p < 0.014$ in the bit-flip noise, $p < 0.03$ in the amplitude damping noise, and $p < 0.022$ in the depolarizing noise.

VII. DISCUSSION AND CONCLUSIONS

In this paper we study the constructions of QBCs using Bayesian networks. Based on four kinds of Bayesian networks, four QBCs are implemented, that is, the naïve-QBC, SPODE-QBC, TAN-QBC, and symmetric-QBC. We apply these QBCs to image classification and propose the QBC-based image classification algorithm. To reduce the computational complexity, the local feature sampling method is designed to extract a small number of binary attributes from a huge number of image pixels. By retaining essential attributes, this method achieves high classification accuracies with reduced feature size. Fine-tuning the sampling block size and experimenting with various convolution kernels could enhance the accuracy of QBCs.

The classification effects of QBCs are verified on the MindQuantum platform for the MNIST and Fashion-MNIST data sets, respectively. The simulation results show that QBCs perform well for image classification. The TAN-QBC achieves the highest average accuracies on both the MNIST and Fashion-MNIST data sets. While the SPODE-QBC and the symmetric-QBC also show good classification effects. Compared with the naïve-QBC with the simplest feature dependencies among attributes, three seminaïve QBCs with additional dependencies can further improve the classification accuracy at the cost of the increase of a certain computation complexity. This suggests that considering more complex feature dependencies among attributes in Bayesian networks can improve the accuracy of related QBCs.

We also compare QBCs with the classical Bayes classifier and QCNN that use a much larger size of features. The results show that QBCs using a fewer size of features still have advantages in some cases. Our QBCs are the quantum implementations of classical Bayes classifiers, which make use of quantum superposition to compute multiple different states simultaneously. The classification accuracies of QBCs on some data are higher than that of classical Bayes classifier. The reason is that the local feature sampling method (data preprocessing) used in our QBCs can extract more effective features from these data. Unlike QCNN classifiers [12,14], QBCs do not require a time-consuming training process. They only require to calculate the corresponding parameters statistically and load them into quantum circuits, while the classification decision can be made by measuring quantum circuits, which is faster, lower in computational complexity, and less resource-consuming than QCNN. The analysis shows that four QBCs are at least N times faster than QCNN in both the quantum and classical parts.

In our simulations, the QBCs used for image classification considered Bayesian networks with two-state nodes. Each node in the network is represented by a single-qubit in the related QBC circuit. These QBCs can be extended to more complex images or data sets, but several factors require careful consideration:

(1) Attribute (node) number n: Choosing an appropriate n directly impacts the amount of information extracted from raw data, the scale of the quantum circuit, and data processing complexity.

(2) Feature sampling method: An effective feature sampling method can obtain high-quality binary or multivalued attributes. Besides, the selections of the sampling point locations and block sizes also affect the sampling effectiveness.

(3) Nodes with multiple states: Nodes with multiple states carry more information for handling complex data sets. As is stated in Ref. [33], a node with m states can be represented by $\lfloor \log_2 m \rfloor$ qubits. Consequently, the corresponding quantum circuits will also be more complex.

(4) Nodes dependency: Selecting dependencies among attribute nodes is crucial for constructing effective QBCs. Exploring diverse variants of Bayesian networks to develop QBCs that perform effectively across various data sets is an intriguing research direction.

The code that support the findings of this study is available at [53].

ACKNOWLEDGMENTS

This project was supported by the National Natural Science Foundation of China (Grant No. 61601358). The authors thank anonymous reviewers for their helpful comments and suggestions that improved the quality of this paper.

[1] K. Mosegaard and M. Sambridge, Monte Carlo analysis of inverse problems, Inverse Probl. **18**, R29 (2002).

[2] M. Modarres and K. Groth, *Reliability and Risk Analysis* (CRC Press, Boca Raton, 2023).

[3] N. Daniels, Justice, health, and healthcare, Am. J. Bioeth. **1**, 2 (2001).

[4] P. Gary Jarrett, Logistics in the health care industry, Int. J. Phys. Distrib. Logist. Manag. **28**, 741 (1998).

[5] H. Wong, W. Lin, L. Huitema, and E. Arnaud, Multi-polarization reconfigurable antenna for wireless biomedical system, IEEE Trans. Biomed. Circuits Syst. **11**, 652 (2017).

[6] J. Kwisthout, Most probable explanations in bayesian networks: Complexity and tractability, Int. J. Approx. Reason. **52**, 1452 (2011).

[7] D. M. Chickering, D. Heckerman, and C. Meek, Large-sample learning of Bayesian networks is NP-hard, J. Mach. Learn. Res. **5**, 1287 (2004).

[8] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput. **26**, 1484 (1997).

[9] L. K. Grover, Quantum mechanics helps in searching for a needle in a haystack, Phys. Rev. Lett. **79**, 325 (1997).

[10] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum support vector machine for big data classification, Phys. Rev. Lett. **113**, 130503 (2014).

[11] Y. Dang, N. Jiang, H. Hu, Z. Ji, and W. Zhang, Image classification based on quantum K-Nearest-Neighbor algorithm, Quantum Inf. Process. **17**, 239 (2018).

[12] I. Cong, S. Choi, and M. D. Lukin, Quantum convolutional neural networks, Nat. Phys. **15**, 1273 (2019).

[13] I. Kerenidis, J. Landman, and A. Prakash, Quantum algorithms for deep convolutional neural networks, arXiv:1911.01117.

[14] T. Hur, L. Kim, and D. K. Park, Quantum convolutional neural network for classical data classification, Quantum Mach. Intell. **4**, 3 (2022).

[15] L. H. Gong, J. J. Pei, T. F. Zhang, and N. R. Zhou, Quantum convolutional neural network based on variational quantum circuits, Opt. Commun. **550**, 129993 (2024).

[16] M. Schuld and F. Petruccione, Quantum ensembles of quantum classifiers, Sci. Rep. **8**, 2772 (2018).

[17] A. Macaluso, L. Clissa, L. Stefano, and C. Sartori, Quantum ensemble for classification, arXiv:2007.01028.

[18] I. C. Araujo and A. J. Da Silva, Quantum ensemble of trained classifiers, in *2020 International Joint Conference on Neural Networks (IJCNN)* (IEEE, Glasgow, UK, 2020), pp. 1–8.

[19] A. Macaluso, S. Lodi, and C. Sartori, Quantum algorithm for ensemble learning, in *21st Italian Conference on Theoretical Computer Science* (RWTH Aachen, Aachen, 2020), pp. 149–154.

[20] X. Y. Zhang and M. M. Wang, An efficient combination strategy for hybrid quantum ensemble classifier, Int. J. Quantum Inf. **21**, 2350027 (2023).

[21] Z. Liu, P. X. Shen, W. Li, L. M. Duan, and D. L. Deng, Quantum capsule networks, Quantum Sci. Technol. **8**, 015016 (2023).

[22] E. Farhi and S. Gutmann, Quantum computation and decision trees, Phys. Rev. A **58**, 915 (1998).

[23] R. Heese, P. Bickert, and A. E. Niederle, Representation of binary classification trees with binary features by quantum circuits, Quantum **6**, 676 (2022).

[24] W. Li and D. L. Deng, Recent advances for quantum classifiers, Sci. China Phys. Mech. Astron. **65**, 220301 (2022).

[25] S. Youssef, Quantum mechanics as an exotic probability theory, arXiv:quant-ph/9509004.

[26] R. R. Tucci, Quantum Bayesian nets, Int. J. Mod. Phys. B **9**, 295 (1995).

[27] M. Ozols, M. Roetteler, and J. Roland, Quantum rejection sampling, ACM Trans. Comput. Theory **5**, 1 (2013).

[28] G. H. Low, T. J. Yoder, and I. L. Chuang, Quantum inference on Bayesian networks, Phys. Rev. A **89**, 062315 (2014).

[29] N. Wiebe and C. Grandade, Can small quantum systems learn?, Quantum Info. Comput. **17**, 0568 (2017).

[30] C. Moreira and A. Wichert, Quantum-like Bayesian networks for modeling decision making, Front. Psychol. **7**, 11 (2016).

[31] S. Woerner and D. J. Egger, Quantum risk analysis, npj Quantum Inf. **5**, 15 (2019).

[32] G. Escrig, R. Campos, P. A. M. Casares, and M. A. Martin-Delgado, Parameter estimation of gravitational waves with a quantum metropolis algorithm, Class. Quantum Grav. **40**, 045001 (2023).

[33] S. E. Borujeni, S. Nannapaneni, N. H. Nguyen, E. C. Behrman, and J. E. Steck, Quantum circuit representation of Bayesian networks, Expert Syst. Appl. **176**, 114768 (2021).

[34] W. Fathallah, N. B. Amor, and P. Leray, An optimized quantum circuit representation of bayesian networks, in *European Conference on Symbolic and Quantitative Approaches with Uncertainty* (Springer, Cham, 2023), pp. 160–171.

[35] I. Rish, An empirical study of the naive Bayes classifier, in *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence* (Seattle, 2001), pp. 41–46.

[36] K. M. Leung, Naive Bayesian classifier, Polytechnic University, Department of Computer Science/Finance and Risk Engineering, 2007, https://cse.engineering.nyu.edu/~mleung/FRE7851/f07/naiveBayesianClassifier.pdf.

[37] I. Kononenko, Semi-naive Bayesian classifier, in *Machine Learning–EWSL-91: European Working Session on Learning*, Lecture Notes in Computer Science, Vol. 482 (Springer, Berlin, Heidelberg, 1991), pp. 206–219.

[38] Z.-H. Zhou, *Machine Learning* (Springer Nature, Singapore, 2021).

[39] Mindquantum Developer, Mindquantum, version 0.9.11, https://gitee.com/mindspore/mindquantum.

[40] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE **86**, 2278 (1998).

[41] H. Xiao, K. Rasul, and R. Vollgraf, Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms, arXiv:1708.07747.

[42] J. Hoffmann-Jørgensen and G. Pisier, The law of large numbers and the central limit theorem in Banach spaces, Ann. Probab. **4**, 587 (1976).

[43] P. M. Camerini, The min-max spanning tree problem and some extensions, Inf. Process. Lett. **7**, 10 (1978).

[44] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).

[45] D. Yu, H. Wang, P. Chen, and Z. Wei, Mixed pooling for convolutional neural networks, in *Rough Sets and Knowledge*

Technology: 9th International Conference, RSKT 2014 (Springer, Cham, 2014), pp. 364–375.

[46] P. Dutilleul, The MLE algorithm for the matrix normal distribution, J. Stat. Comput. Simul. **64**, 105 (1999).

[47] M. Boullé, MODL: A Bayes optimal discretization method for continuous attributes, Mach. Learn. **65**, 131 (2006).

[48] Y. Liu, Y. Zhou, S. Wen, and C. Tang, A strategy on selecting performance metrics for classifier evaluation, Int. J. Mob. Comput. **6**, 20 (2014).

[49] O. Caelen, A Bayesian interpretation of the confusion matrix, Ann. Math. Artif. Intell. **81**, 429 (2017).

[50] E. W. Weisstein, Sample variance, https://mathworld.wolfram.com/SampleVariance.html.

[51] D. Zwillinger, *CRC Standard Mathematical Tables and Formulas* (Chapman and Hall/CRC, New York, 2000).

[52] Gaussian naive Bayes, https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html.

[53] https://github.com/wang-labs/QBCs/.