Neurocomputing 493 (2022) 435–444

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Accelerating spiking neural networks using quantum algorithm with high success probability and high calculation accuracy

Yanhu Chen [a], Cen Wang [b], Hongxiang Guo [a,*], Xiong Gao [a], Jian Wu [a]

[a] Institute of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China
[b] Photonic Transport Network Laboratory, KDDI Research Inc., Saitama-ken 356-8502, Japan

ARTICLE INFO

Article history:
Received 3 December 2020
Revised 1 July 2021
Accepted 3 February 2022
Available online 7 February 2022

Keywords:
Quantum computing
Neuromorphic computing
Spiking neural network
Scalable quantum circuits

ABSTRACT

Spiking neural networks (SNNs) are a kind of neuromorphic computing which meticulously imitates the operations of biological nervous systems. Thus, the SNN is seen as a promising approach to further improve the level of intelligence that the legacy artificial neural networks achieved. In a spiking neuron of an SNN, it is a key step with the highest computing complexity to find out the moment when the output stimuli occur. Classic computer (i.e., electrical computer) based acceleration cannot reduce the computational complexity of the operations in the spiking neurons. As a result, the SNN suffers from scaling and deeper emulation problems. Considering these, in this paper, we propose a quantum algorithm to reduce the complexity of the key steps in the SNN and use this algorithm to build a quantum spiking neuron network (QSNN). More specifically, first, we give mathematical proof that the problem of finding the output stimuli is approximately equal to the problem of calculating the unsigned vector inner products, which can transfer to a quantum operation. Second, we design a scalable quantum circuit of QSNN for data of any dimension and evaluate its basic success probability and calculation accuracy. Third, to improve the QSNN performance, we propose a method to improve the minimum success probability to 99.8%, by repeatedly performing the quantum circuit only 11 times. Fourth, we prove that the computational complexity of the QSNN is a log-polynomial relationship with the data dimension, which is much lower than that of the linear complexity of the classic SNN. Finally, we apply the QSNN to solve classification tasks on two real-world datasets, i.e., MNIST and fashion MNIST which both have two noise levels (i.e., no noise and 50% of the noise). The experiment results show that in addition to the acceleration, QSNN and SNN have equal classification accuracy, which suggests the feasibility and robustness of the QSNN.

© 2022 Elsevier B.V. All rights reserved.

1. Introduction

Spiking neural networks (SNNs) are regarded as the next-generation neural architecture, which imitates the biological nervous system as a potential field to reach higher-level intelligence such as brain-like learning [1–5]. Commonly, the leaky integrate-and-fire (LIF) spiking neuron, inspired by the experiment results from the neuroscience, is a basic unit of various kinds of SNNs [6–9]. More specifically, a potential on a spiking neuron oscillates as the corresponding responses to a series of temporal-spatial stimuli. The spiking neuron only fires at the time that the potential crosses an established threshold (i.e., a voltage). Then, the fired spiking neuron generates an output stimulus. This output stimulus can be retrieved or transmitted to downstream spiking neurons [6]. Therefore, to make the SNN work, it is a key step on each spiking neuron to calculate the responses (i.e., the moment and the value) and to determine when to fire.

Compared to a multiplication in a neuron of the traditional artificial neural networks (ANNs), the calculation (i.e., the non-linear emulation of a response and the determination of an output stimulus) in the spiking neuron is much more complicated. The per-neuron complexity and large scale (i.e., millions of spiking neurons) of SNN from two aspects significantly increase the requirement of the computing power. Although high-performance hardware such as field-programmable gate arrays (FPGA) [10–12] can speed up the computing by data parallel and operation acceleration, the high computational complexity of every spiking neuron is hard to reduce.

* Corresponding author.
E-mail addresses: yanhuchen@bupt.edu.cn (Y. Chen), cenwang@bupt.edu.cn (C. Wang), hxguo@bupt.edu.cn (H. Guo), xionggao@bupt.edu.cn (X. Gao), jianwu@bupt.edu.cn (J. Wu).

https://doi.org/10.1016/j.neucom.2022.02.004
0925-2312/© 2022 Elsevier B.V. All rights reserved.

On the other hand, through well designing of a quantum algorithm on a quantum computer, it is able to reduce the computational complexity depending on the superposition and entanglement natures of the quantum. It is a more cost-effective solution than increasing the computing power per unit area on the processors (which will be eventually stopped by the quantum tunneling effect[13]). Considering such an advantage, in recent years, multi-type ANNs have been implemented on quantum computers (the details are illustrated in Section 2) to reduce the computational complexity and further expand the scale of neural networks [14–25]. However, the quantum efficiency for advanced neural netoworks, the SNN is not seen to be explored in these works.

Inspired by the aforementioned quantum neural networks, in this paper, we propose a quantum spiking neural network (QSNN) architecture. More specifically, we utilize a quantum counterpart to replace the key step that has the highest computing complexity in each spiking neuron of the SNN. Through mathematical derivation, we verify that the key step can be transformed into the operation of vector inner products, which can be accelerated on a quantum computer. By speeding up each spiking neuron, the whole SNN can be accelerated. Considering these, we mainly do the following contributions:

(1) To our best knowledge, we are the first to propose a QSNN: using a quantum algorithm to accelerate the classic spiking neural network. Correspondingly, we design the quantum circuit of QSNN to adapt learning requirements with arbitrary data dimension and data accuracy.
(2) We give the mathematical proof for the minimum success probability and calculation accuracy of SNN implemented on a quantum computer which are the key performance indicators of a quantum algorithm, and then provide an approach to further improve the minimum success probability of QSNN.
(3) Based on the analysis of quantum gate complexity combined with the minimum success probability, we verify that the computational complexity of QSNN is log-polynomial, which is much lower than the linear computational complexity of the classic SNN.

Additionally, to verify the feasibility and robustness of QSNN, we do two experiments based on two real-world datasets both with two different noise levels (noise-free and 50% of the noise), MNIST and fashion MNIST. The experimental results suggest that the QSNN has good robustness and shows the equal accuracy as the SNN.

2. Related work

Dating back to 1995, KaK et al. first proposed the concept of quantum neural computation. They suggested that it is a novel research field to combine neural computing and quantum computing to form a new paradigm. In 2000, Narayanan et al. [15] proposed quantum artificial neural network (QANN) architectures and compared the performance of quantum/classic neural networks. They concluded that the QANN has overwhelming advantages in both computing speed and model scale. Since then, deep learning models based on quantum computing are actively and widely studied.

Mainstream researches of QANNs are to implement feedforward neural networks by designing quantum circuits. Kouda et al. [16] proposed to use the quantum phase for the implementation of the qubit neural network and discussed its learning efficiency. Tacchino et al. proposed a quantum perceptron [20] and further constructed a multilayer quantum neural network based on the quantum perceptron [21]. Killoran et al. [22] proposed a continuous-variable QANN. Beer et al. [25] proposed an approach for training deep QANNs. In addition, as the modern neural networks vary in architectures, some researchers also explored quantum versions of the special architectures. For example, Cong et al. [23] proposed a quantum convolutional neural network (QCNN). QCNN can be used not only in the classic pattern recognition but also in devising a novel quantum error correction scheme. Rebentrost et al. [26] proposed a quantum Hopfield neural network (QHNN), QHNN can be used to store quantum data and even recognize virus RNA sequences. Pierre-Luc et al. [19] proposed a quantum generative adversarial network (QGAN). Then, Zoufal et al. [24] used QGAN for learning and loading quantum states of random amplitude in polynomial complexity with the number of qubits. Romero et al. [18] proposed a quantum autoencoder to efficiently compress the quantum data. In order to build a more generalized QANN model, Benedetti et al. [27] creatively introduced the gradient descent method to update the parameters of the quantum circuits.

3. Quantum spiking neural network model

The SNNs have multiple variations, for example, the diverse implementations of spiking neuron and backpropagation methods (e.g., spike-timing-dependent plasticity [28]), and the different precision of the emulations (e.g., dendritic event-based processing algorithm [29,8]). Our work follows Tempotron [6], a kind of SNNs works in a supervised manner with backpropagation and takes a leaky integrate-and-fire (LIF) spiking neuron as a basic unit. The feedforward and feedback propagation of Tempotron is described in Section 3.1.

3.1. Learning processing

In the spiking neuron architecture of Tempotron, as shown in Fig. 1, given an observation period T, each synapse receives the input stimuli sequences and changes in the potential as the responses. The soma of the spiking neuron is connected to all synapses, and each connection has a tunable weight parameter. During T, the potential of the soma is the weighted sum of the spikes on the synapses and can be described in Eq. (1).

$$V(t) = \sum_{k} w_k \sum_{t_i} K(t - t_i) + V_{rest} \tag{1}$$

where, V_{rest} denotes the rest potential, defaulted $V_{rest} = 0; w_k$ denotes the weight of the k-th synapse; t_i denotes the moment that the i-th stimulus on the k-th synapse. At the same time, the k-th synapse releases a spike to the soma. Function $K(t - t_i)$ emulates the spike shape from each synapse to the soma as Eq. (2).

$$K(t - t_i) = V_0\left(e^{\frac{-(t-t_i)}{\tau}} - e^{\frac{-(t-t_i)}{\tau_s}}\right)H(t - t_i) \tag{2}$$

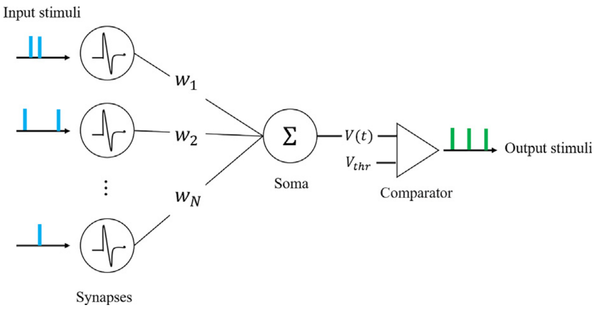


Fig. 1. A spiking neuron architecture.

where τ and τ_s are hyper-parameters and denote the decay time; V_0 is a normalized parameter, and $V_0 = \frac{\tau_s \tau}{\tau - \tau_s}$; $H(\cdot)$ is a Heaviside step function. Note that t denotes the current moment, t_i denotes the i-th stimulus input moment. Only when the potential $V(t^*)$ is over the threshold, the spiking neuron will produce an output stimulus at t^*. Once an output stimulus generates, the spiking neuron will have a refractory period. In this period, the spiking neuron does not react to any input stimuli. The top part in Fig. 2 shows that the spiking neuron receives the input stimuli from different synapses (represented as the horizontal slot) at different moments. Red and blue colors represent a positive and a negative pattern, respectively. The bottom part in Fig. 2 shows potential variations caused by these input stimuli. Only the potential stimulated by the positive input pattern grows beyond the threshold V_{thr}.

For a forward propagation, when the spiking neuron fires, it outputs a stimulus which can be read out or be used as an input of the downstream spiking neurons. The outputs of the spiking neuron(s) in the final layer represent the learning result of one iteration. For example, for a binary classification task, if the final spiking neuron generates output stimuli, the input item is judged as “1” class, otherwise, as “0” class.

In backward propagation, the rule of weight updating is shown in Eq. (3).

$$\Delta w_i = \lambda \sum_{t_i < t_{max}} K(t_{max} - t_i) \tag{3}$$

where t_{max} denotes the moment that the spiking neuron reaches its maximum potential. $\lambda > 0$ is a hyperparameter. If the prediction is not close enough to the ground truth, the weight w_i of any i-th synapse will be turned Δw_i. The learning object is to minimize the loss function $\mathscr{L}$.

$$\mathscr{L} = \sum_{p \in postive} \left(V_{thr} - V^{(p)}(t_{max}) \right) + \sum_{n \in negtive} \left(V^{(n)}(t_{max}) - V_{thr} \right) \tag{4}$$

3.2. Adjusting SNN model to adapt quantum computing

We find that the key step with the highest computational complexity of SNN is in the forward propagation, that is to find out the fire moment of a spiking neuron. In order to do so, a spiking neuron should capture every moment and value when the potential arrives at a local maximum, and then judges whether the local maximum value crosses the threshold V_{thr}. The most intuitive way is using the derivative of $V(t)$ for t to obtain all the local maximum potentials $V(t_{lmax})$ of the spiking neuron and their corresponding moments t_{lmax}, then compare $V(t_{lmax})$ with V_{thr}. Through mathematical derivation, we verify that the localizations of the moments that the potential crosses the threshold can be transformed into the calculation of the unsigned inner products of two vectors, which can be accelerated by using quantum computers.

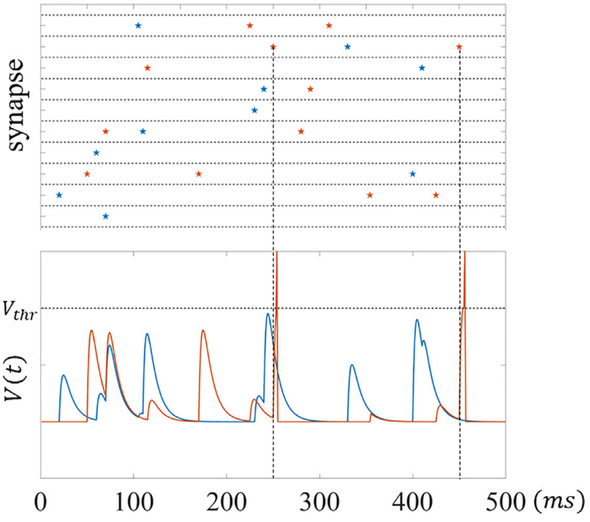


Fig. 2. The potential variations of a positive pattern (i.e., with threshold crossing) and a negative pattern (i.e., without threshold crossing), which are drawn by red and blue color, respectively, in a spiking neuron during the observation period. The localization of a star denotes a moment that the corresponding synapse receives a stimulus.

Assuming that the total number of stimuli in an input stimuli sequence is J_a, the set of the arrived moments of the stimuli is $D = \{t_1, t_2, \cdots t_J \cdots t_{J_a}\}$ (any $t_J \leqslant T$) and the number of elements in D is $\|D\|$. Because an arrived stimulus at t_J does not affect the potential before t_J, we cannot directly seek the derivative of $V(t)$ in Eq. (1). However, we can seek a derivative of $V_J(t)$ in Eq. (5), which is a correct operation for the first J stimuli. Therefore, we actually need to compute $\|D\|$ derivative operations.

$$V_J(t) = \sum_{j=1}^{J} w_j K(t - t_j) \tag{5}$$

where w_j denotes the weight of the synapse received the j-th input stimulus and t_j denotes the arrival time the j-th input stimulus. Given the function $K(\bullet)$ in Eq. (2), the derivative of $V_J(t)$ with respect for t is shown as Eq. (6):

$$\frac{dV_J(t)}{dt} = V_0 \sum_{j=1}^{J} w_j \left(\frac{1}{\tau_s} e^{-\frac{t-t_j}{\tau_s}} - \frac{1}{\tau} e^{-\frac{t-t_j}{\tau}} \right) \tag{6}$$

Then, we use $t\prime_J$ to denote the moment of the J-th local maximum potential, under the J input stimuli.

$$t\prime_J = V_0 \left[\ln \frac{\tau}{\tau_s} + \ln \left(\sum_{j=1}^{J} w_j e^{\frac{t_j}{\tau_s}} \right) - \ln \left(\sum_{j=1}^{J} w_j e^{\frac{t_j}{\tau}} \right) \right] \tag{7}$$

Becasue the summation operation in Eq. (7) can be rewritten using vectors, as shown in Eq. (8) the problem of finding the local maximum potential can be converted to an operation of calculating the inner product of two vectors.

$$t\prime_J = V_0 \left[\ln \frac{\tau}{\tau_s} + \ln (\vec{w}^T \vec{t_{\tau_s}}) - \ln (\vec{w}^T \vec{t_\tau}) \right] \tag{8}$$

where $\vec{w} = [w_1, w_2, \cdots, w_J]^T, \vec{t_{\tau_s}} = [e^{t_1/\tau_s}, e^{t_2/\tau_s}, \cdots, e^{t_J/\tau_s}]^T$ and $\vec{t_\tau} = [e^{t_1/\tau}, e^{t_2/\tau_s}, \cdots, e^{t_J/\tau}]^T$. Two constraints, $\vec{w}^T \vec{t_{\tau_s}} > 0$ and $\vec{w}^T \vec{t_\tau} > 0$, should be satisfied due to the $\ln(\bullet)$ function. Thus, the inner product of vector $\vec{w}$ and vector $\vec{t_{\tau_s}}$ or $\vec{t_\tau}$ must be unsigned.

From Eq. (7), we can see that to obtain $t\prime_J$ requires higher massive computing powers as the number of input stimuli increasing. To calculate $t\prime_J$, we need to obtain the unsigned inner product of $\vec{w}$ and $\vec{t_{\tau_s}}$, $\vec{w}$ and $\vec{t_\tau}$. To reduce computational complexity of this part which costs the highest computing powers in the classic SNN, and then to obtain further accelerations accordingly, we propose a **q**uantum algorithm to compute the **u**nsigned **i**nner **p**roduct of vectors (QUIP) and then use this algorithm to construct QSNN.

4. The Acceleration process of QSNN

Our QSNN aims to accelerate this process to obtain efficiency optimizations. Then, we further design the scalable quantum circuits of QSNN, which can adapt learning requirements with arbitrary data dimension and data accuracy. We also evaluate the success probability and calculation accuracy of QSNN. We then provide an approach to further improve the minimum success probability. Additionally, based on the gate complexity of the quantum circuits and the minimum success probability, we illustrate that QSNN can obtain an exponential acceleration compared to the classic SNN.

4.1. Quantum circuit of QUIP

The origin quantum approach to compute the unsigned inner product is swap-test [30]. The swap-test consists of three parts, an ancillary qubit and two quantum registers, as shown in Fig. 3. U_t and U_w can transform $|0\rangle^{\otimes\lceil\log_2 J\rceil}$ to the two vectors $|w\rangle, |t\rangle$, and the inner product of these two vectors are needed to be calculated:

$$U_w|0\rangle_w = |w\rangle \tag{9}$$

$$U_t|0\rangle_t = |t\rangle \tag{10}$$

where $|w\rangle = \left(\sum_{j=0}^{J-1} w_j^2\right)^{-\frac{1}{2}} \vec{w}, |t\rangle = \left(\sum_{j=0}^{J-1} e^{\frac{2t_j}{\tau_s}}\right)^{-\frac{1}{2}} \vec{t_{\tau_s}}$ or $|t\rangle = \left(\sum_{j=0}^{J-1} e^{\frac{2t_j}{\tau}}\right)^{-\frac{1}{2}} \vec{t_\tau}$. In the swap-test, an ancillary qubit is introduced. We will obtain the quantum state $|\Psi\rangle$, after performing the swap-test.

$$|\Psi\rangle = \frac{1}{2}[|0\rangle_{anc}(|w\rangle|t\rangle + |t\rangle|w\rangle) + |1\rangle_{anc}(|w\rangle|t\rangle - |t\rangle|w\rangle)] \tag{11}$$

Where, $|0\rangle_{anc}$ and $|1\rangle_{anc}$ are the two states of the ancillary qubit. The state probabilities of the ancillary qubit are $P(|0\rangle) = \frac{1}{2} + \frac{1}{2}|\langle w|t\rangle|^2, P(|1\rangle) = \frac{1}{2} - \frac{1}{2}|\langle w|t\rangle|^2$. We find that the state probabilities of the ancillary qubit contain our expected inner product $\langle w|t\rangle$.

Because the swap-test uses U_w, U_t to prepare the initial states of $|w\rangle, |t\rangle$ and $\lceil\log_2 J\rceil$ controlled-swap gates to compute the inner product. Inspired by the parameterized quantum circuits and generative adversarial network model [24], the complexity of preparing the initial quantum is polynomial relationship with the number of qubits. Therefore, the complexity of the swap-test is $O(\text{poly}\lceil\log_2 J\rceil)$. In classic computers, computing the inner product requires J multiplications, so the complexity is $O(J)$.

However, the measurement operator will cause the collapse of an ancillary qubit, and only 0 or 1 is the measurement result. To obtain the probability of each ancillary qubit state, we must repeatedly execute the swap-test and count the measurement result. If the acceptable calculation accuracy of ancillary qubit state probability requires an γ-digit binary floating-point number, 4^γ copies of swap-test will be run and measured (see Appendix A for the mathematical proof). These copies undoubtedly cancel the advantages of a quantum algorithm for computing inner products.

Considering these, to avoid the repeated executions, we propose QUIP in which the core idea is to combine the quantum amplitude estimation (QAE) algorithm [31,32] and the swap-test algorithm. Fig. 4 then shows the QUIP circuits containing a control register and a target register with m and $1 + 2\lceil\log_2 J\rceil$ qubits, respectively.

The specific steps of QUIP are as follows.

(1) Because $P(|0\rangle) + P(|1\rangle) = 1$, we here use a trick to parameterize the inner products using to θ, in this way we can obtain the specific expressions of $\langle w|t\rangle$. We define $P(|0\rangle) = \cos^2\theta$ and $P(|1\rangle) = \sin^2\theta$.

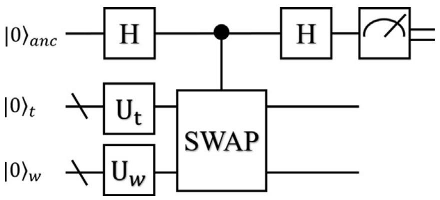


Fig. 3. The quantum swap-test circuit. The top qubit is an ancillary qubit, the next two are two quantum registers.

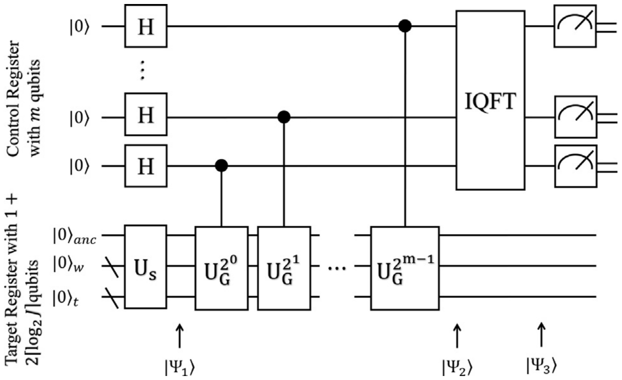


Fig. 4. The QUIP circuits for calculating and estimating $\langle w|t\rangle$. U_s denotes the swap-test operator; U_G denotes the generalized Grover operator; *IQFT* denotes the inverse quantum Fourier transformer.

Thus, after performing U_s, the quantum state of the target register is $|\Psi\rangle$ which equals to $[\cos\theta, \sin\theta]^T$. Further, in Eq. (12), we construct the mapping relationship between θ and $\langle w|t\rangle$:

$$\langle w|t\rangle = \sqrt{\cos 2\theta} \tag{12}$$

where $\theta \in [0, \frac{\pi}{4}]$. This is because $P(|0\rangle) = \frac{1}{2} + \frac{1}{2}|\langle w|t\rangle|^2, P(|1\rangle) = \frac{1}{2} - \frac{1}{2}|\langle w|t\rangle|^2$are probabilities between 0 and 1.

(2) As shown in Fig. 4, $|\Psi_1\rangle$ consists of an equal superposition state and $|\Psi\rangle$.

$$|\Psi_1\rangle = \frac{1}{\sqrt{2^m}}\left(\sum_{k=0}^{2^m-1}|k\rangle\right)|\Psi\rangle \tag{13}$$

(3) $|\Psi_2\rangle$ is obtained by applying several controlled U_G operators on $|\Psi_1\rangle$, as shown in Eq. (14):

$$|\Psi_2\rangle = \frac{1}{\sqrt{2^m}}\left(\sum_{k=0}^{2^m-1}|k\rangle U_G^k|\Psi\rangle\right) \tag{14}$$

where U_G denotes a generalized Grover operator [32,33]. U_G can be further divided into four sub-operators, the oracle operator O, the conditional phase shift operator I_0, the swap-test operator U_s and its inverse operator U_s^{-1}.A specific U_G operator is depicted in Fig. 5. U_G operator can be written as:

$$U_G = -U_s I_0 U_s^{-1} O = \begin{bmatrix} \cos 2\theta & \sin 2\theta \\ -\sin 2\theta & \cos 2\theta \end{bmatrix} \tag{15}$$

(4) Furthermore, the quantum phase kick-back has to be mentioned [34]. That is, if the input quantum state of the target register is one of the eigenvectors of the controlled operator, the

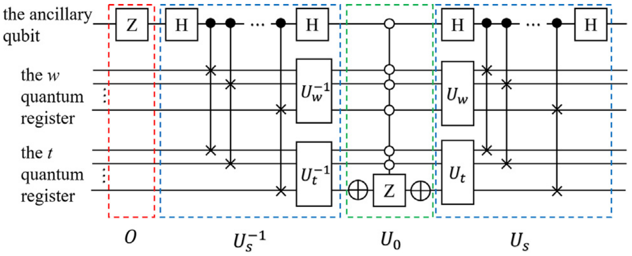


Fig. 5. The quantum circuits of the generalized Grover operator.

target register will not change and the control register will receive a phase that is equal to an eigenvalue of the corresponding eigenvector. The eigenvectors of U_G are written as $|\mu_1\rangle = [1, -i]^T, |\mu_2\rangle = [1, i]^T$, so that $|\Psi\rangle = e^{i\theta}|\mu_1\rangle + e^{-i\theta}|\mu_2\rangle$. Therefore,$|\Psi_2\rangle$ is rewritten as:

$$|\Psi_2\rangle = \frac{e^{i\theta}}{\sqrt{2^m}} \sum_{k=0}^{2^m-1} e^{i2k\theta}|k\rangle|\mu_1\rangle - \frac{e^{-i\theta}}{\sqrt{2^m}} \sum_{k=0}^{2^m-1} e^{-i2k\theta}|k\rangle|\mu_2\rangle \tag{16}$$

(5) Applying *IQFT* to $|\Psi_2\rangle$, we can obtain $|\Psi\rangle$:

$$|\Psi_3\rangle = e^{i\theta}|\frac{2^m\theta}{\pi}\rangle|\mu_1\rangle + e^{-i\theta}|2^m\left(1 - \frac{\theta}{\pi}\right)\text{mod}2^m\rangle|\mu_2\rangle \tag{17}$$

Ideally, when $2^m\theta/\pi$ is an integer,$|\frac{2^m\theta}{\pi}\rangle$ is a column vector, and its $2^m\theta/\pi$-th element is 1, others are 0. If $2^m\theta/\pi$ is an integer, then $2^m(1 - \frac{\theta}{\pi})\text{mod}2^m$ is also an integer. If $2^m\theta/\pi$ is not an integer, Eq. (17) will no longer be correct. We will discuss a general case in Section 4.2.

(6) Finally, we measure the control register. The measurement result may be one of the two values with equal probability, and a r is a binary value with m bits.

$$r = 2^m\frac{\theta}{\pi} \quad or \quad r = 2^m\left(1 - \frac{\theta}{\pi}\right)\text{mod}2^m \tag{18}$$

Because $\theta \in \left[0, \frac{\pi}{4}\right]$, except for $\theta = 0, 2^m\theta/\pi$ and $2^m(1 - \frac{\theta}{\pi})\text{mod}2^m$ have no overlapping values, which means that there is no ambiguity in one measurement. Thus, we can construct the mapping relationship between r and $\langle w|t\rangle$, as described in Eq. (19):

$$\langle w|t\rangle = \begin{cases} \sqrt{\cos\frac{r\pi}{2^{m-1}}} & 0 \leqslant r \leqslant 2^{m-2} \\ \sqrt{\cos\left(2\pi - \frac{r\pi}{2^{m-1}}\right)} & 3 \times 2^{m-2} \leqslant r \leqslant 2^m - 1 \end{cases} \tag{19}$$

In summary, in QUIP, according to the measurement of the quantum control register, we can obtain the result r, a binary integer with length m. Based on the Eq. (19), we can precisely calculate the $\langle w|t\rangle$, when $2^m\theta/\pi$ can be perfectly expressed as a binary integer with length m. In this way, we can efficiently obtain $\langle w|t\rangle$ through QUIP, while avoiding repeated executions of swap-test and statistical measurement results, thereby accelerating the SNN model.

4.2. Success probability and calculation accuracy

Although we have given the mapping relationship between r and θ, it is not perfect, since we require the measurement result r must be an integer. However, since θ changes consistently in $\left[0, \frac{\pi}{4}\right]$, and $2^m\theta/\pi$ before a measurement is not always an integer. The error between the actual r and the measurement $\widetilde{r}$ will propagate to the $\langle w|t\rangle$. It is necessary to analyze the error between the actual r and the measurement $\widetilde{r}$. Since the analyzations based on $2^m(1 - \frac{\theta}{\pi})\text{mod}2^m$ and $2^m\theta/\pi$ are similar, we only do that for $r = 2^m\theta/\pi$.

For sake of the generality, we assume that an estimated phase $\widetilde{\theta}$ and then get $\widetilde{r} = 2^m\theta/\pi$, where $\widetilde{r}$ is a binary integer with m bits. The actual value r is certainly not a binary integer with length m, due to the continuous variations of phase θ. We use a remainder Δr to represent the error term between $\widetilde{r}$ and actual r, as shown in Eq. (20):

$$r/2^m = \widetilde{r}/2^m + \Delta r = (0.r_1r_2\cdots r_m)_2 + \Delta r \tag{20}$$

where r_j is a binary value and $r_j \in \{0, 1\}$; the range of Δr is $\left[-2^{-(m+1)}, 2^{-(m+1)}\right)$ (i.e., $\widetilde{r} = \text{round}(r)$). Thus, we use the variable r to replace θ in the Eq. (16) and get Eq. (21).

$$\begin{aligned} |\Psi_2\rangle \quad &= \tfrac{e^{i\pi r/2^m}}{\sqrt{2^m}} \sum_{k=0}^{2^m-1} e^{i2\pi rk/2^m}|k\rangle|\mu_1\rangle + \\ &\tfrac{e^{-i\pi r/2^m}}{\sqrt{2^m}} \sum_{k=0}^{2^m-1} e^{-i2\pi rk/2^m}|k\rangle|\mu_2\rangle \end{aligned} \tag{21}$$

For simplification, because the two parts in $|\Psi_2\rangle$ are similar and the global phase does not affect the measurement, we only choose the left part of $|\Psi_2\rangle$ and ignore the global phase. Then, the quantum state $|S\rangle$ is described as Eq. (22).

$$|S\rangle = \frac{1}{\sqrt{2^m}} \sum_{k=0}^{2^m-1} e^{i2\pi rk/2^m}|k\rangle \tag{22}$$

The 2^m dimensional *IQFT* operator, $IQFT_m$, can be described as Eq. (23):

$$IQFT_m = \frac{1}{2^{\frac{m}{2}}} \begin{bmatrix} e^{-\frac{i2\pi(0\times0)}{2^m}} & e^{-\frac{i2\pi(0\times1)}{2^m}} & \cdots & e^{-\frac{i2\pi(0\times(2^m-1))}{2^m}} \\ e^{-\frac{i2\pi(1\times0)}{2^m}} & e^{-\frac{i2\pi(1\times1)}{2^m}} & \cdots & e^{-\frac{i2\pi(1\times(2^m-1))}{2^m}} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-\frac{i2\pi((2^m-1)\times0)}{2^m}} & e^{-\frac{i2\pi((2^m-1)\times1)}{2^m}} & \cdots & e^{-\frac{i2\pi((2^m-1)\times(2^m-1))}{2^m}} \end{bmatrix} \tag{23}$$

Now, applying $IQFT_m$ to $|S\rangle$, we can obtain the quantum state $|r\rangle$:

$$\begin{aligned} |r\rangle = IQFT_m|S\rangle = \tfrac{1}{2^m}&\left[\sum_{k=0}^{2^m-1} e^{i2\pi(rk-0\times k)/2^m},\right. \\ &\left.\sum_{k=0}^{2^m-1} e^{i2\pi(rk-1\times k)/2^m}, \quad \cdots, \quad \sum_{k=0}^{2^m-1} e^{i2\pi\left(rk-\left(2^m-1\right)\times k\right)/2^m}\right]^T \end{aligned} \tag{24}$$

From the geometric perspective, all column vectors of $IQFT_m$ can form a group of orthogonal bases, and $|r\rangle$ is the projections of $|S\rangle$ on all orthogonal bases of $IQFT_m$. According to Eq. (24), in the case of $\Delta r = 0$, the vector $|S\rangle$ is parallel to the r-th orthogonal basis and $|S\rangle$ is orthogonal to other bases. This ideal case has been displayed in Eq. (17). Thus, the r-th element of $|r\rangle$ is 1 and others are 0 which means the measurement result is fixed. In another case, if $\{\Delta r \neq 0$ and $\Delta \in \left[-2^{-m+1}, 2^{-(m+1)}\right)\}$, $|S\rangle$ is not orthogonal or parallel to any orthogonal basis, but $|S\rangle$ has projections on all orthogonal bases. The square of projection length on the h-th orthogonal basis is the probability of the measurement h.Therefore, the measurement result of $|r\rangle$ may be any integer from $\left[0, 2^m - 1\right]$. Let the probability of the measurement h be p_h, which is described in Eq. (25).

$$\begin{aligned} p_h = |\langle h|S\rangle|^2 = \tfrac{1}{2^{2m}}&\left|\left(\sum_{k=0}^{2^m-1} e^{-i2\pi(h\times k)/2^m}|k\rangle\right)\times\right. \\ &\left.\left(\sum_{k=0}^{2^m-1} e^{i2\pi(r\times k)/2^m}|k\rangle\right)\right|^2 = \tfrac{1}{2^{2m}}\left|\sum_{k=0}^{2^m-1} e^{-i2\pi(h-r)k/2^m}\right|^2 \end{aligned} \tag{25}$$

It is obvious to see that the summation in the right part of Eq. (25) is a geometric series. According to the summation formula of the geometric series, we can further simplify p_h in Eq. (26).

$$p_h = \frac{1}{2^{2m}}\left|\frac{1 - e^{-i2\pi(h-r)}}{1 - e^{-i2\pi(h-r)/2^m}}\right|^2 \tag{26}$$

Based on p_h, we can define the success probability and the calculation accuracy for QUIP:

Definition 1 (the success probability): The success probability $p_{\widetilde{r}}$ is defined as the probability which the measurement result of quantum control register is $\widetilde{r}$, as shown in Eq. (27).

$$p_{\widetilde{r}} = \frac{1}{2^{2m}}\left|\frac{1 - e^{-i2\pi\Delta r\times2^m}}{1 - e^{-i2\pi\Delta r}}\right|^2 \tag{27}$$

Definition 2 (the calculation accuracy): The calculation accuracy ε is defined as an absolute value of using the actual local maximum potential $\langle w|t\rangle$ minus the estimated local maximum potential $\widetilde{\langle w|t\rangle}$, when the measurement result is $\tilde{r}$, as shown in Eq. (28).

$$\varepsilon = \left|\langle w|t\rangle - \widetilde{\langle w|t\rangle}\right| \tag{28}$$

We first discuss $p_{\tilde{r}}$. Because the formats of the numerator and denominator in Eq. (26) are similar, we take the denominator as a case to show the trick for simplifying $p_{\tilde{r}}$. We treat the denominator as a vector difference on the complex number field which is highlighted in red in Fig. 6(a). Besides, because the two vectors, 1 and $e^{-i2\pi\Delta r}$, have the same length, the two vectors and their difference together form an isosceles triangle. The value of the denominator equals the vector difference and can be simply calculated as shown in Fig. 6(b). In this way, we can give the simplified measurement probability $p_{\tilde{r}}$ when the $\Delta r \neq 0$ and $\Delta r \in \left[-2^{-(m+1)}, 2^{-(m+1)}\right)$, as described in Eq. (29).

$$p_{\tilde{r}} = \frac{1}{2^{2m}}\frac{\sin^2(2^m\pi\Delta r)}{\sin^2(\pi\Delta r)} \tag{29}$$

Then, we try to find the minimum success probability, which is an indicator of the feasibility of QSNN. We can get the minimum success probability using the derivative of $p_{\tilde{r}}$.

$$\frac{\mathrm{d}p_{\tilde{r}}}{\mathrm{d}\Delta r} = \frac{\pi\sin^2\left(2^{m+1}\pi\Delta r\right)}{2^m\sin^2(\pi\Delta r)} - \frac{2\pi\sin^2(2^m\pi\Delta r)\cos(\pi\Delta r)}{2^{2m}\sin^3(\pi\Delta r)} \tag{30}$$

Considering the range of Δr, we can obtain the minimum $p_{\tilde{r}}$ in Eq. (31).

$$\min p_{\tilde{r}} = \frac{1}{2^{2m}}\frac{\sin^2(2^m\pi\Delta r)}{\sin^2(\pi\Delta r)} \approx \frac{\sin^2(2^m\pi\Delta r)}{2^{2m}(\pi\Delta r)^2} = \frac{4}{\pi^2} \tag{31}$$

Next, we discuss ε. The error ε is related to $\tilde{r}$ and thus has a relationship with Δr, when m is deterministic. Similarly, by computing $\frac{d\varepsilon}{d\Delta r}$, we can obtain the maximum error, when $\Delta r \rightarrow 2^{-(m+1)}$ or $\Delta r \rightarrow -2^{-(m+1)}$.

$$\max\varepsilon = \left|\sqrt{\cos\left(\frac{\pi\tilde{r}}{2^{m-1}} + \frac{\pi}{2^{2(m-1)}}\right)} - \sqrt{\cos\frac{\pi\tilde{r}}{2^{m-1}}}\right| \tag{32}$$

In summary, in this section, we have completely analyzed the success probability and calculation accuracy of QUIP, which are the key performance of QSNN. Thus far, the original minimum success probability $p_{\tilde{r}}$ for finding the local maximum potential is $\frac{4}{\pi^2}$. The maximum error ε has a mapping relationship with the number of qubits m and the measurement $\tilde{r}$ of the control register.

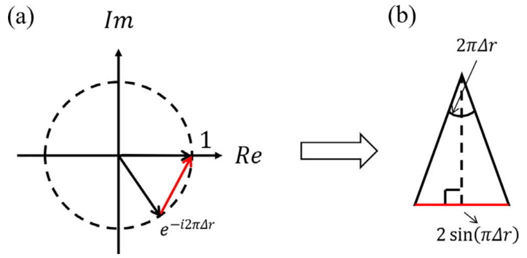


Fig. 6. The diagram of symbolic-graphic combination for explaining the simplified process of $p_{\tilde{r}}$.

4.3. Improvement of minimum success probability

In Section 4.2, we give the lower bound of the success probability and the maximum error, as shown in Eq. (27), (28) respectively. However, the success probability is too low (i.e., only $\frac{4}{\pi^2}$) to be acceptable. In this section, we want to further improve the minimum success probability, even if the calculation accuracy gets a slight deterioration. The accuracy loss can be compensated by increasing the number of qubits in control register.

Specifically, we use two approaches to improve the minimum success probability together. First, we now introduce a slack condition, we consider $\tilde{r}+1, \tilde{r}$, and $\tilde{r}-1$ as the correct result not just $\tilde{r}$. When m is relatively larger, r becomes larger accordingly, and this slack condition may cause a slight deterioration of the calculation accuracy. The probability p to get $\tilde{r}+1, \tilde{r}$, and $\tilde{r}-1$ in a measurement of the control register is described as Eq. (33).

$$p = p_{\tilde{r}+1} + p_{\tilde{r}} + p_{\tilde{r}-1} \approx \frac{\sin^2\left[2^m\pi\left(\Delta r+2^{-m}\right)\right]}{2^{2m}\left[\pi\left(\Delta r+2^{-m}\right)\right]^2} + \frac{\sin^2(2^m\pi\Delta r)}{2^{2m}(\pi\Delta r)^2} + \frac{\sin^2\left[2^m\pi\left(\Delta r-2^{-m}\right)\right]}{2^{2m}\left[\pi\left(\Delta r-2^{-m}\right)\right]^2} \tag{33}$$

Similar to Eq. (31), we can obtain the minimum value of p, when $\Delta r \rightarrow 2^{-(m+1)}$ or $\Delta r \rightarrow -2^{-(m+1)}$.

$$\min p = \frac{76}{9\pi^2} \tag{34}$$

Additionally, repeatedly running the QUIP and measuring the control register q times can help further improve the final success probability p_q which is described as Eq. (35):

$$p_q = \sum_{k=\lfloor\frac{1}{2}q\rfloor+1}^{q} C_q^k p^k (1-p)^{q-k} \tag{35}$$

where $C_q^k = \frac{q!}{k!(q-k)!}$, p_q denotes the success probability of obtaining the correct result from at least $\frac{q}{2}$ tests in repeated q times. Obviously, if $p \geqslant 0.5, p_q$ will must be greater than p. This approach will cause p_q close to 1 using a few copies of QUIP and the measurements, which especially is suitable for the case that the initial probability of a certain random event is greater than 0.5.

4.4. The complexity of QSNN

For classic computers, the complexity for calculating an inner product of two vectors is $O(J)$, where J is the dimension of each vector. Given the number of vector pairs as a constant s (i.e., is related to the number of stimuli, layers and epochs of SNN), in the neural network model, the main complexity of SNN is $O(sJ)$. As for QSNN, we want to obtain the gate complexity of QUIP which is used to calculate the vector inner product. According to proof on complexity of quantum phase estimation (QPE) algorithm proposed by Arvid, the complexity of QPE is polynomial to the complexity of its embedded black box operator Q [35]. In our quantum circuits, the Q is U_G. Since QPE and QAE share the same architecture, we can use the analysis of QPE's complexity for obtaining complexity of QAE. Therefore, we only need to further analyze the complexity of U_G. As for the four sub-operators of a U_G, the complexity of O operator is $O(1)$; the complexity of U_0 operator (i.e., the complexity of a multi-controlled operator) is $O\left(\lceil\log_2 J\rceil^2\right)$ through adding the corresponding number of extra ancillary qubits [31]; the U_s operator includes $\lceil\log_2 J\rceil^2$ controlled-swap gates, U_w and U_t. U_w and U_t are essentially two operators used to prepare the initial quantum state $|w\rangle$ and $|t\rangle$, respectively. According to the previous studies, a polynomial complexity between the problem of preparing an initial quantum state and the number of qubits [24]. Therefore, in total, the complexity of a single test of QUIP is $O(\text{poly}\lceil\log_2 J\rceil)$. Considering the s, the complete complexity is $O(s \times \text{poly}\lceil\log_2 J\rceil)$. Consequently, the computational complexity of QSNN is log-polynomial and is much lower than the linear computational complexity of the classic SNN.

5. Numerical simulation and applied experiment

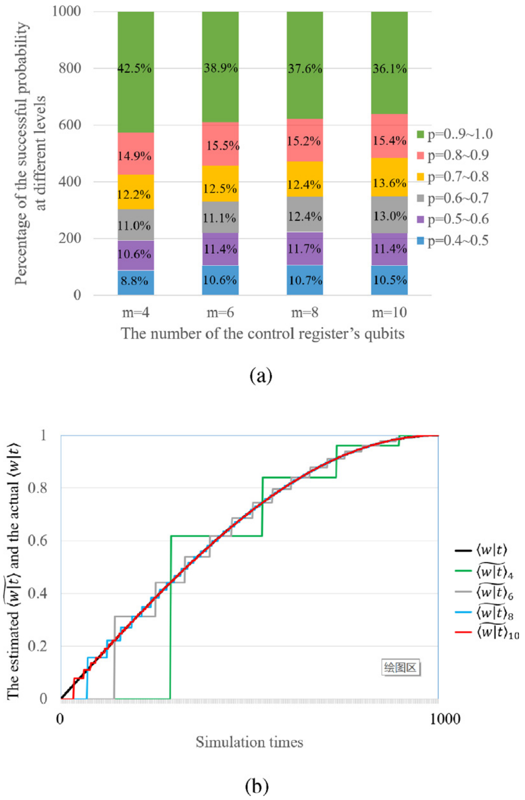
In this section, we first verify the correctness of the Eqs. (27, 28, 35), which are the key steps of QSNN, through a group of numerical simulations; Then, to verify the feasibility of QSNN, we do an image classification task using a real-world dataset.

5.1. Numerical Simulation

The schemes for the numerical simulations on the success probability and calculation accuracy of QSNN are as follows:

(1)We randomly generate 1000 pairs of $|w\rangle, |t\rangle$, and ensure their inner product result distribute in $[0,1]$, thus the numerical simulations can completely reflect the variations of $|w\rangle, |t\rangle$.
(2)We set m qubits for the control register in QUIP, where $m = 4, 6, 8, 10$. The simulation results of the success probability and the calculation accuracy are shown in Fig. 7.
(3)To verify the approaches proposed in Section 4.3 to further improve the success probability, we set the repetitions of QUIP, q = 1,3,5,7,9,11. The simulation results of the success probability improvements are shown in Fig. 8.

Fig. 7(a) shows the success probability when the control register gives the correct result under different m. It can be concluded that, on one hand, the simulation results are in accord with the theoretical proof in Eq. (29), that is, the success probabilities of all $\langle w|t\rangle$ are greater than $0.4(\approx 4/\pi^2)$. On the other hand, the number of control register qubits does not affect the success probability of QUIP. Fig. 7(b) then shows the significant improvements of the calculation accuracy with the growth of m. From Fig. 7(b), the control register can meet the usable calculation accuracy without a large number of qubits. Fig. 8 shows the success probability that Δr changes within the scope of the definition, i.e., $\Delta r \in \left[-2^{-(m+1)}, 2^{-(m+1)}\right)$. For the worst instance, when $q = 11, p = \frac{76}{9\pi^2}$, then the final success probability $p_q \approx 0.998$.



(a)

(b)

Fig. 7. The diagram of the simulation results of 1000 pairs of vector inner products with $m = 4, 6, 8, 10$. (a) The results of $\langle X|Y\rangle$ and $\langle X|Y\rangle_m$. (b) The proportion of different successful probability.

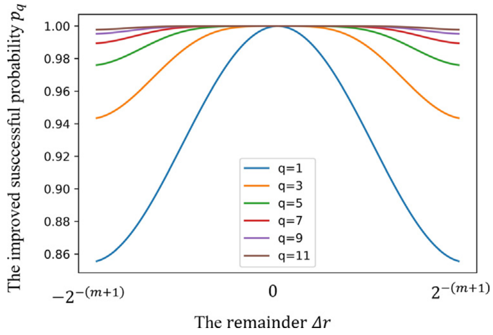


Fig. 8. The simulation results of the success probability improvements.

5.2. Image classification by QSNN

To verify the feasibility and robustness of QSNN, we implement QSNN on image classification using two datasets from real-world, "MNIST" and "fashion MNIST", with zero noise and 50% random noise, respectively. this experiment, we only expect to show that the QSNN accelerates the task with no performance loss compared to classic SNN. Thus, we simplify the original multiple classification task into a binary classification task. According to whether one spiking neuron outputs a stimulus, QSNN can classify the image into a positive or negative case. For MNIST, the image whose label is '0' is the positive case, and the image whose label is '1' is the negative case. For fashion MNIST, the image whose label is 'shirt' is the positive case and the image whose label is 'trouser' is the negative case.

We briefly illustrate how to encode an image into stimuli as the inputs of QSNN. First, we transform a gray image into a binary image. Second, we input an image with 28×28 pixels into a pooling layer and get an image with 14×14 pixels. Then, we padding one row and one column to get an image with 15×15 pixels. Third, as shown in Fig. 9, each 3×3 pixels is regarded as a binary string of length 9. We convert each binary string to a decimal value as the time point of an input stimulus, following the Eq. (36). We set that each synapse only has one input stimulus; thus, an image with 15×15 pixels is transferred to 25 stimuli on 25 synapses.

$$t_{inp} = \sum_{k=0}^{8} 2^k \alpha_k \tag{36}$$

where t_{inp} denotes the time point of a stimulus; α_k denotes the k-th pixel value (i.e. 0 or 1).

The experiment results for the two datasets are shown in Fig. 10. we can see that the performance of QSNN is approximately equal to the performance of SNN. Consequently, compared with the original SNN model, we can verify that QSNN has the same performance in real-world image classifications.

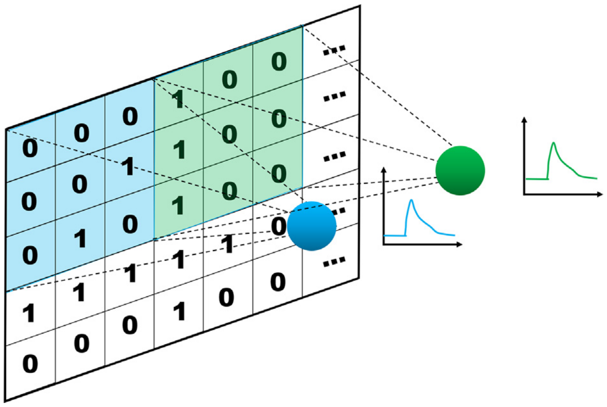


Fig. 9. The encoding scheme for images of handwritten characters.

6. Discussion

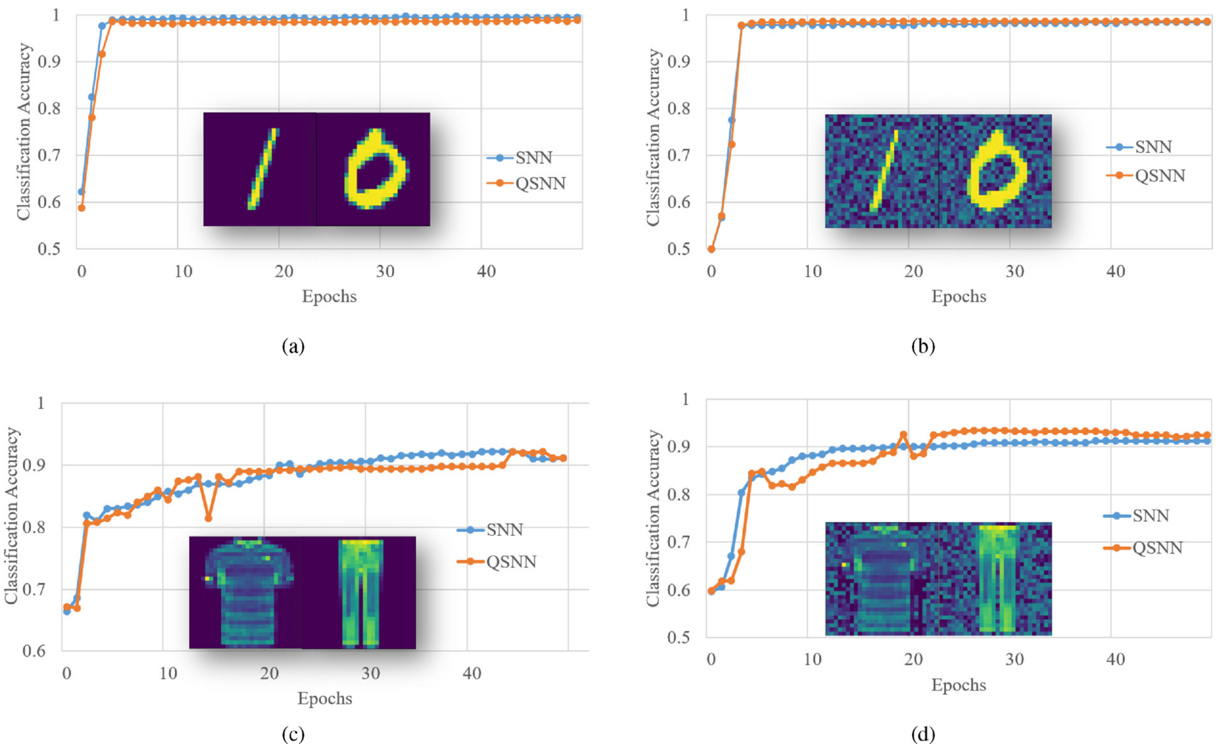
Though our QSNN is based on a specific SNN model, Tempotron, the idea of using quantum computers to accelerate other kinds of SNN models is also feasible. According to the aforementioned description, the SNN models which can be accelerated by using quantum computers need to satisfy the following characteristics: (1) The basic unit of the SNN models is a LIF spiking neuron. (2) The spiking neuron generates stimuli according to whether the local maximum potential crosses the threshold potential.

The other learning rules of SNNs adjust the weight according to the features of the stimuli. For example, Spike-Timing-Dependent Plasticity (STDP) algorithm requires the information of two or more stimuli intervals [36]. On the other hand, some SNN models do not use the local maximum potential. For example, the dendritic event-based processing (DEP) algorithm utilizes average membrane voltage over a period of time as an indicator to adjust the weight [8,29]. Thus, its acceleration approach using quantum computers needs to further explore.

Additionally, we have noticed that there are some studies based on large-scale hardware platforms with Non-von Neumann architecture to simulate different regions of a human brain [10–12], including designing the novel structure of SNNs and optimizing the routes of information communication between different regions of a simulated human brain. Such novel works can be seen as the extension of the neuromorphic computing, which can inspire our future works, for example, to find a quantum approach to emulate the specific brain region, or to construct a classic-quantum mixture hardware platform to combine our QSNN model with other novel neuromorphic computing structures.

7. Conclusion

In this paper, we proposed QSNN, in which we design and implement a quantum computing algorithm for the moment localizations of threshold crossings with the highest computational complexity in the classic SNN. More specifically, we proposed the QUIP algorithm to calculate the unsigned inner products converted from the moment localizations of threshold crossings. Then, We gave complete proof of the original success probability and calculation accuracy of QSNN. To further improve the minimum success probability, we used a slack condition and slight repetitions to improve the minimum success probability to nearly 100%. Through analysis, it can be verified that, QSNN outperforms the classic SNN in computational complexity. The computational complexity of QSNN is log-polynomial to the data dimension, while the computational complexity of the classic SNN is linear to the data dimension. This suggests that our quantum algorithm can significantly accelerate SNNs, even for those with extremely high input dimensions. To verify our mathematical proof of the minimum success probability and the calculation accuracy, we do numerical simulations by configuring different numbers m of qubits in the control regis-



(a)

(b)

(c)

(d)

Fig. 10. The comparison of SNN and QSNN on MNIST and fashion MNIST datasets with free noise and 50% random noise. The number of iterative is 50. (a) MNIST with free noise. (b) MNIST with 50% random noise. (c) Fashion MNIST with free noise. (d) Fashion MNIST with 50% random noise.

ter. We found that m not impact the minimum success probability, however, can significantly improve the calculation accuracy. This suggests that our QSNN can meet the requirements of arbitrary computational accuracy. We also gradually increase the repetitions to see the improvements of the minimum success probability, and we found that when the repetitions $q = 11$, the minimum success probability reaches 0.998. Additionally, we use our QSNN to solve the real-world image classification tasks on two datasets with different noise levels. The experimental results verify the feasibility and the robustness of our QSNN.

Code Availability Statement

Code and datasets for the QSNN model can be obtained from a GitHub repository (https://github.com/yanhuchen/QSNN).

CRediT authorship contribution statement

Yanhu Chen: Conceptualization, Methodology, Software. **Cen Wang:** Writing - review & editing. **Hongxiang Guo:** Writing - review & editing. **Xiong Gao:** Methodology, Supervision. **Jian Wu:** Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Measurement accuracy of quantum state probabilities in the swap-test

The amplitude of a quantum state is usually used to store data information. The square of amplitude is measurement probability. In many circumstances, we want to know the probability. For instance, in this paper, we want to know the probability $P(|1\rangle)$ of the ancillary qubit of the swap-test. However, when we measure the ancillary qubit, it will collapse to 0 or 1with the probability of $P(|0\rangle)$ or $P(|1\rangle)$, respectively. In this section, we give mathematical proof for the relationship between the accuracy of an ancillary qubit state probability $P(|1\rangle)$ and the number of copies n_a of the swap-test.

In order to obtain the probability of a qubit state, we must repeatedly execute the swap-test and count the measurement result. Without loss of generality, we respectively regard each execution of the swap-test and measurement as a random event and an independent binomial distributed variable. Let the number of random events be n_a; let the set of variables be $X = \{x_1, x_2, \cdots, x_{n_a}\}$. Then the binomial distribution is represented $S_{n_a} \sim B(n_a, P(|1\rangle))$. According to De Moivre-Laplace central limit theorem [37], if n_a is large enough, for any real number $a, b (a < b)$, the probability $P(a < \sum_{k=1}^{n_a} x_k \leqslant b)$ that $\sum_{k=1}^{n_a} x_k$ is in $(a, b]$ is described as Eq. (37):

$$P\left(a < \sum_{k=1}^{n_a} x_k \leqslant b\right) \approx \Phi\left(\frac{b - n_a P(|1\rangle)}{\sqrt{n_a P(|1\rangle) P(|0\rangle)}}\right) - \Phi\left(\frac{a - n_a P(|1\rangle)}{\sqrt{n_a P(|1\rangle) P(|0\rangle)}}\right) \tag{37}$$

where $\Phi(x)$ is the standard Gaussian distribution function which is described as Eq. (38).

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{y^2}{2}} \mathrm{d}y \tag{38}$$

In swap-test, the interval $(a, b]$ represents the calculation accuracy;$P(a < \sum_{k=1}^{n_a} x_k \leqslant b)$ represents the probability that $n_a P(|1\rangle) \in (a, b]$. Let $a = (1 - \delta) n_a P(|1\rangle)$and $b = (1 + \delta) n_a P(|1\rangle)$. Thus, the calculation accuracy is expressed by δ. Eq. (37) is rewritten as Eq. (39).

$$P\left(a < \sum_{k=1}^{n_a} x_k \leqslant b\right) \approx 2\Phi\left(\delta \sqrt{n_a} \sqrt{\frac{P(|1\rangle)}{P(|0\rangle)}}\right) \tag{39}$$

We conclude that δ and n_a^2 are inversely proportional, under the same probability condition from equation (A3).

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 2015..

[3] Wolfgang Maass, Networks of spiking neurons: the third generation of neural network models, Neural Networks 10 (9) (1997) 1659–1671.

[4] Filip Ponulak, Andrzej Kasinski, Introduction to spiking neural networks: Information processing, learning and applications, Acta neurobiologiae experimentalis 71 (4) (2011) 409–433.

[5] Qiang Yu, K.C. Tan, Huajin Tang, Pattern recognition computation in a spiking neural network with temporal encoding and learning. In International Joint Conference on Neural Networks, 2012..

[6] Rorü, The tempotron: a neuron that learns spike timing-based decisions, Nature Neurosci. (2006).

[7] C. Teeter, R. Iyer, V. Menon, N. Gouwens, D. Feng, J. Berg, A. Szafer, N. Cain, H. Zeng, M. Hawrylycz, Generalized leaky integrate-and-fire models classify multiple neuron types, Nature Commun. 9 (1) (2018) 709.

[8] Shuangming Yang, Tian Gao, Jiang Wang, Bin Deng, Benjamin Lansdell, Bernabe Linares-Barranco, Efficient spike-driven learning with dendritic event-based processing, Front. Neurosci. 15 (2021) 97.

[9] Shuangming Yang, Jiang Wang, Nan Zhang, Bin Deng, Yanwei Pang, and Mostafa Rahimi Azghadi. Cerebellumorphic: Large-scale neuromorphic model and architecture for supervised motor learning. IEEE Trans. Neural Networks Learn. Syst., 2021..

[10] S. Yang, J. Wang, B. Deng, C. Liu, H. Li, C. Fietkiewicz, K.A. Loparo, Real-time neuromorphic system for large-scale conductance-based spiking neural networks, IEEE Trans. Cybern. (2018) 1–14.

[11] B Yang, S. Deng, J. Wang, H. Li, M. Lu, Y. Che, X. Wei, and K.A. Loparo. Scalable digital neuromorphic architecture for large-scale biophysically meaningful neural network with multi-compartment neurons. IEEE Trans. Neural Networks Learn. Syst., pages 1–15, 2019..

[12] S. Yang, J. Wang, X. Hao, H. Li, and K.A. Loparo. Bicoss: Toward large-scale cognition brain with multigranular neuromorphic architecture. IEEE Trans. Neural Networks Learn. Syst., PP(99):1–15, 2021..

[13] Mohsen Razavy. Quantum Theory of Tunneling (second ed.). World Scientific, 2014.

[14] Subhash C. Kak, Quantum Neural Computing, Syst. Control Inform. 52 (3–4) (1995) 143–160.

[15] Ajit Narayanan, Menneer, Tammy, Quantum artificial neural network architectures and components, Inform. Sci. (2000).

[16] Noriaki Kouda, Nobuyuki Matsui, Haruhiko Nishimura, Ferdinand Peper, Qubit neural network and its learning efficiency, Neural Comput. Appl. 14 (2) (2005) 114–121.

[17] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, Seth Lloyd, Quantum machine learning, Control Theory Appl. 549 (7671) (2017) 195.

[18] Jonathan Romero, Jonathan P. Olson, Alan Aspuru-Guzik, Quantum autoencoders for efficient compression of quantum data, Quantum Sci. Technol. 2 (4) (2017) 045001.

[19] Pierre-Luc Dallaire-Demers, Nathan Killoran, Quantum generative adversarial networks, Phys. Rev. A 98 (1) (2018) 012324.

[20] Francesco Tacchino, Chiara Macchiavello, Dario Gerace, Daniele Bajoni, An Artificial Neuron Implemented on an Actual Quantum Processor. npj Quantum Inform..

[21] Francesco Tacchino, Panagiotis Barkoutsos, Chiara Macchiavello, Ivano Tavernelli, Dario Gerace, and Daniele Bajoni. Quantum implementation of an artificial feed-forward neural network. 2019..

[22] Nathan Killoran, Thomas R. Bromley, Juan Miguel Arrazola, Maria Schuld, and Seth Lloyd. Continuous-variable quantum neural networks. Phys. Rev. Res., 1 (3), 2019..

[23] Iris Cong, Soonwon Choi, Mikhail D. Lukin, Quantum convolutional neural networks, Nature Phys. 15 (12) (2019) 1273–1278.

[24] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. Quantum Generative Adversarial Networks for learning and loading random distributions. npj Quantum Inform., 5(1):103, 2019..

[25] Kerstin Beer, Dmytro Bondarenko, Terry Farrelly, Tobias J. Osborne, Robert Salzmann, Daniel Scheiermann, Ramona Wolf, Training deep quantum neural networks, Nature Commun. 11 (1) (2020) 808.

[26] Patrick Rebentrost, Thomas R. Bromley, Christian Weedbrook, Seth Lloyd, Quantum Hopfield neural network, Phys. Rev. A 98 (4) (2018) 042308.

[27] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. 2019..

[28] Natalia Caporale, Yang Dan, Spike timing–dependent plasticity: a hebbian learning rule, Annu. Rev. Neurosci. 31 (2008) 25–46.

[29] Jordan Guerguiev, Timothy P. Lillicrap, Blake A. Richards, Towards deep learning with segregated dendrites, ELife 6 (2017) e22901.

[30] Harry Buhrman, Richard Cleve, John Watrous, Ronald De Wolf, Quantum fingerprinting, Phys. Rev. Lett. 87 (16) (2001) 167902.

[31] Michael A. Nielsen, Isaac L Chuang, Quantum Computation and Quantum Information, 10th Anniversary Edition., Cambridge University Press, 2011.

[32] Gilles Brassard, Peter Høyer, Michele Mosca, Alain Tapp, Quantum amplitude amplification and estimation, in: Samuel J. Lomonaco, Howard E. Brandt (Eds.), Contemporary Mathematics, vol. 305, American Mathematical Society, Providence, Rhode Island, 2002, pp. 53–74.

[33] L.K. Grover. A fast quantum mechanical algorithm for database search. Phys. Rev. Lett, 79, 1997..

[34] Gerard J. Milburn, An introduction to quantum computing, Rinton Press, Incorporated, 2007.

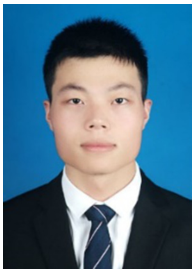
[35] Arvid J. Bessen. A Lower Bound for Quantum Phase Estimation. Phys. Rev. A 71 (4):–, 2005..

[36] S.A. Lobov, A.N. Mikhaylov, V.B. Kazantsev, Spatial properties of stdp in a self-learning spiking neural network enable controlling a mobile robot, Front. Neurosci., 14:–, 2020..

[37] Dennis Wackerly, William Mendenhall, Cengage Learning, Mathematical Statistics with Applications, Duxbury Press, 2008.

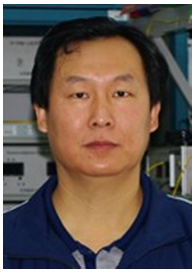
Yanhu Chen received the B.S. degree in Control Technology and Instrument from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2017. He is currently pursuing the Ph.D. degree in electronic science and technology at Beijing University of Posts and Telecommunications(BUPT), Beijing, China. His research interests include quantum computation and machine learning.

Cen Wang received the bachelor's degree in telecommunication engineering from Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, China, in 2014. He received the Ph.D. from Beijing University of Posts and Telecommunication in 2019. He has internship experience in KDDI Research Lab.INC, japan. His research interests include optical interconnection, AI assisted network resource scheduling, and SDN control.



Xiong Gao received the B.S. degree in communication engineering from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2016. He is currently pursuing the Ph.D. degree in electronic science and technology at Beijing University of Posts and Telecommunications (BUPT), Beijing, China. His research interests include routing algorithm in complex networks, data center optical networks and distributed cloud data center networks.

Hongxiang Guo received the B.E. and Ph.D. degrees in electrical engineering from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2000 and 2005, respectively. From 2005 to 2008, he was a Post-doctoral Researcher at KDDI R&D Laboratories, Inc., Saitama, Japan and was involved in research on intelligent all-optical networks and their control and management technologies. He is currently with BUPT. His current research interests include photonic networks, optical computing, and quantum computing algorithms.



Jian Wu received the B.S. degree in optoelectronics from the Beijing Institute of Technology, Beijing, China, in 1995, and the Ph.D. degree in physical electronics from Tsinghua University, Beijing, in 1999. From 1999 to 2001, he was a Postdoctoral Fellow at the Optical Communication Center, Beijing University of Posts and Telecommunications (BUPT), Beijing, where he was involved in the area of high-speed optical networks and all-optical signal processing. He is currently a Professor in the State Key Laboratory of Information Photonics and Optical Communications, BUPT. His research interests include optical packet/burst switching networks, network architecture and simulation, all-optical signal processing, optical time-domain multiplexing, high-speed optical transmission systems, nonlinear fiber optics, and high-speed opto-electronic devices.