Article

Quantum superposition inspired spiking neural network

Quantum superposition state encoding

$$|I(\theta_i)\rangle = \frac{1}{2^n} \sum_{i=0}^{2^{2n}-1} (cos(\theta_i)|x_i\rangle + sin(\theta_i)|\bar{x}_i\rangle) \otimes |i\rangle$$

Transfer to spikes

Original   Reverse θ

QS-SNN model

Two-compartment Neuron

Output

TCK synapse

Input

Soma V_i

Dendrite V_i^b Dendrite V_i^b

Yinqian Sun, Yi Zeng, Tielin Zhang

yi.zeng@ia.ac.cn

Highlights

Quantum and neuroscience research inspire new methods for artificial intelligence

QS-SNN integrates characteristics of quantum superposition state and brain spikes

QS-SNN can handle the background reverse visual inputs well

QS-SNN model is robust to reverse pixels noise and Gaussian noise on visual inputs

iScience

Article

Quantum superposition inspired spiking neural network

Yinqian Sun,[1,4,6] Yi Zeng,[1,2,3,4,5,6,7,*] and Tielin Zhang[1]

SUMMARY

Despite advances in artificial intelligence models, neural networks still cannot achieve human performance, partly due to differences in how information is encoded and processed compared with human brain. Information in an artificial neural network (ANN) is represented using a statistical method and processed as a fitting function, enabling handling of structural patterns in image, text, and speech processing. However, substantial changes to the statistical characteristics of the data, for example, reversing the background of an image, dramatically reduce the performance. Here, we propose a quantum superposition spiking neural network (QS-SNN) inspired by quantum mechanisms and phenomena in the brain, which can handle reversal of image background color. The QS-SNN incorporates quantum theory with brain-inspired spiking neural network models from a computational perspective, resulting in more robust performance compared with traditional ANN models, especially when processing noisy inputs. The results presented here will inform future efforts to develop brain-inspired artificial intelligence.

INTRODUCTION

Many machine learning methods using quantum algorithms have been developed to improve parallel computation. Quantum computers have also been shown to be more powerful than classical computers when running certain specialized algorithms, including Shor's quantum factoring algorithm (Shor, 1999), Grover's database search algorithm (Grover, 1996), and other quantum-inspired computational algorithms (Manju and Nigam, 2014).

Quantum computation can also be used to find eigenvalues and eigenvectors of large matrices. For example, the traditional principal components analysis (PCA) algorithm calculates eigenvalues by decomposition of the covariance matrix; however, the computational resource cost increases exponentially with increasing matrix dimensions. For an unknown low-rank density matrix, quantum-enhanced PCA can reveal the quantum eigenvectors associated with the large eigenvalues; this approach is exponentially faster than the traditional method (Lloyd et al., 2014).

K-means is a classic machine learning algorithm that classifies unlabeled data sets into k distinct clusters. A quantum-inspired genetic algorithm for K-means has been proposed, in which a qubit-based representation is employed for exploration and exploitation in discrete "0" and "1" hyperspace. This algorithm was shown to obtain the optimal number of clusters and the optimal cluster centroids (Xiao et al., 2010). Quantum algorithms have also been used to speed up the solving of subroutine problems and matrix inversion problems (Harrow et al., 2009); for example, Grover's algorithm (Grover, 1996) provides quadratic speedup of a search of unstructured databases.

The quantum perceptron and quantum neuron computational models combine quantum theory with neural networks (Schuld et al., 2014). Compared with the classical perceptron model, the quantum perceptron requires fewer resources and benefits from the advantages of parallel quantum computing (Schuld et al., 2015; Torrontegui and García-Ripoll, 2019). The quantum neuron model (Cao et al., 2017; Mangini et al., 2020) can also be used to realize classical neurons with sigmoid or step function activation by encoding inputs in quantum superposition, thereby processing the whole data set at once. Deep quantum neural networks (Beer et al., 2020) raise the prospect of deploying deep learning algorithms on quantum computers.

[1]Research Center for Brain-Inspired Intelligence, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

[2]Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai 200031, China

[3]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

[4]School of Future Technology, University of Chinese Academy of Sciences, Beijing 100190, China

[5]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100190, China

[6]These authors contributed equally

[7]Lead contact

*Correspondence:
yi.zeng@ia.ac.cn

https://doi.org/10.1016/j.isci.2021.102880

Spiking neural networks (SNNs) represent the third generation of neural network models (Maass, 1997) and are biologically plausible from neuron, synapse, network, and learning principles perspectives. Unlike the perceptron model, neurons in an SNN accept signals from pre-synaptic neurons, integrating the post-synaptic potential and firing a spike when the somatic voltage exceeds a threshold. After spiking, the neuron voltage is reset in preparation for the next integrate-and-fire process. SNNs are powerful tools for representation and processing of spatial-temporal information. Many types of SNNs have been proposed for different purposes. Examples include visual pathway-inspired classification models (Zenke et al., 2015; Zeng et al., 2017), basal ganglia-based decision-making models (Héricé et al., 2016; Cox and Witten, 2019; Zhao et al., 2017), and other shallow SNN (Khalil et al., 2017; Shrestha and Orchard, 2018). Different SNN may include different types of biologically plausible neurons, e.g., the leaky integrate-and-fire (LIF) model (Gerstner and Kistler, 2002), Hodgkin–Huxley model, Izhikevich model (Izhikevich, 2003), and spike response model (Gerstner, 2001). In addition, many different types of synaptic plasticity principles have been used for learning, including spike-timing-dependent plasticity (Dan and Poo, 2004; Frémaux and Gerstner, 2016), Hebbian learning (Song et al., 2000), and reward-based tuning plasticity (Héricé et al., 2016).

Quantum superposition SNN has theoretic basis in both biology (Vaziri and Plenio, 2010) and computational models (Kristensen et al., 2019). From one perspective, spiking neuron models, such as the LIF and Izhikevich models, can be reformed by quantum algorithms in order to accelerate their processing using a quantum computer. On the other hand, quantum effects such as entanglement and superposition are regarded as special information-interactive methods and can be used to modify the classical SNN framework to generate similar behavior to that of particles in the quantum domain. In this work, we follow the latter approach. More specifically, we use a quantum superposition mechanism to encode complementary information simultaneously and further transfer it to spike trains, which are suitable for SNN processing. In our proposed quantum superposition SNN (QS-SNN) model, quantum state representation is integrated with spatiotemporal spike trains in SNN. This characteristic is conducive to good model performance not only on standard image classification tasks but also when handling color-inverted images. QS-SNN encodes the original image and the color-inverted image in the format of quantum superposition; the changing background context demonstrated by the spiking phase and spiking rate contains the image pixels' identity information.

We combine quantum superposition information encoding with SNN for three reasons. First, the possible influence of quantum effects on biological processes and the related quantum brain hypothesis have been theoretically investigated (Vaziri and Plenio, 2010; Fisher, 2015; Weingarten et al., 2016). Second, quantum superposition states are represented by vectors in complex Hilbert space, in contrast to traditional artificial neural network (ANN), which operate in real space only; this is more representative of brain spikes, as the spiking rate and spiking phase spatiotemporal property also have complex number representation. In essence, SNN are more appropriate for quantum-inspired superposition information encoding. Third, current quantum machine learning methods, especially those used for quantum image processing, focus on encoding a classical image in the quantum state, with the image processing methods accelerated by quantum computing (Iyengar et al., 2020). There has been less exploration of the possibility of using a quantum superposition state coding mechanism for different pattern information-processing frameworks. More importantly, owing to the use of statistical methods and fitting functions, current ANN show a huge performance drop when required to recognize a background-inverted image. This inspired us to develop a new information representation method unlike that used in traditional models. The integration of characteristics from SNN and quantum theory is intended to achieve a better representation of multi-states and potentially enable easier solving of tasks that are challenging for traditional ANN and SNN models.

The subsequent sections describe how complementary superposition information is generated and transferred to spatiotemporal spike trains. A two-compartment SNN is used to process the spikes. The proposed model, combining complementary superposition information encoding with the SNN spatiotemporal property, can successfully recognize a background color-inverted image, which is hard for traditional ANN models.

Complementary superposition information encoding

Quantum image processing

Quantum image processing combines image processing methods with quantum information theory. There are many approaches to internal representation of an image in a quantum computer, including flexible

representation of quantum images (FRQIs), NEQR, GQIR, MCQI, and QBIP (Iyengar et al., 2020; Mastriani, 2020), which transfer the image to appropriate quantum states for the next step of quantum computing. Our approach is inspired by the FRQI method (Le et al., 2011), as shown in Equations (1) and (2):

$$\left| I(\theta) \right\rangle = \frac{1}{2^n} \sum_{i=0}^{2^{2n}-1} \left(sin(\theta_i)|0\rangle + cos(\theta_i)|1\rangle \right)|i\rangle, \tag{Equation 1}$$

$$\theta_i \in \left[0, \frac{\pi}{2}\right], i = 1, 2, 3, ..., 2^{2n} - 1, \tag{Equation 2}$$

where $\left| I(\theta) \right\rangle$ is the quantum image, qubit $|i\rangle$ represents the position of a pixel in the image, and $\theta = (\theta_0, \theta_1, ..., \theta_{2^{2n}-1})$ encodes the color information of the pixels. FRQI satisfies the quantum state constraint in Equation (3):

$$\| \, |I(\theta)\rangle \| = \frac{1}{2^n} \sqrt{\sum_{i=0}^{2^{2n}-1} \left(cos^2\theta_i + sin^2\theta_i \right)} = 1. \tag{Equation 3}$$

Complementary superposition spikes

We propose a complementary superposition information encoding method and establish a linkage between quantum image formation and spatiotemporal spike trains. The complement code is widely used in computer science to turn subtraction into addition. We encode the original information and complementary information into a superposition state; one example is shown in Equation (4), with the rightmost sign bit removed and taking the complement:

$$\left| I(\theta_i) \right\rangle = cos(\theta_i)|[0000001]_b\rangle + sin(\theta_i)|[1111110]_b\rangle. \tag{Equation 4}$$

Equation (4) is an illustration of how complementary superposition information encoding works, with no factual significance. In this work, we focus on quantum image superposition encoding, as in Equation (5). However, it should be noted that any form of information that has a complement format, not just an image, can be encoded as a superposition state. Images in complementary quantum superposition states are further transferred to spike trains, as depicted in Figure 1. An image in its complementary state has an inverted background.

$$\left| I(\theta) \right\rangle = \frac{1}{2^n} \sum_{i=0}^{2^{2n}-1} \left(cos(\theta_i)|x_i\rangle + sin(\theta_i)\left|\overline{x}_i\right\rangle \right) \otimes |i\rangle, \tag{Equation 5}$$

$$\theta_i \in \left[0, \frac{\pi}{2}\right], i = 1, 2, 3, ..., 2^{2n} - 1. \tag{Equation 6}$$

The complementary quantum superposition encoding is shown in Equations (5) and (6), where the $|i\rangle$ represent pixel positions. Unlike FRQI, which uses qubits only for color encoding, here we use complementary qubits for encoding both original image pixels x_i and the color-inverted image $\overline{x}_i$ with $\overline{x}_i = 1 - x_i$, supposing the pixel x_i domain ranges from 0 to 1.0. The parameter θ_i represents the degree of quantum image $|I\rangle$, mixing the original state $|x\rangle$ and reverse state $|\overline{x}\rangle$.

We designed quantum circuit for the generation of quantum superposition image $|I(\theta_i)\rangle$ as shown in the Figure 2A, which is also discussed in (Le et al., 2011; Dendukuri and Luu, 2018). The quantum state $|x_i\rangle$ is processed by Hadamard transform H and controlled NOT gate to form the complementary state $|\beta_{ix_i}\rangle$ with:

$$\left| \beta_{ix_i} \right\rangle = \frac{|0, x_i\rangle + (-1)^i |1, \overline{x}_i\rangle}{\sqrt{2}}, \tag{Equation 7}$$

Then rotation matrices R_i is used to encode phase information as

$$R_i = \begin{bmatrix} cos\dfrac{\theta_i}{2} & -sin\dfrac{\theta_i}{2} \\[2mm] sin\dfrac{\theta_i}{2} & cos\dfrac{\theta_i}{2} \end{bmatrix} \tag{Equation 8}$$

Finally, the superposition state $|I(\theta_i)\rangle$ is measured and two states are retrieved with probability P_i and Q_i.
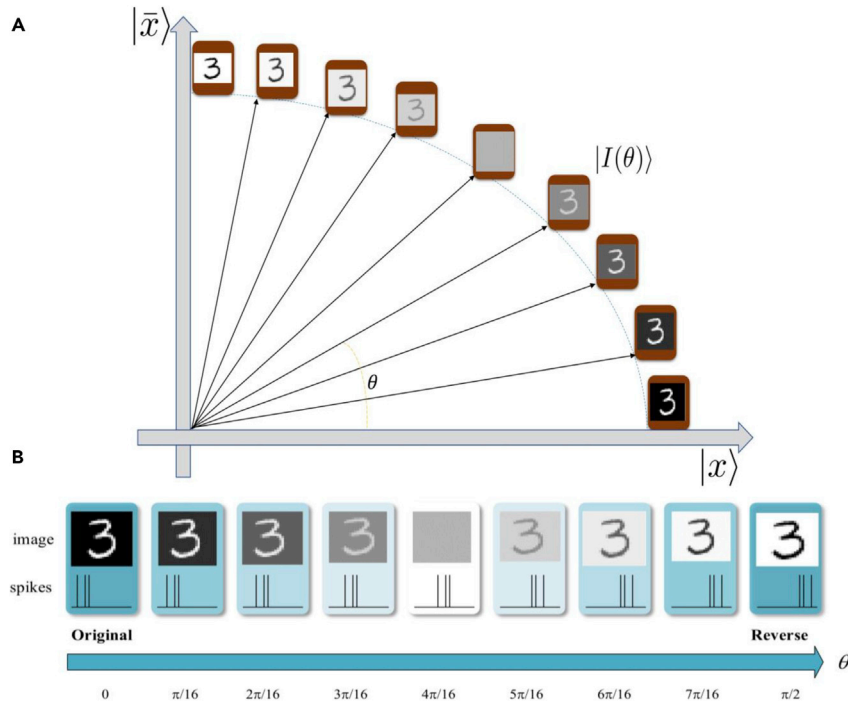
Figure 1. Quantum complementary superposition information encoding

(A) The horizontal axis and longitudinal axis represent $|x\rangle$ and $|\bar{x}\rangle$, respectively. The parameter θ in Equation (5) measures the degree to which the image background is inverted, from $\theta = 0$ (the original image) to the complementary state $\theta = \frac{\pi}{2}$ (totally inverted background).

(B) The top pictures show images inverted to different degrees, and the spikes to which they are encoded. The bottom axis corresponds to the value of θ. It should be noted that the pictures are intuitive demonstration instead of exact display.

The complex information in quantum encoding is similar to signal processing in SNN. Neuron spikes can encode spatiotemporal information with specific spiking rates and spiking times, which can be used to represent quantum information.

Neuron spikes have the attribute of spatiotemporal dimension, which are identical in shape but differ significantly in frequency and phase, seeing Izhikevich neuron model (Izhikevich, 2003) in Figure S1, and are well-suited to the implementation of the vector form quantum image in Equation (5). We use spike trains with firing rate r_i and firing phase ϕ_i to represent quantum image state $|I(\theta_i)\rangle$. As shown in Figure 2A, the spike trains containing information of $|I(\theta_i)\rangle$ can be generated using Equation(9) and (10).

$$r_i = \frac{\||I(\theta)\rangle\| - sin(\phi_i)}{cos(\phi_i) - sin(\phi_i)}, \qquad \text{(Equation 9)}$$

$$\phi_i = \mathcal{F}\left\{ arctan\left(\frac{P_j}{Q_j}\right) \middle| j = 1, 2, ..., N. \right\}, \qquad \text{(Equation 10)}$$

Notation $\mathcal{F}\{X_i\}$ is set operation, and is specific in different tasks. The superposition state encoding $|I(\theta_i)\rangle$ is transferred to spike trains $S_i(t; \phi_i)$, which is generated from a Poisson spikes $S_i(t)$ with spike rate r_i and extended phase ϕ_i shown as Equation (11). Here, T is the time interval of neuron processing spikes received from pre-synaptic neurons, and T_{sp} is the spiking time window in this period, as shown in Figure 2B:

$$S_i(t; \phi_i) = S_i(t - t_0), \qquad \text{(Equation 11)}$$

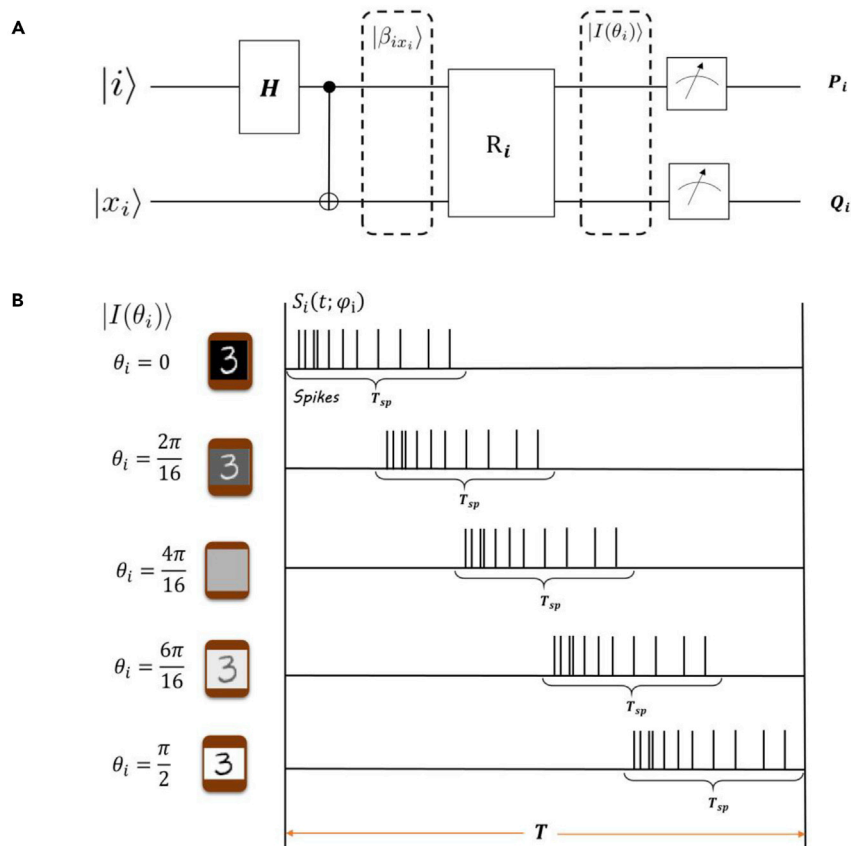$$t_0 = \frac{\phi_i}{\pi/2} * (T - T_{sp}). \qquad \text{(Equation 12)}$$

Figure 2. Quantum superposition spike trains

(A) The circuit to generate quantum image. Only one image pixel state is depicted for perspicuity.

(B) A schematic diagram shows the transformation of quantum superposition states to spike trains $S_i(t; \phi_i)$. With parameter θ_i increasing, spike trains are shifted in time dimension. T is a simulation period in which spikes emerge within the T_{sp} time window. Also, the relation of parameter θ_i and spiking phase ϕ_i is intuitive example and not exact correspondence.

Two-compartment SNN

Synapses with time-differential convolution

Synapses play an important part in the conversion of information from spikes in pre-synaptic neurons to membrane potential (or current) in post-synaptic neurons. In this work, the time-differential kernel (TCK) convolution synapse is used, as shown in Equations (13) and (14) and Figure S2. The spikes, S_i, from pre-synaptic neurons are convoluted with a kernel and then integrated with the dendrite membrane potential $V_b(t)$. This process can be considered as a stimulus-response convolution with the form of a Dirac function (Urbanczik and Senn, 2014):

$$\begin{cases} \kappa(t) = \zeta(t) - \zeta(-t) \\ \zeta(t) = \Theta(t)\left(e^{-\frac{t}{\tau}}\right) \end{cases}, \qquad \text{(Equation 13)}$$

$$V_j^b(t) = \sum_i w_{i,j} \left\| \int_{-T}^{+T} \kappa(\tau) S_i(\tau) \, d\tau \right\|. \qquad \text{(Equation 14)}$$

Two-compartment neurons

Both the hidden layer and the output layer contain biologically plausible two-compartment neurons, which dynamically update the somatic membrane potential $V_i(t)$ with the dendrite membrane potential $V_i^b(t)$, as shown in Figure S2.

In the compartment neuron model, $V_i^h(t)$ is the membrane potential of neuron i in the hidden layer, which is updated with Equation (15); g_B, g_L, and τ_L are hyperparameters that represent synapse conductance, leaky conductance, and the integrated time constant, respectively; $V_j^{PSP}(t)$ is the synaptic input from neuron j; $V_i^{h,b}(t)$ is the dendrite potential with adjustable threshold b_i^h in the hidden layer; and w_{ij}^h is the synaptic weight between the input and hidden layers:

$$\begin{cases} \tau_L \dfrac{dV_i^h(t)}{dt} = -V_i^h(t) + \dfrac{g_B}{g_L}\left(V_i^{h,b}(t) - V_i^h(t)\right) \\[2mm] V_i^{h,b}(t) = \sum_j w_{ij}^h V_j^{PSP}(t) + b_i^h \\[2mm] V_j^{PSP}(t) = \left\| \int_{-T}^{+T} \kappa(\tau) S_j(\tau)\, d\tau \right\|. \end{cases} \qquad \text{(Equation 15)}$$

The somatic neuron model in the output layer contains 10 neurons corresponding to 10 classes of the MNIST data set. As shown in Equation (16), the hidden layer neurons deliver signals to the output layer with integrated spike rate r_i, which is differentiable; hence, it can be tuned with back-propagation. Here, $V_i^o(t)$ is the membrane potential in the output layer, $V_i^{o,b}(t)$ is the dendrite potential, and r_{max} is the hyperparameter for rescaling of fire-rate signals:

$$\begin{cases} \tau_L \dfrac{dV_i^o(t)}{dt} = -V_i^o(t) + \dfrac{g_B}{g_L}\left(V_i^{o,b} - V_i^o(t)\right) \\[2mm] V_i^{o,b}(t) = \sum_j w_{ij}^o r_j(t) + b_i^o \\[2mm] r_j = r_{max}\,\sigma\left(V_j^h\right) \\[2mm] \sigma(x) = 1/\left(1 - exp(-x)\right). \end{cases} \qquad \text{(Equation 16)}$$

The shallow three-layered architecture is shown in Figure 3. The input layer receives quantum spikes with encoding of complementary qubits. The hidden layer of QS-SNN consists of a two-compartment model with time-differential convolution synapses. Neurons in the output layer are integrated spike rate neurons, which receive an integrated fire rate from pre-synaptic neurons, as well as the teaching signal V_i as information about class labels.

Computational experiments

We examined the performance of the QS-SNN framework on a classification task using background color-inverted images from the MNIST (LeCun et al., 2010) and Fashion-MNIST (Xiao et al., 2017) data sets. QS-SNN encodes the original image and its color-inverted mirror as complementary superposition states and transfers them to spiking trains as an input signal to the two-compartment SNN. The dendrite prediction and proximal gradient methods used to train this model can be found in STAR Methods.

For comparison, we also tested several deep learning models on the color-inverted data sets, including a fully connected ANN, a 10-layers CNN (Lecun et al., 1998), VGG (Simonyan and Zisserman, 2015), ResNet (He et al., 2016), and DeseNet (Huang et al., 2017). All models are trained with original image x_i and then tested on the background reverse image $I(\theta_i)$. The only difference is that, for QS-SNN, quantum superposition state image $|I(\theta_i)\rangle$ is transferred to spike trains which is compatible with spiking neural networks. In other words, our essential idea is that the spatiotemporal property of neuron spikes enables the brain to transform spatially variant information into time-differential information.

In this work, we formulated the spike trains transformation as the quantum superposition shown in Equation (5). And we have also demonstrated the numerical calculation of $I(\theta_i)$ which is used for traditional ANN and CNN model testing in STAR Methods.

In addition, it is worth noting that the superposition state image $|I(\theta_i)\rangle$ is constructed from original information $|x_i\rangle$ and reverse image $|\bar{x}_i\rangle$. The original image and its complementary reverse information are maintained in the superposition state encoding $|I(\theta_i)\rangle$ at the same time. Because SNN is not processing pixel
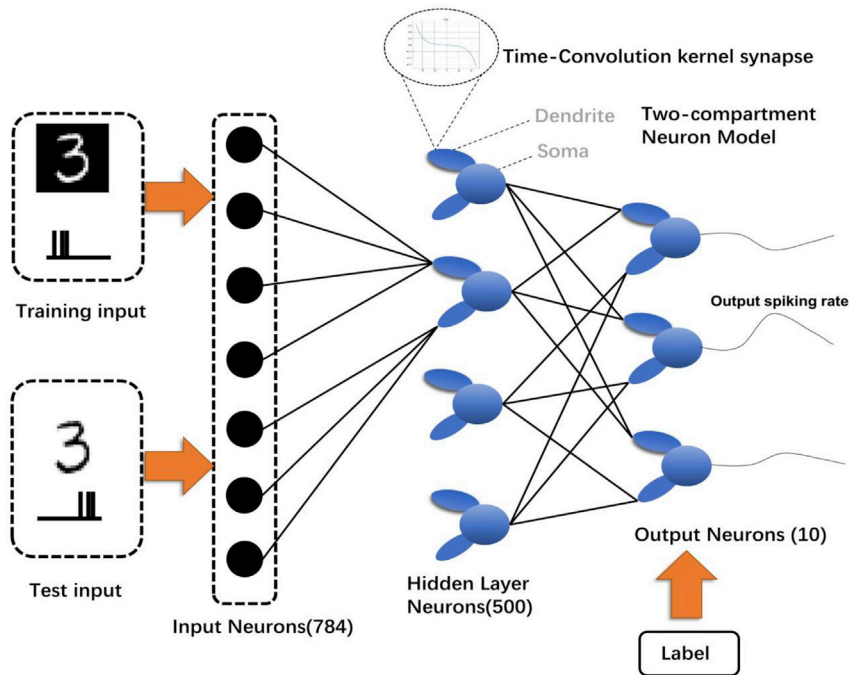
Figure 3. Quantum superposition SNN, three-layer architecture of QS-SNN with TCK synapses and two-compartment neurons

Images are transferred to spikes as network inputs. The hidden layer is composed of 500 two-compartment neurons with dendrite and soma. The output layer contains 10 two-compartment neurons corresponding to 10 classes. In the training period, only original images are fed to the network, whereas in the test period, the trained network is tested with inverted background images. Neurons with maximum spiking rates at the output layer are taken as the network prediction and output.

value directly, we transformed $|I(\theta_i)\rangle$ to spike trains, which can be regarded as different expression of superposition state encoding image in spatiotemporal dimension.

Standard and color-inverted images

The standard MNIST data set contains images of 10 classes of handwritten digits from 0 to 9; images are 28x28 pixels in size, with 60,000 and 10,000 training and test samples, respectively. Fashion-MNIST has the same image size and the same training and testing split as MNIST but contains grayscale images of different types of clothes and shoes.

The original MNIST and Fashion-MNIST images and their color-inverted versions, with different degrees of inversion as measured by parameter θ, are depicted in Figures 4A and 4B, respectively. To be specific, the spiking phase estimating operation $\mathcal{F}\{X_i\}$ in Equation (10) is set as piecewise selection function as

$$\phi_i = \begin{cases} arctan\left(\dfrac{P_j}{Q_j}\right), & j = i, \\ 0, & j \neq i. \end{cases} \qquad \text{(Equation 17)}$$

Robustness to reverse pixel noise and Gaussian noise

Besides the effects of changing the whole background, we were interested in the capability of QS-SNN to handle other types of destabilization of images. For this purpose, we added reverse spike pixels and Gaussian noise to the MNIST and Fashion-MNIST images, and further tested the performance of QS-SNN in comparison with that of ANN and CNN. Reverse spike noise is created by randomly flipping image pixels to their reverse color and can be described as $Reverse(image[i]) = 1 - image[i]$. The position i of the pixel to be flipped is randomly chosen, as shown in Figures 5A and 5B. The noisy images were encoded and processed in the same way as described in Algorithm 1. However, in the color-inverted experiment, all pixels of reverse degree θ_i were the same, resulting in the same change being applied to the whole image. By contrast, in the reverse pixel noise experiment, only a proportion of randomly chosen image pixels were
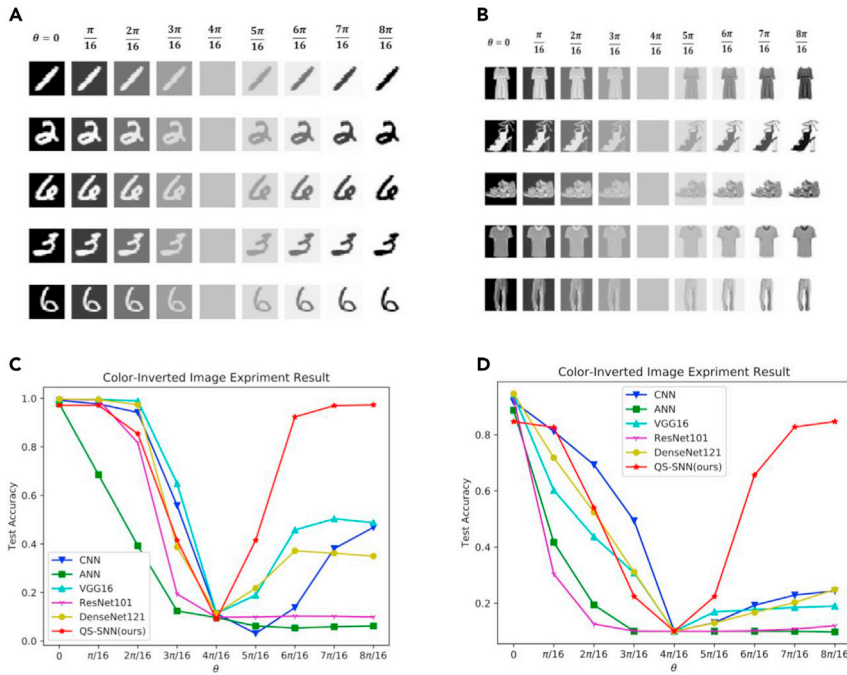
Figure 4. Classification of color-inverted images

(A) MNIST background color-inverted images. Parameter θ takes values from 0 to $\frac{\pi}{2}$, denoting the degree of color inversion.

(B) Fashion-MNIST background color-inverted images.

(C) Background color-inverted MNIST classification results. QS-SNN initially showed performance degeneration similar to that of the fully connected ANN and CNN, with θ values from 0 to $\frac{4\pi}{16}$. However, as the background-inversion degree further increased, QS-SNN gradually recovered its accuracy, whereas the other networks did not. When the background was totally inverted ($\theta = \frac{8\pi}{16}$), QS-SNN showed almost the same performance as when classifying original images, whereas the second-best network (VGG16) retained only half its original accuracy.

(D) Background color-inverted Fashion-MNIST results. Similar results as in the MNIST experiment were achieved, with QS-SNN showing an even greater advantage (right-hand side).

changed; thus, every image pixel had a specific θ_i parameter. These image pixels are transferred to spike trains with heterogeneous phase ϕ_i. Specially in reverse pixel experiment we took the mean operation for $\mathcal{F}\{\cdot\}$ as the estimation phase:

$$\phi_i = \frac{1}{N} \sum_{j=1}^{N} arctan\left(\frac{P_j}{Q_j}\right) \qquad \text{(Equation 18)}$$

Additive white Gaussian noise (AWGN) is commonly used to test system robustness. We also examined the performance of the proposed QS-SNN on AWGN MNIST and Fashion-MNIST images, as shown in Figures 6A and 6B.

In contrast to color-inverted noise, AWGN results in uncorrelated disturbances on the original image. We were interested in the robustness of our proposed method when faced with this challenging condition. The procedure used to process AWGN images was the same as that used in the reverse pixel noise experiment, except that the whole image phase was estimated using half the median operation:

$$\phi_i = M\left\{arctan\left(\frac{P_j}{Q_j}\right)\middle| j = 1, ..., N\right\} \qquad \text{(Equation 19)}$$

RESULTS

Standard and color-inverted data sets experiment

We constructed a three-layer QS-SNN with 500 hidden layer neurons and 10 output layer neurons. The structure of the experimental fully connected ANN was set to be the same for comparison. A simple
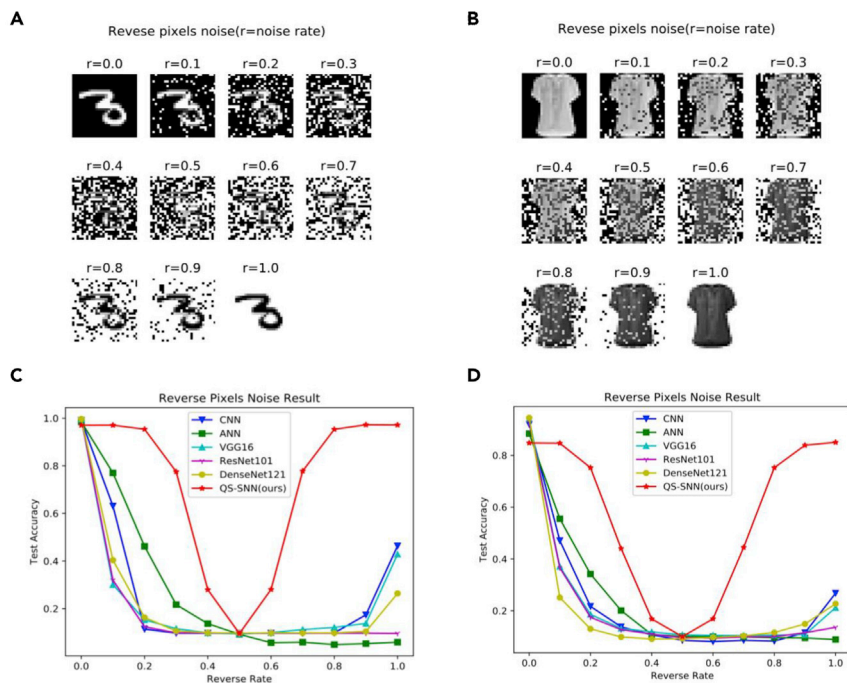
Figure 5. Reverse spike noise results

(A) Reverse spike noise MNIST. The possibility of pixel inversion is controlled by parameter *r*. When *r* = 0, no noise is added, i.e., the image is original data. When *r* = 1.0, all pixels are flipped.

(B) Reverse spike noise Fashion-MNIST.

(C) Classification of MNIST images with reverse noise. QS-SNN performed better compared with the inverted background experiment.

(D) Classification of Fashion-MNIST images with reverse noise.

CNN structure with three convolution layers and two pooling operations was used to determine the ability of different feature extraction methods to deal with inverted background images. We also tested VGG16, ResNet101, and DenseNet121 to investigate whether deeper structures could classify color-inverted images correctly. ANN, CNN, and QS-SNN were trained for 20 epochs with the Adam optimization method, and the learning rate was set to 0.001. VGG16, ResNet101, and DenseNet121 were trained for 400 epochs using stochastic gradient descent with learning rate 0.1, momentum 0.9, weight decay 5e-4, and learning rate decay 0.8 every 10 epochs. In the training phase, only the original image ($\theta = 0$) was used; the testing phase used different color-inverted images (θ ranging from 0 to $\frac{\pi}{2}$). All results were obtained from the final epoch test step.

The results showed that the traditional fully connected ANN and convolution models struggled to handle huge changes in image properties such as background reversal, even when the spatial features of the image remained the same. Our proposed method showed much better performance than these traditional models (see Figures 4C and 4D and Tables S2 and S3 for details). Significant performance degradation occurred when processing color-inverted images with ANN and CNN, and even deeper networks such as VGG16, ResNet101, and Densenet121 experienced problems with color-inverted image classification. By contrast, QS-SNN, although affected by a similar performance drop when images were made blurry (θ from 0 to $\frac{4\pi}{16}$), regained its ability when the images' backgrounds were inverted, and the clarity was improved (θ from $\frac{4\pi}{16}$ to $\frac{8\pi}{16}$). When the image color was fully inverted ($\theta = \frac{8\pi}{16}$), QS-SNN retained the same accuracy as when classifying the original data ($\theta = 0$) and correctly recognized color-inverted MNIST and Fashion-MNIST images.

Robustness to noise experiments

Compared with other state-of-the-art models, the performance of QS-SNN was closer to human vision capacity. As more flipped-pixel noise was added to the images ($r = 0$ to 0.5), they became increasingly difficult

<div style="border:1px solid #000; padding:10px;">

Algorithm 1. The learning procedure of QS-SNN

1. Initialize weights $W_{j,i}$ with random uniform distribution, membrane potential states V_i, and other related hyperparameters as in Table S1.

2. Start training procedure with only original images in training dataset, $\theta_i = 0$:

   2.1 Load training samples.

   2.2 Construct quantum superposition state representations of images.

   2.3 Input neuron spikes as Poisson process with spiking rate and phase time according to quantum superposition image.

   2.4 Process time-differential convolution to obtain dynamical updating of membrane potential of post-synaptic neurons.

   2.5 Update multi-layer membrane potential.

   2.6 Train the QS-SNN with dendrite prediction and proximal gradient.

   2.7 Select neurons in output layer with maximum spiking rate as the output class.

3. Start test procedure using color-inverse images with different degree of color inversion from the test dataset, $\theta_i = 0, \frac{\pi}{16}, \ldots, \frac{8\pi}{16}$.

   3.1 Load the test samples and transfer to spike trains as in steps 2.2 and 2.3.

   3.2 Test the performance of the trained QS-SNN on color-inverse images.

   3.3 Output the test performance.

</div>

to recognize, as indicated by the left side of the "U"-curve for QS-SNN in Figures 5C and 5D. However, as more noise was added to the pixels, the image features became clear again. When $r = 1.0$, with all pixels reversed, there was no conflict with the features of the original image when $r = 0$. QS-SNN can exploit these conditions owing to its image superposition encoding method (Equation 5), which is similar to the human vision system. As shown in Figures 5C and 5D and Tables S4 and S5, randomly inverting image pixels caused substantial performance degradation of ANN and CNN, as well as of the deep networks. On the contrary, the red "U"-shaped curve for QS-SNN indicated that it recovered its accuracy as the image's features became clear but the background was inverted ($r = 0.6$ to 1.0).

Gaussian noise influenced all networks significantly, with all methods showing a performance drop as noise (standard deviation; *std*) increased, as shown in Figures 6C and 6D and Tables S6 and S7. QS-SNN behaved more stably on the AWGN image processing task, with accuracies of 90.2% and 82.3% on the MNIST and Fashion-MNIST data sets, respectively, for $std = 0.4$; by contrast, the other methods achieved no more than 60% and 50%, respectively. Images with $std = 0.4$ are not very difficult for human vision to distinguish. Thus, by combining a brain-inspired spiking network with a quantum mechanism, we obtain a more robust approach to images with noise disturbance, similar to the performance of human vision.

DISCUSSION

This work aimed to integrate quantum theory with a biologically plausible SNN. Quantum image encoding and quantum superposition states were used for information representation, followed by processing with a spatial-temporal SNN. A time-convolution synapse was built to obtain neuron process phase information, and dendrite prediction with a proximal gradient method was used to train the QS-SNN. The proposed QS-SNN showed good performance on color-inverted image recognition tasks that were very challenging to other models. Compared with traditional ANN models, QS-SNN showed better generalization ability.

It is worth noting that the quantum brain hypothesis is quite controversial. Nevertheless, this paper does not aim to provide direct persuasive evidence for the quantum brain hypothesis but to explore novel information-processing methods inspired by quantum information theory and brain spiking signal transmission.
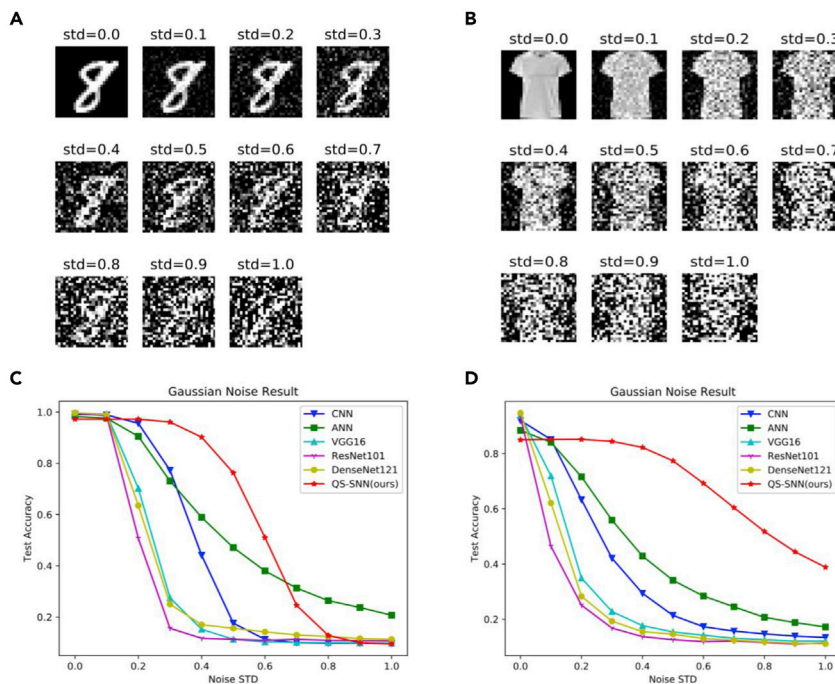
Figure 6. Gaussian noise image classification

(A) Additive white Gaussian noise on MNIST. The mean of Gaussian random noise was set to zero, and different *std* values were used.

(B) Additive white Gaussian noise on Fashion-MNIST.

(C) Classification of MNIST images with Gaussian noise. Compared with other networks, QS-SNN showed much slower degeneration.

(D) Classification of Fashion-MNIST images with Gaussian noise. QS-SNN performed even better compared with its results on MNIST.

Limitations of the study

Our model was inspired by quantum image processing methods, in particular, quantum image super-position state presentation. The model and corresponding experiments were run on a classical computer and did not use any quantum hardware; thus, our work could not benefit from quantum computing. Artificial neurons can be reformed to run on quantum computers (Schuld et al., 2014; Cao et al., 2017; Mangini et al., 2020). Efforts to build a quantum spiking neuron are still at a preliminary stage (Kristensen et al., 2019). Simulating spiking neuronal networks on a classical computer is hindered by heavy resource consumption and slow processing.

Future work includes modifying both the quantum superposition encoding strategy and the SNN architecture to suit quantum computing better. In computational neuroscience research, neuronal spikes are typically generated with the Poisson process, which samples data from the binomial distribution. Quantum bits, also named qubits, are fundamental components in a quantum computer. A qubit can take the value of "0" or "1" with a certain probability, which is very similar to neuronal spikes. Thus, a set of qubits can encode all possible states of a spike train, as well as the quantum superposition images, in the quantum computer. Although it requires much effort to reconstruct spiking neural models suited for quantum computing, it is significant in neurology and artificial intelligence research to explore more quantum-inspired mechanisms to explain brain functions that traditional theories fail to.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE

- RESOURCE AVAILABILITY
  - ○ Lead contact
  - ○ Materials availability
  - ○ Data and code availability
- METHOD DETAILS
  - ○ Generating background inverse image
  - ○ Learning procedure of the QS-SNN algorithm

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2021.102880.

AUTHOR CONTRIBUTIONS

Y.S. wrote the code, performed the experiments, analyzed the data, and wrote the manuscript. Y.Z. proposed and supervised the project and contributed to writing the manuscript. T.Z. participated in helpful discussions and contributed to writing the manuscript.

DECLARATION OF INTERESTS

The authors declare that they have no competing interests.

REFERENCES

Beer, K., Bondarenko, D., Farrelly, T., Osborne, T.J., Salzmann, R., Scheiermann, D., and Wolf, R. (2020). Training deep quantum neural networks. Nat. Commun. *11*, 1–6.

Cao, Y., Guerreschi, G.G., and Aspuru-Guzik, A. (2017). Quantum neuron: an elementary building block for machine learning on quantum computers. arXiv, arXiv:1711.11240.

Cox, J., and Witten, I.B. (2019). Striatal circuits for reward learning and decision-making. Nat. Rev. Neurosci. *20*, 482–494.

Dan, Y., and Poo, M.M. (2004). Spike timing-dependent plasticity of neural circuits. Neuron *44*, 23–30.

Dendukuri, A., and Luu, K. (2018). Image processing in quantum computers. arXiv, arXiv:1812.11042.

Fisher, M.P. (2015). Quantum cognition: the possibility of processing with nuclear spins in the brain. Ann. Phys. *362*, 593–602.

Frémaux, N., and Gerstner, W. (2016). Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules. Front. Neural Circuits 9, 85.

Gerstner, W. (2001). A framework for spiking neuron models: the spike response modelF.

Moss and S. Gielen, eds. (2001)., *4* (Elsevier), pp. 469–516.

Gerstner, W., and Kistler, W.M. (2002). Spiking Neuron Models: Single Neurons, Populations, Plasticity (Cambridge University Press).

Grover, L.K. (1996). A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing (ACM), pp. 212–219.

Harrow, A.W., Hassidim, A., and Lloyd, S. (2009). Quantum algorithm for linear systems of equations. Phys. Rev. Lett. *103*, 150502.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (IEEE), pp. 770–778.

Héricé, C., Khalil, R., Moftah, M., Boraud, T., Guthrie, M., and Garenne, A. (2016). Decision making under uncertainty in a spiking neural network model of the basal ganglia. J. Integr. Neurosci. *15*, 515–538.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K.Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (IEEE), pp. 4700–4708.

Iyengar, S.S., Kumar, L.K., and Mastriani, M. (2020). Analysis of five techniques for the internal representation of a digital image inside a quantum processor. arXiv, arXiv:2008.01081.

Izhikevich, E.M. (2003). Simple model of spiking neurons. IEEE Trans. Neural Netw. *14*, 1569–1572.

Khalil, R., Moftah, M.Z., and Moustafa, A.A. (2017). The effects of dynamical synapses on firing rate activity: a spiking neural network model. Eur. J. Neurosci. *46*, 2445–2470.

Kristensen, L.B., Degroote, M., Wittek, P., Aspuru-Guzik, A., and Zinner, N.T. (2019). An artificial spiking quantum neuron. arXiv, arXiv:1907.06269.

Le, P.Q., Dong, F., and Hirota, K. (2011). A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. Quan. Inf. Process. *10*, 63–84.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. Proc. IEEE *86*, 2278–2324. https://doi.org/10.1109/5.726791.

LeCun, Y., Cortes, C., and Burges, C. (2010). MNIST Handwritten Digit Database (ATT Labs). http://yann.lecun.com/exdb/mnist.

Lloyd, S., Mohseni, M., and Rebentrost, P. (2014). Quantum principal component analysis. Nat. Phys. *10*, 631.

Maass, W. (1997). Networks of spiking neurons: the third generation of neural network models. Neural Netw. *10*, 1659–1671.

Mangini, S., Tacchino, F., Gerace, D., Macchiavello, C., and Bajoni, D. (2020). Quantum computing model of an artificial neuron with continuously valued input data. Mach. Learn. Sci. Technol. *1*, 045008.

Manju, A., and Nigam, M.J. (2014). Applications of quantum inspired computational intelligence: a survey. Artif. Intell. Rev. *42*, 79–156.

Mastriani, M. (2020). Quantum image processing: the truth, the whole truth, and nothing but the truth about its problems on internal image representation and outcomes recovering. arXiv, arXiv:2002.04394.

Schuld, M., Sinayskiy, I., and Petruccione, F. (2014). The quest for a quantum neural network. Quan. Inf. Process. *13*, 2567–2586.

Schuld, M., Sinayskiy, I., and Petruccione, F. (2015). Simulating a perceptron on a quantum computer. Phys. Lett. A *379*, 660–663.

Shor, P.W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev. *41*, 303–332.

Shrestha, S.B., and Orchard, G. (2018). SLAYER: spike layer error reassignment in time. In Advances in Neural Information Processing Systems, S. Bengio and H. Wallach, et al., eds. (Curran Associates), pp. 1412–1421.

Simonyan, K., and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. arXiv *1409.1566*, 1–14.

Song, S., Miller, K.D., and Abbott, L.F. (2000). Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. Nat. Neurosci. *3*, 919.

Torrontegui, E., and García-Ripoll, J.J. (2019). Unitary quantum perceptron as efficient universal approximator. Europhys. Lett. *125*, 30004.

Urbanczik, R., and Senn, W. (2014). Learning by the dendritic prediction of somatic spiking. Neuron *81*, 521–528.

Vaziri, A., and Plenio, M.B. (2010). Quantum coherence in ion channels: resonances, transport and verification. New J. Phys. *12*, 085001.

Weingarten, C.P., Doraiswamy, P.M., and Fisher, M.P.A. (2016). A new spin on neural processing: quantum cognition. Front. Hum. Neurosci. *10*, 541.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv, arXiv:cs.LG/1708.07747.

Xiao, J., Yan, Y., Zhang, J., and Tang, Y. (2010). A quantum-inspired genetic algorithm for k-means clustering. Exp. Syst. Appl. *37*, 4966–4973.

Zeng, Y., Zhang, T., and Xu, B. (2017). Improving multi-layer spiking neural networks by incorporating brain-inspired rules. Sci. China Inform. Sci. *60*, 052201.

Zenke, F., Agnes, E.J., and Gerstner, W. (2015). Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks. Nat. Commun. *6*, 6922.

Zhao, F., Zhang, T., Zeng, Y., and Xu, B. (2017). Towards a brain-inspired developmental neural network by adaptive synaptic pruning. In The 24th International Conference on Neural Information Processing (Springer), pp. 182–191.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and algorithms		
CNN	Lecun et al.,1998	https://github.com/tensorflow/docs-l10n/blob/master/site/zh-cn/tutorials/images/cnn.ipynb
VGG16	Simonyan and Zisserman, 2015	https://github.com/pytorch/vision/blob/6db1569c89094cf23f3bc41f79275c45e9fcb3f3/torchvision/models/vgg.py
ResNet	He et al., 2016	https://github.com/pytorch/vision/blob/6db1569c89094cf23f3bc41f79275c45e9fcb3f3/torchvision/models/resnet.py
DensNet	Huang et al., 2017	https://github.com/pytorch/vision/blob/6db1569c89094cf23f3bc41f79275c45e9fcb3f3/torchvision/models/densenet.py
QS-SNN	This paper	https://github.com/Brain-Inspired-Cognitive-Engine/Q-SNN
Other		
MNIST	LeCun et al., 2010	http://yann.lecun.com/exdb/mnist/
Fashion-MNIST	Xiao et al., 2017	https://github.com/zalandoresearch/fashion-mnist

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Yi Zeng (yi.zeng@ia.ac.cn).

Materials availability

This study did not generate new unique reagents.

Data and code availability

The Python scripts can be downloaded from the GitHub repository: https://github.com/Brain-Inspired-Cognitive-Engine/Q-SNN.

METHOD DETAILS

Generating background inverse image

Different degree of background color-inverse images $|I(\Theta)\rangle$ using for experiment are generated according to the quantum superposition encoding. Here we describe the numerical form value of quantum image used to test model. Suppose the original image is represented by X and the reversed image is $\overline{X}$. Parameter Θ controls the proportion of original image and reverse image in background color-inverse images with

$$I(\Theta) = X\cos(\Theta) + \overline{X}\sin(\Theta) \qquad \text{(Equation 20)}$$

Learning procedure of the QS-SNN algorithm

Dendrite prediction (Urbanczik and Senn, 2014) and proximal gradient methods are used for tuning of QS-SNN. In Equation (21), I_i is the teaching current, which is the integration of correct labels in $g_{E_i}(E_E - U_1)$ and wrong labels in $g_{I_i}(E_I - U_1)$. E_E (8 mV) and E_I (-8 mV) are the excitatory and inhibitory standard membrane potentials, respectively. The teaching current is injected to the soma of neurons

in the output layer, generating added potential V_{l_i} with membrane resistance r_B, as shown in Equation (22):

$$\begin{cases} I_i^{ject} = g_{E_i}(E_E - U_1) + g_{I_i}(E_I - U_1) \\ g_{E_i} = \begin{cases} 1, & i = label, \\ 0, & i \neq label. \end{cases} \\ g_{I_i} = \begin{cases} 0, & i = label, \\ 1, & i \neq label. \end{cases} \\ V_{l_i} = r_B \cdot I_i^{ject}, \end{cases} \quad \text{(Equation 21)}$$

$$\tau_L \frac{dV_i^o(t)}{dt} = -V_i^o(t) + \frac{g_B}{g_L}\left(V_i^{o,b} - V_i^o(t)\right) + V_{l_i} - V_i^o(t). \quad \text{(Equation 22)}$$

Setting the left side of Equation (22) to zero and $V_i^o = V_{l_i}$, we get the steady state of somatic potentials V_i^o and $V_i^{o,b}$ with $V_i^{o*} = g_B/(g_B + g_L)V_i^{o,b}$. The dendrite prediction rule defines the soma-dendrite error as Equation (23):

$$L = \frac{1}{2}\sum_{i=0}^{N}\left\|r_{max}\sigma\left(V_i^o\right) - r_{max}\sigma\left(V_i^{o*}\right)\right\|^2. \quad \text{(Equation 23)}$$

Minimizing this error based on the differential chain rule, we obtain updated synaptic weights w_{ij}^o, as shown in Equations (24) and (25):

$$\frac{\partial L}{\partial w_{ij}^o} = \frac{\partial L}{\partial V_i^{o,b}}\frac{\partial V_i^{o,b}}{\partial w_{ij}^o} \quad \text{(Equation 24)}$$

$$= r_{max}\frac{g_B}{g_B + g_L}\left[\sigma\left(V_i^{o*}\right) - \sigma\left(V_i^o\right)\right]\sigma'\left(V_i^{o*}\right)r_j$$

$$= \delta_i^o r_j,$$

$$\frac{\partial L}{\partial b_o^1} = \frac{\partial L}{\partial V_i^{o,b}}\frac{\partial V_i^{o,b}}{\partial b_i^o} \quad \text{(Equation 25)}$$

$$= r_{max}\frac{g_B}{g_B + g_L}\left[\sigma\left(V_i^{o*}\right) - \sigma\left(V_i^o\right)\right]\sigma'\left(V_i^{o*}\right)$$

$$= \delta_i^o.$$

Equation (26) shows the iterative updating of synaptic weights w_{ij}^y and bias b_i^y:

$$\begin{cases} w_{ij}^o \leftarrow w_{ij}^o - \eta\frac{\partial L}{\partial w_{ij}^o} \\ b_i^o \leftarrow b_i^o - \eta\frac{\partial L}{\partial b_i^o}. \end{cases} \quad \text{(Equation 26)}$$

For the hidden layer, error signal δ_i is passed from the previous layer, and neuron synapses w_{ij}^h are adapted using Equations (27) and (28):

$$\frac{\partial L}{\partial w_{ij}^h} = \sum_k \frac{\partial L}{\partial V_k^{o,b}}\frac{\partial V_k^{o,b}}{\partial r_i^h}\frac{\partial r_i^h}{\partial V_i^h}\frac{\partial V_i^h}{\partial V_i^{h,b}}\frac{\partial V_i^{h,b}}{\partial w_{ij}^h}$$

$$= \sum_k \delta_k^o w_{ki}^o r_{max}\frac{g_B}{g_B + g_L}\sigma'\left(V_i^h\right)V_j^{PSP} \quad \text{(Equation 27)}$$

$$= \sum_k \delta_k^o \delta_i^h w_{ki}^o V_j^{PSP},$$

$$\frac{\partial L}{\partial b_i^h} = \sum_k \frac{\partial L}{\partial V_k^{o,b}}\frac{\partial V_k^{o,b}}{\partial r_i^h}\frac{\partial r_i^h}{\partial V_i^h}\frac{\partial V_i^h}{\partial V_i^{h,b}}\frac{\partial V_i^{h,b}}{\partial b_i^h}$$

$$= \sum_k \delta_k^o w_{ki}^o r_{max}\frac{g_B}{g_B + g_L}\sigma'\left(V_i^h\right) \quad \text{(Equation 28)}$$

$$= \sum_k \delta_k^o \delta_i^h w_{ki}^o.$$

Equation (29) shows the iterative updating of synaptic weights w_{ij}^h and bias b_i^h:

$$\begin{cases} w_{ij}^h \leftarrow w_{ij}^h - \eta \dfrac{\partial L}{\partial w_{ij}^h} \\[2mm] b_i^h \leftarrow b_i^h - \eta \dfrac{\partial L}{\partial b_i^h}. \end{cases}$$

(Equation 29)

The training and test procedure for the QS-SNN model is shown in Algorithm 1.