A Memristive Spiking Neural Network Circuit with Selective Supervised Attention Algorithm

Zekun Deng, Chunhua Wang, Hairong Lin, *Member, IEEE,*
and Yichuang Sun, *Senior Member, IEEE,*

***Abstract*—Spiking neural networks (SNNs) are biologically plausible and computationally powerful. The current computing systems based on the von Neumann architecture are almost the hardware basis for the implementation of SNNs. However, performance bottlenecks in computing speed, cost, and energy consumption hinder the hardware development of SNNs. Therefore, efficient non-Neumann hardware computing systems for SNNs remain to be explored. In this paper, a selective supervised algorithm for spiking neurons inspired by the selective attention mechanism is proposed, and a memristive spiking neuron circuit as well as a memristive SNN circuit based on the proposed algorithm are designed. The memristor realizes the learning and memory of the synaptic weight. The proposed algorithm includes a top-down selective supervision method and a bottom-up selective supervision method. Compared with other supervised algorithms, the proposed algorithm has excellent performance on sequence learning. Moreover, top-down and bottom-up attention encoding circuits are designed to provide the hardware foundation for encoding external stimuli into top-down and bottom-up attention spikes, respectively. The proposed memristive SNN circuit can perform classification on the MNIST dataset and the Fashion-MNIST dataset with superior accuracy after learning a small number of labeled samples, which greatly reduces the cost of manual annotation and improves the supervised learning efficiency of the memristive SNN circuit.**

***Index Terms*—Selective attention, memristor, spiking neural network, circuit design, supervised algorithm, sequence learning, image classification.**

I. Introduction

SPIKES are considered to be an important carrier of neural information processing. As a computational model inspired by the brain, spiking neural networks (SNNs) are biologically plausible and computationally powerful [1]. SNNs are designed to simulate the function of the brain and imitate neural information processing methods. Since processing neural information by encoding precise temporal spike sequences, SNNs are considered to be more capable than other types of neural networks in performing temporal spike pattern recognition and real-time calculations [2]. Recently, many studies have combined SNNs into different new application scenarios such as batch normalization [3], dynamic vision sensing [4], 3D image recognition [5] and visual explanations [6], and these works provide effective methods for the application of large-scale SNNs.

At present, the computing systems based on the von Neumann architecture are almost the hardware basis for the implementation of the SNN [7]. Since the memory and processing units of the computing system are physically separated, storage and computing need to transmit a large amount of data to each other. With the significant growth of computing demands, the computing speed, cost and energy consumption of computing systems based on von Neumann architecture have reached performance bottlenecks [8]. Therefore, the study of efficient non-Neumann hardware computing systems has become one of the key methods to break through the performance bottleneck. In 1971, Chua [9] proposed a basic element that defines the relationship between electric charge and magnetic flux, and the element was named memristor. The memristor is a passive device with nonlinear and nonvolatile properties. Numerous studies have shown that the properties of memristor are similar to biological synapses, and the conductance of memristors can represent the weight of synapses [10]. In 2008, a nano memristor device was first realized at HP Labs [11]. Subsequently, various physical and mathematical memristor models have been proposed [12]–[16]. Recently, memristor-based neuron circuits and neural network circuits have been successively proposed [17]–[26]. Meanwhile, researchers have conducted many studies on memristive SNN circuits. For example, Hu et al. [27] proposed a memristor-based dynamic synapse for SNNs. Ankit et al. [28] proposed a reconfigurable and energy-efficient architecture (RESPARC) and designed an efficient memristor-based SNN circuit for recognition applications. Shukla et al. [29] proposed a clock-less SNN with memristive synapses for simultaneous learning and recognition. Zhao et al. [30] proposed a memristor-based SNN circuit with inhibitory synapses to realize the mechanisms of lateral inhibition and homeostasis.

Since SNNs have complex discontinuities and implicit nonlinear mechanisms, supervised learning algorithms used in traditional artificial neural networks are difficult to achieve the same effect in SNNs [31]. In the past few years, many studies on supervised learning methods of spiking neurons (SNs) have been proposed. For example, the researches of synaptic plasticity enlightened Ponulak and Kasiński [32] combined spike-timing-dependent plasticity (STDP) and anti-STDP to supervised training for SNs, and proposed a classical remote supervised method (ReSuMe) in 2010. Later, Xu et al. [33] proposed a perceptron-based SN learning rule (PBSNLR) with temporal encoding, which transforms supervised learning into

This work is supported by National Natural Science Foundation of China (62271197, 61971185, 62201204), the China Postdoctoral Science Foundation (2022M71104). (*Corresponding author: Chunhua Wang, email: wch1227164@hnu.edu.cn.*)

Zekun Deng, Chunhua Wang and Hairong Lin are with College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China.

Yichuang Sun is with the School of Engineering and Computer Science, University of Hertfordshire, Hatfield AL10 9AB, U.K.

a classification problem. In 2016, Lin et al. [34] proposed a spiking training kernel learning rule (STKLR) suitable for SNs. Futhermore, in the study of supervised learning methods for memristive SNN circuits, a supervised learning model was proposed in [35] to implement error backpropagation for the SNN circuit, where memristors act as electronic synaptic devices to store simulated synaptic weights. Recently, Chen et al. [36] designed a three-terminal memristive synapse in the SNN circuit, where the ReSuMe is performed in the SNN for image classification. Duan et al. [37] reported a SNN consisting of volatile NbO_x memristor-based neurons and nonvolatile TaO_x memristor-based synapses for supervised online learning. Zhou et.al [38] proposed a memristive spiking neural network trained with backpropagation through time (BPTT) learning rules for supervised learning.

The selective attention mechanism is the ability of biological vision to selectively process visual information in the environment. The selective attention mechanism can be divided into top-down (TD) approach for goal-driven mechanism and bottom-up (BU) approach for stimulus-driven mechanism [39], [40]. In visual scenes, the TD selective attention not only selectively enhances the attention of target objects, but also selectively suppresses neuronal responses evoked by other non-target objects. When the features of objects in the scene are relatively salient, the BU selective attention is automatically stimulated. The selective attention mechanism can effectively alleviate the problem of information overload, allowing the brain to dynamically learn real-time information with limited attention resources [41]. Recently, a memristor-based full-circuit implementation of transformer network (TN) with multi-heads and single attention layer was proposed [42], but it focus on the BU attention approach and lack of consideration for the TD attention approach. Therefore, how to combine the TD approach and the BU approach of the selective attention mechanism into the memristive SNN circuit is still an unsolved problem.

The main contributions of this paper are as follows:

1) Combining the selective attention mechanism with supervised learning, a selective supervised attention algorithm (SSAA) is proposed. The SSAA includes a TD selective supervision method and a BU selective supervision method. Compared with other supervised algorithms, the SSAA has faster learning speed, less learning time and higher learning accuracy on sequence learning.
2) Based on the SSAA and memristors, a selective supervised memristive spiking neuron citcuit (SSMSNC) is designed, and a selective supervised memristive spiking neural network citcuit (SSMSNNC) is constructed for image classification. In addition, TD and BU attention encoding circuits are designed to extract and encode attention spikes from images.
3) The SSMSNNC can perform accurate classification on the MNIST dataset and the Fashion-MNIST dataset after learning a small number of labeled samples, and has superior classification accuracy compared with other memristive SNN circuits. This work greatly reduces the cost of manual annotation and improves the supervised learning efficiency of memristive SNN circuits.

The rest of this paper is organized as follows. In Section II, the design of the SSAA is proposed, and performance analysis and comparison of sequence learning are performed. In Section III, the memristor model used in PSPICE simulations is introduced, and then the SSMSNC is designed and analyzed. In Section IV, attention encoding circuits for the SSMSNC are designed and analyzed. In Section V, application of the SSMSNNC in image classification is presented, and the robustness for classification is analyzed. In Section VI, power consumption, parameter effects, circuit comparison and non-ideality are discussed. Finally, Section VII summarizes the work of this paper.

II. Algorithm Design and Analysis

A. The Selective Supervised Attention Algorithm

With limited information processing resources, the human eyes receive tens of millions of bits of visual information from the real world. The selective attention mechanism of human vision tends to focus on the desired captured information and ignore relatively irrelevant information, thereby reducing the amount of information that needs to be processed and recognizing the external environment more quickly [43].

Inspired by the selective attention mechanism of biological vision principles, the SSAA combines the TD attention approaches of the goal-driven mechanism and the BU attention approaches of the stimulus-driven mechanisms to dynamically modify synaptic weights of the SN. The working flow of the SSAA is shown in Fig. 1. The SSAA includes a TD selective supervised attention algorithm (TDSSAA) and a BU selective supervised attention algorithm (BUSSAA). All samples need to be encoded to make the TDSSAA and the BUSSAA work correctly. The encoding method for samples is introduced in Section IV.

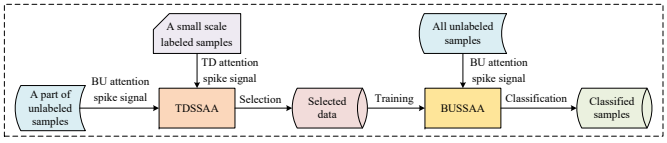


Fig. 1. The working flow of the SSAA. TD attention spike signals are obtained from a small scale of labeled samples. In the learning stage, the TDSSAA selects images similar to TD attention spikes from the BU attention spike signals of a part of unlabeled samples. The BU attention spike signals of the selected samples constitute the selected data, which are used as training data for the BUSSAA to modify synaptic weights. After training, the BUSSAA classifies all unlabeled samples in the testing stage.

The number of synapses in the SN is set to n, each synapse receives corresponding synaptic spikes in parallel. The SSAA is defined as:

$$\Delta w_i(t) = (-1)^{\left(1-x_i(t)\right)} \alpha_i x_{\mathrm{s}i}(t) x_{\mathrm{c}}(t), \tag{1}$$

$$x_{\mathrm{s}i}(t) = 1 - x_{\mathrm{TDSSAA}}(t)\left(1 - x_{\mathrm{BU}i}(t)\right), \tag{2}$$

$$x_i(t) = \begin{cases} x_{\mathrm{TD}i}(t) x_{\mathrm{BU}i}(t), & x_{\mathrm{TDSSAA}}(t) = 1 \\ x_{\mathrm{BU}i}(t), & x_{\mathrm{TDSSAA}}(t) = 0, \end{cases} \tag{3}$$

where $\Delta w_i(t)$ is the variation of the ith synaptic weight, α_i is the learning rate of the ith synaptic weight. $x_i(t)$ is the synaptic control signal of the ith synapse, where $i = 1, 2, \ldots, n$. $x_{\mathrm{c}}(t)$ is the control signal for the SSAA to learn or test. If $x_{\mathrm{c}}(t) = 1$, the SSAA performs learning and the synaptic learning occurs, otherwise the SSAA performs testing and synaptic weights are unchanged. $x_{\mathrm{TD}i}(t)$ and $x_{\mathrm{BU}i}(t)$ are the TD and BU attention spike signals of the ith synapse, respectively. $x_{\mathrm{TDSSAA}}(t)$ is a pulse signal for the SSAA to perform the TDSSAA. $x_{\mathrm{s}i}(t)$ is the transmission control signal of the ith synapse. In the SSAA, the amplitudes of all spikes and pulses are set to 1.

The presence or absence of pulses in $x_{\mathrm{TDSSAA}}(t)$ is the criterion for distinguishing between TDSSAA and BUSSAA. During the pulse of $x_{\mathrm{TDSSAA}}(t)$, the SSAA performs the TDSSAA. If a spike occurs in $x_{\mathrm{TD}i}(t)$ and $x_{\mathrm{BU}i}(t)$ simultaneously, then a spike is generated in $x_i(t)$. Without the pulse of $x_{\mathrm{TDSSAA}}(t)$, the SSAA performs the BUSSAA, and the spike occurring in $x_i(t)$ is consistent with $x_{\mathrm{BU}i}(t)$. The transmission control signal $x_{\mathrm{s}i}(t)$ of the ith synapse is produced by the interaction of $x_{\mathrm{TDSSAA}}(t)$ and $x_{\mathrm{BU}i}(t)$. If $x_{\mathrm{TDSSAA}}(t) = 1$ and $x_{\mathrm{BU}i}(t) = 0$, then $x_{\mathrm{s}i}(t) = 0$, the transmission channel of the ith synapse is turned off. In other cases, the transmission channel of the ith synapse is turned on.

Algorithm 1 The Selective Supervised Attention Algorithm

$i \leftarrow 1$
while $i \leq n$ **do**
  if $x_c(t) = 1$ **then**
    if $x_{\mathrm{TDSSAA}}(t) = 1$ **then**
      $x_i(t) = x_{\mathrm{TD}i}(t) x_{\mathrm{BU}i}(t)$
    end if
    if $x_{\mathrm{TDSSAA}}(t) = 0$ **then**
      $x_i(t) = x_{\mathrm{BU}i}(t)$
    end if
    if $x_{\mathrm{TDSSAA}}(t) \neq 1$ and $x_{\mathrm{BU}i}(t) \neq 0$ **then**
      $x_{\mathrm{s}i}(t) = 1$
    end if
    if $x_{\mathrm{TDSSAA}}(t) = 1$ and $x_{\mathrm{BU}i}(t) = 0$ **then**
      $x_{\mathrm{s}i}(t) = 0$
    end if
    if $x_{\mathrm{s}i}(t) = 1$ **then**
      if $x_i(t) = 1$ **then**
        $\Delta w_i = \alpha_i$
      end if
      if $x_i(t) = 0$ **then**
        $\Delta w_i = -\alpha_i$
      end if
    end if
    if $x_{\mathrm{s}i}(t) = 0$ **then**
      $\Delta w_i = 0$
    end if
  end if
  if $x_c(t) = 0$ **then**
    $\Delta w_i = 0$
  end if
end while

In the learning stage of the SSAA, if the transmission channel of the ith synapse is turned on, then the synaptic weight of the ith synapse increases corresponding to $x_i(t) = 1$ and decreases corresponding to $x_i(t) = 0$. The weights of all synapses are modified globally in parallel during the learning stage. The detailed process of synaptic weight modification in the SSAA is shown in Algorithm 1.

During the learning stage and the testing stage, the output signals of the SSAA is defined as:

$$x_{\mathrm{o}}(t) = \begin{cases} \partial_1 \sum_{i=1}^{n} \big(w_i(t) x_{\mathrm{s}i}(t)\big), & x_{\mathrm{c}}(t) = 1 \\ \partial_2 \sum_{i=1}^{n} \big(w_i(t) x_{\mathrm{s}i}(t) x_i(t)\big), & x_{\mathrm{c}}(t) = 0, \end{cases} \tag{4}$$

$$s_{\mathrm{o}}(t) = \begin{cases} 1, & x_{\mathrm{o}}(t) \geq x_{\mathrm{th}} \\ 0, & x_{\mathrm{o}}(t) < x_{\mathrm{th}}, \end{cases} \tag{5}$$

where $x_{\mathrm{o}}(t)$ is the output signal of the SSAA, $w_i(t)$ is the weight of the ith synapse. ∂_1 is the output gain coefficient in the case of $x_{\mathrm{c}}(t) = 1$ and ∂_2 is the output gain coefficient in the case of $x_{\mathrm{c}}(t) = 0$. $s_{\mathrm{o}}(t)$ is the output spikes of the SSAA after the threshold comparison of $x_{\mathrm{o}}(t)$, and x_{th} is the threshold of the SSAA.

When $x_{\mathrm{c}}(t) = 1$, all synaptic output signals are accumulated with the output gain coefficient of ∂_1 to obtain $x_{\mathrm{o}}(t)$. When $x_{\mathrm{c}}(t) = 0$, only synaptic output signals under the condition of $x_i(t) = 1$ are accumulated with the output gain coefficient of ∂_2 to get $x_{\mathrm{o}}(t)$. If $x_{\mathrm{o}}(t) \geq x_{\mathrm{th}}$, then a spike occurs in $s_{\mathrm{o}}(t)$, otherwise no spike occurs.

B. Sequence Learning and Performance Analysis

In order to quantitatively evaluate the learning accuracy of sequence learning, a correlation-based metric C is used to represents the correlation between two spike trains [44]. C is calculated as follows:

$$C = \frac{\overrightarrow{h_1} \cdot \overrightarrow{h_2}}{\left|\overrightarrow{h_1}\right| \left|\overrightarrow{h_2}\right|}, \tag{6}$$

where $\overrightarrow{h_1}$ and $\overrightarrow{h_2}$ are the vectors representing a convolution of the two spike trains with a low-pass Gaussian filter. $\overrightarrow{h_1} \cdot \overrightarrow{h_2}$ is the inner product. $\left|\overrightarrow{h_1}\right|$ and $\left|\overrightarrow{h_2}\right|$ are the Euclidean norms of $\overrightarrow{h_1}$ and $\overrightarrow{h_2}$, respectively. The Guassian filter function with the parameter σ is given by $f(t, \sigma) = exp(-t^2/2\sigma^2)$, where σ is the width of the Guassian filter function. The two spike trains are uncorrelated at $C = 0$, and completely correlated at $C = 1$.

To analyze the learning performance of the SSAA, we perform supervised sequence learning on multi-spike input pattern by the TDSSAA. Some work uses the SNNs trained by DNN-to-SNN conversion for supervised learning [45], [46], but the key disadvantage of these SNNs is high latency. Although a DNN-to-SNN conversion and finetuning algorithm was proposed in [47] to reduce the conversion error for small number of time steps by minimizing the difference between SNN and DNN activation functions. However, compared with the SNNs that directly implement supervised algorithms, the SNNs trained by DNN-to-SNN conversion still have higher conversion costs. In traditional supervised learning methods, the aim of sequence learning is to make the SN learn and fire a specific spike train with the precise firing times of spikes [32], [33]. If running time is considered, supervised

learning of SNs is equivalent to temporal spike learning and classification problems. As we all know, the human brain has a high degree of parallelism, where neurons and synapses need to process large amounts of information in parallel. Obviously, temporal spike learning cannot meet the needs of large-scale real-time and parallel information processing. Different from the traditional supervised learning methods of the SN, the desired output spike pattern in our algrithm is input into each synapse in parallel at each learning epoch. In traditional supervised learning methods, the length of the spike train is determined by the length of time. In our algorithm, the length of the spike train is defined by the number of spike sites that occur at the same time. Due to the parallel computation, the number of spike sites for the desired output spike pattern and input spike pattern is consistent with the number of synapses, and one learning epoch in our algorithm only occupies one spike site in each pattern. Different input spike patterns are input into the corresponding synapses at different learning epochs. The schematic diagram of sequence learning in our algorithm is shown in Fig. 2. Therefore, all synaptic weights are modified simultaneously according to the TDSSAA in each learning epoch, and the output spike pattern is obtained in each learning epoch.

Since the modification of the synaptic weight corresponds to the adjustment of the output result, the synaptic weight is served as the basis for the output spike as follows:

$$sw_i(t) = \begin{cases} 1, & \partial_3 w_i(t) \geq xw_{thi} \\ 0, & \partial_3 w_i(t) < xw_{thi}, \end{cases} \tag{7}$$

where ∂_3 is the output gain coefficient of synapses. xw_{thi} is the threshold of the ith synapse. If $\partial_3 w_i(t) \geq xw_{thi}$, then a spike occurs in the $sw_i(t)$, otherwise no spike occurs.

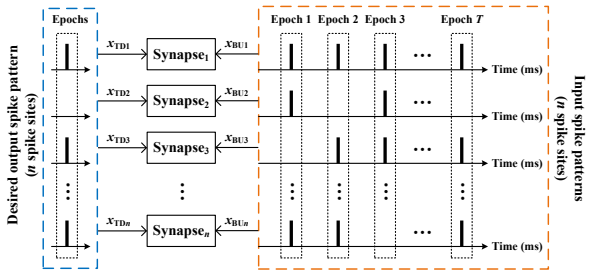


Fig. 2. The illustration of sequence learning for the SSAA. The length of the spike train is defined by the number of spike sites. The number of spike sites for the desired output spike pattern and input spike patterns is consistent with the number of synapses, and one learning epoch only occupies one spike site in each pattern. The time sequences of the desired output spikes are input into synapses as the TD attention spike signals, and the time sequences of the input spikes are input into synapses as the BU attention spike signals.

We define that C is calculated from the desired output spike pattern and the output spike pattern. Fig. 3(a) and (b) show the desired output spike pattern and the input spike patterns for simulation. The simulated output spike patterns are shown in Fig. 3(c). In sequence learning of the TDSSAA, $x_{\mathrm{c}}(t) = 1$ and $x_{\mathrm{TDSSAA}}(t) = 1$. The simulation parameters of the SSAA for sequence learning are set as follows: $\alpha_i = 0.1$, $\partial_3 = 0.1$, $xw_{\mathrm{th}i} = 0.05 \ \ (i = 1, 2, \ldots, n)$. In the simulations, the SSAA with 300 synaptic inputs is trained on the spike patterns of the desired output spikes and the input spikes with the Poisson process at about 150 Hz. The spike duration is set to 0.1 ms, and a learning epoch is set to 1 ms. The simulation time step is set to 1 ms, thus the number of time steps is consistent with the number of learning epochs. The spikes are fired at the corresponding spike sites. Since the physical implementation of synaptic weights has weight boundaries, such as memristors, therefore, the variation range of the synaptic weights in the SSAA is set to [0, 1].

The initial synaptic weights are set as the random distribution in the interval [0, 1], as shown in Fig. 4(a). The synaptic weights after the simulation are shown in Fig. 4(b), where the synaptic weights corresponding to the desired output spikes are trained to reach the maximum weight of 1, and the other synaptic weights are reduced to the minimum weight of 0. This result reflects selective modification of synapses. As shown in Fig. 4(b), the output spike pattern exactly matches the desired output spike pattern at the 17th learning epoch, and the learning accuracy reaches $C = 1$. Since one learning epoch in the TDSSAA is 1 ms, the learning time of sequence learning is greatly saved and the learning efficiency is greatly improved.

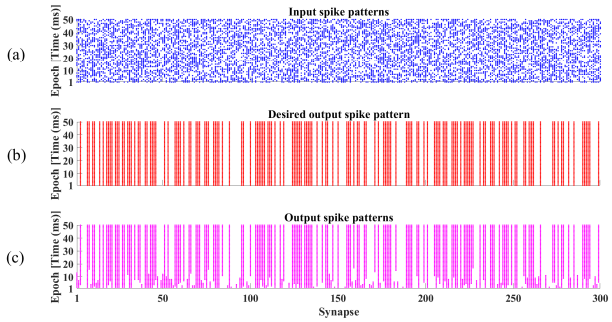


Fig. 3. The schematic diagram of spike patterns. (a) The input spike patterns. (b) The desired output spike pattern. (c) The output spike patterns.

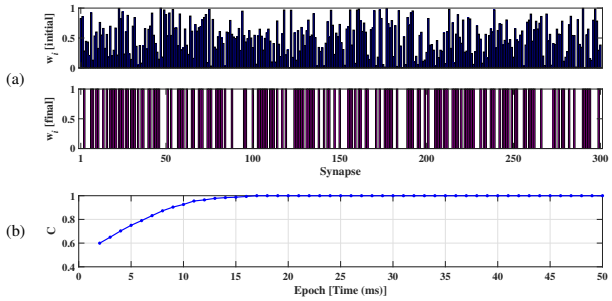


Fig. 4. The simulation results of sequence learning. (a) Initial synapse weights and final synapse weights. (b) The learning accuracy of sequence learning.

Considering the influence of parameters on the learning performance, learning epochs required for the SSAA to learn to reach $C = 1$ under different parameters are simulated, as shown in Fig. 5. With the increase of number of synapses in Fig. 5(a), learning epochs to reach $C = 1$ increase accordingly, and the floating range of learning epochs also increases accordingly. Thereby the more the number of synapses, the longer the learning time. In Fig. 5(b), the number of synapses is set to 500. As α_i $(i = 1, 2, \ldots, n)$ increases, learning epochs to reach $C = 1$ decrease rapidly, and the floating range of learning epochs decreases accordingly. Too small α_i makes

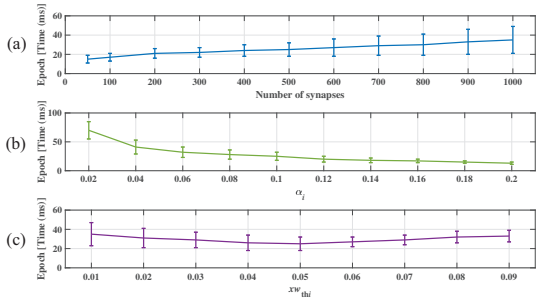


Fig. 5. Learning performance analysis under different parameters. (a) Different number of synapses. (b) Different learning rate α_i $(i = 1, 2, \ldots, n)$ of synaptic weights. (c) Different synaptic threshold xw_{thi} $(i = 1, 2, \ldots, n)$ of synapses.

the update of synaptic weights slow, and the time cost of learning is greater. While too large α_i leads to fast update of synaptic weights, the fine weight modification process cannot be observed. In Fig. 5(c), the number of synapses is set to 500. As xw_{thi} $(i = 1, 2, \ldots, n)$ increases, learning epochs to reach $C = 1$ decrease before $xw_{\mathrm{thi}} = 0.05$ and increases afterwards. The floating range of learning epochs decreases before $xw_{\mathrm{thi}} = 0.07$ and increases afterwards. Therefore, how to balance the number of synapse, α_i and xw_{thi} to achieve optimal learning in practical applications is a key consideration.

C. Comparison of Learning Performance

In this section, the supervised learning performances of the TDSSAA, the ReSuMe, the PBSNLR and the STKLR are compared. The firing of spikes at equidistant time points in the other algorithms corresponds to the firing of spikes at the spike site of the spike pattern in the TDSSAA. First, the learning performances of the four algorithms with different lengths of spike trains are compared. In the simulations, the length of the spike train ranges from 400 ms to 2800 ms with the time interval of 400 ms. The firing rate of spike trains is set to $\frac{1}{2L_s}$, where L_s is the length of spike train. The generation of all spikes follows the homogeneous Poisson process. As shown in Fig. 6(a), the TDSSAA can reach the maximum learning accuracy $C = 1$ in any length of the spike trains, while the maximum learning accuracy of the other three algorithms decreases gradually with the increase of the sequence length. Meanwhile, learnnig epochs of the TDSSAA compared to the other three algorithms are also the least as shown in Fig. 6(b).

Second, the learning performances of the four algorithms with different firing rates of spike trains are compared. In the simulations, the number of synapses is set to 500. The firing rate of the spike train ranges from 100 Hz to 400 Hz with the rate interval of 100 Hz. The generation of all spikes follows the homogeneous Poisson process. As shown in Fig. 7, the TDSSAA can always reach the maximum learning accuracy $C = 1$ at different firing rates, while the maximum learning accuracy of the other three algorithms are limited by the firing rate of the spike train. Learning epochs of the other three algorithms are significantly larger than that of the TDSSAA.

Therefore, compared with the other traditional supervised algorithms, the proposed SSAA has better performance in learning speed, learning time and learning accuracy.

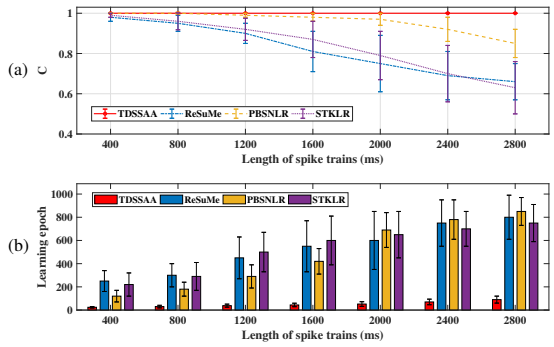


Fig. 6. The comparison of the learning performance in the different length of spike trains. (a) Learning accuracy. (b) Learnnig epoch.

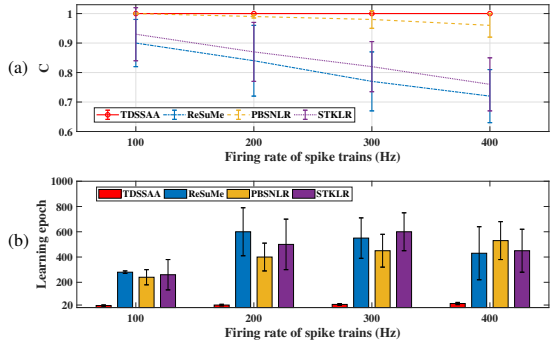


Fig. 7. The comparison of the learning performance in the different firing rate of spike trains. (a) Learning accuracy. (b) Learnnig epoch.

III. Circuit Design and Analysis of the SSMSNC

A. Memristor Model

In recent years, many memristor models have been proposed based on the experimental data or device characteristics, such as TiO_2 memristor model [48], TEAM model [49], VTEAM model [50] and AIST-based memristor model [14]. The TiO_2 memristor model lacks threshold voltages and thus cannot keep the memristance unchanged in the presence of an input voltage. The TEAM model is current-controlled, which is inconsistent with most practical voltage-controlled memristive devices. The VTEAM model is an extension of the TEAM model to a voltage-controlled memristor model, but the derivative of its state variable is constant when the input voltage is a pulse. The AIST-based memristor model is a voltage-controlled threshold memristor model in which the memristance could be modified only when the applied voltage exceeds the threshold, and the memristance can be controlled by the spike voltage. The AIST-based memristors can describe memristor devices at a level of abstraction sufficient for efficient circuit simulation, which is well suited for the simulation of memristive SNN circuit. The AIST-based memristor model is expressed as:

$$\frac{\mathrm{d}w(t)}{\mathrm{d}t} = \begin{cases} \mu_{\mathrm{v}} \frac{R_{\mathrm{ON}}}{D} \frac{i_{\mathrm{off}}}{i(t)-i_0} F\left(w(t)\right), & v(t) > V_{\mathrm{T+}} > 0 \\ 0, & V_{\mathrm{T+}} \geq v(t) \geq V_{\mathrm{T-}} \\ \mu_{\mathrm{v}} \frac{R_{\mathrm{ON}}}{D} \frac{i(t)}{i_{\mathrm{on}}} F\left(w(t)\right), & 0 > V_{\mathrm{T-}} > v(t), \end{cases} \tag{8}$$

$$f\left(w(t)\right) = 1 - \left(\frac{2w(t)}{D} - 1\right)^{2p}, \tag{9}$$

where i_0, $i_{\rm off}$, and $i_{\rm on}$ are constants, $w(t)$ donates the state variable of memristor, $\mu_{\rm v}$ is the average ion mobility and D is semiconductor film thicknesses. $R_{\rm ON}$ and $R_{\rm OFF}$ are low memristance and high memristance of the memristor respectively. $f(w(t))$ is a window function. p is a positive integer. $V_{\rm T+}$ and $V_{\rm T-}$ are positive and negative threshold voltages, respectively. The PSPICE simulation parameters of memristors are shown in Table I.

TABLE I
SIMULATION PARAMETERS OF MEMRISTORS

Parameter	Value
$R_{\rm ON}$ ($k\Omega$)	1
$R_{\rm OFF}$ ($k\Omega$)	10
$V_{\rm T+}$ (V)	0.05
$V_{\rm T-}$ (V)	-0.05
$\mu_{\rm v}$ ($\rm m^2 s^{-1} \Omega^{-1}$)	1e-12
D (nm)	10
i_0 (A)	1e-6
$i_{\rm off}$ (A)	5e-3
$i_{\rm on}$ (A)	5e-8
p	10

The change curves of the simulated memristor are shown in Fig. 8. The simulation results show that if the applied voltage is positive and exceeds the positive threshold voltage of the memristor, the memristance decreases. If the applied voltage is negative and exceeds the negative threshold voltage of the memristor, the memristance increases. Furthermore, with a positive voltage applied, the memristance decreases rapidly and then slowly. With a negative voltage applied, the memristance increases rapidly and then slowly. These properties of the simulated memristor are consistent with the actual memristor device.

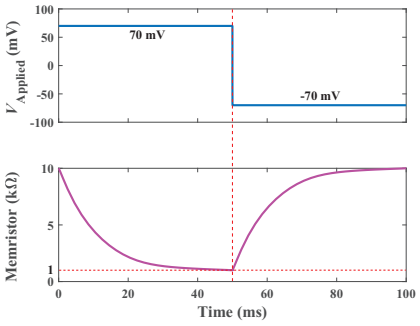


Fig. 8. The change curves of the simulated memristor under applied positive and negative voltages.

B. Circuit Design of the SSMSNC

Based on the SSAA, we design a memristive SN circuit as shown in Fig. 9, which consists of a memristive synapse module, a power supply module and an output module.

The memristive synapse module contains n memristive synapses (MSs) for parallel computing. The circuit of the ith memristive synapse ($\rm MS_{\it i}$) consists of a memristor ($\rm M_{\it i}$), five NMOSs ($\rm N_{s\it i}$, $\rm N_{\it i1}$, $\rm N_{\it i2}$, $\rm N_{\it i3}$, $\rm N_{\it i4}$) and two CMOS NOT logic gates. $\rm M_{\it i}$ is the memristor of $\rm MS_{\it i}$, and the conductance of $\rm M_{\it i}$ represents the synaptic weight. $x_i(t)$ is the synaptic control signal of the $\rm MS_{\it i}$, which is used to control the direction of the current flowing through the $\rm MS_{\it i}$. $x_{{\rm s}i}(t)$ is the control signal of $\rm N_{s\it i}$ in the $\rm MS_{\it i}$.

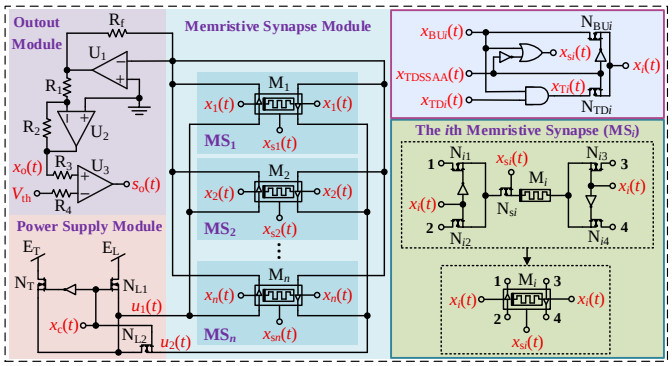


Fig. 9. The structure diagram of the SSMSNC. The ith memristive synapse in the memristive synapse module is illustrated at the bottom right. The power supply module and the output module are indicated in the SSMSNC. The generation circuit of $x_{{\rm s}i}(t)$ and $x_i(t)$ are displayed at the upper right.

The generation circuit for $x_i(t)$ is illustrated in the upper right of Fig. 9, where $x_{{\rm TD}i}(t)$ and $x_{{\rm BU}i}(t)$ are the TD attention and BU attention spike signal of the $\rm MS_{\it i}$, respectively. $x_{\rm TDSSAA}(t)$ is the pulse signal for the SSMSNC to learn with the TDSSAA. $x_{{\rm s}i}(t)$ is the result of the OR logic operation between $x_{{\rm BU}i}(t)$ and NOT-$x_{\rm TDSSAA}(t)$. $x_{{\rm T}i}(t)$ is the result of the AND logic operation between $x_{{\rm TD}i}(t)$ and $x_{{\rm BU}i}(t)$. $\rm N_{BU\it i}$ and $\rm N_{TD\it i}$ are NMOSs.

In the power supply module, the two supply voltages ($\rm E_L$, $\rm E_T$) are the learning voltage and testing voltage of the SSMSNC, respectively. The synaptic supply voltage is recorded as $u_1(t)$ and $u_2(t)$. $\rm N_{L1}$, $\rm N_{L2}$ and $\rm N_T$ are NMOSs controlled by $x_{\rm c}(t)$. $\rm N_{L1}$ and $\rm N_{L2}$ are turned on during the learning stage and turned off during the testing stage. $\rm N_T$ is turned on during the testing stage and turned off during the learning stage.

In the output module, the operational amplifier (OPAMP) $\rm U_1$, the resister $\rm R_f$ and MSs form an inverting summation OPAMP circuit that accumulates the synaptic voltage of MSs. Then the inverting voltage is input to an inverting OPAMP circuit to obtain the output signal $x_{\rm o}(t)$ of the SSMSNC. The OPAMP $\rm U_3$ and the resisters $\rm R_3$ and $\rm R_4$ constitute a comparison OPAMP circuit, and the comparison threshold voltage is $V_{\rm th}$. If the voltage of $x_{\rm o}(t)$ is greater than that of $V_{\rm th}$, then a spike occurs in the spike output signal $s_{\rm o}(t)$, otherwise no spike occurs.

The circuit simulations are all completed in PSPICE. In the simulations of the SSMSNC, $E_{\rm T} = 0.03$ V, $E_{\rm L} = 0.07$ V, R_1, R_2, R_3 and R_4 are set to 1 $k\Omega$. $R_{\rm f}$ is adjusted with the number of MSs. In addition, the positive supply of $\rm U_1$, $\rm U_2$, $\rm U_3$ and CMOS logic gates is $+5$ V, the negative supply of $\rm U_1$ and $\rm U_2$ is -5 V, and the negative supply of $\rm U_3$ and CMOS logic gates is electrical grounding. The turn-on voltages of CMOS logic gates and NMOSs are 3 V. In the SSMSNC, the amplitudes of all spikes and pulses are set to 5 V.

C. Circuit Analysis of the SSMSNC

In the power supply module, $x_{\rm c}(t)$ controls the turn-on and turn-off of the NMOSs $\rm N_{L1}$, $\rm N_{L2}$ and $\rm N_T$. If $x_{\rm c}(t) = 5$ V,

the SSMSNC is in the learning stage. $\mathrm{N_{L1}}$ and $\mathrm{N_{L2}}$ are turned on, and $\mathrm{N_T}$ is turned off, then $u_1(t) = u_2(t) = E_\mathrm{L}$. Since $|E_\mathrm{L}|$ is larger than the threshold voltages $|V_{\mathrm{T}+}|$ and $|V_{\mathrm{T}-}|$ of memristors, the weights of MSs are modified during the learning stage. If $x_\mathrm{c}(t) = 0$ V, the SSMSNC is in the testing stage. $\mathrm{N_T}$ is turned on, $\mathrm{N_{L1}}$ and $\mathrm{N_{L2}}$ are turned off, then $u_1(t) = E_\mathrm{T}$ and $u_2(t) = 0$ V. Since $|E_\mathrm{T}|$ is smaller than the threshold voltages $|V_{\mathrm{T}+}|$ and $|V_{\mathrm{T}-}|$ of memristors. Thereby the weights of MSs are unchanged under the testing stage. The simulation results of the power supply module are shown in Fig. 10.

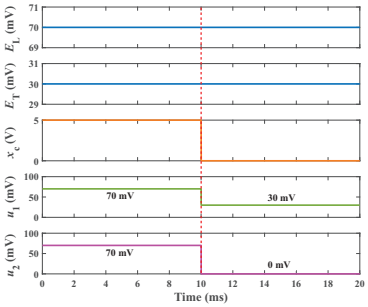


Fig. 10. The simulation results of the power supply module. In the first 10 ms, $x_\mathrm{c}(t) = 5$ V. In the last 10 ms, $x_\mathrm{c}(t) = 0$ V.

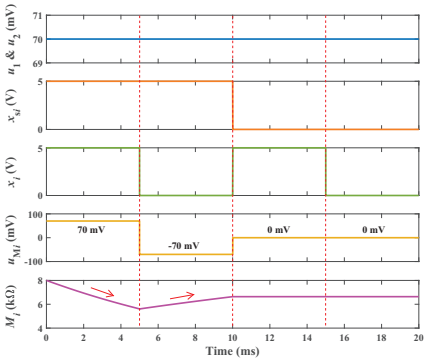


Fig. 11. The simulation results of the MS_i circuit in the learning stage. During the first 5 ms, $x_{\mathrm{s}i}(t) = 5$ V and $x_i(t) = 5$ V, M_i decreases. During 5 ms to 10 ms, $x_{\mathrm{s}i}(t) = 5$ V and $x_i(t) = 0$ V, M_i increases. During 10 ms to 20 ms, $x_{\mathrm{s}i}(t) = 0$ V, M_i is unchanged.

In the MS_i circuit, $x_i(t)$ controls the turn-on and turn-off of the NMOSs N_{i1}, N_{i2}, N_{i3} and N_{i4}, and $x_{\mathrm{s}i}(t)$ controls the turn-on and turn-off of the NMOS $\mathrm{N_{s}}i$. If $x_{\mathrm{s}i}(t) = 5$ V and $x_i(t) = 5$ V, then $\mathrm{N_{s}}i$, N_{i2} and N_{i3} are turned on and the synaptic current is transmitted in the path of $3 \rightarrow 2$. If $x_{\mathrm{s}i}(t) = 5$ V and $x_i(t) = 0$ V, then $\mathrm{N_{s}}i$, N_{i1} and N_{i4} are turned on and the synaptic current is transmitted in the path of $4 \rightarrow 1$. In the learning stage, if $x_{\mathrm{s}i}(t) = 5$ V and $x_i(t) = 5$ V, then the applied voltage of the memristor M_i is $u_1(t)$ and exceeds the positive threshold voltage of memristor. Thereby the memristance of M_i decreases and $\mathrm{G_{M}}i$ increases. If $x_{\mathrm{s}i}(t) = 5$ V and $x_i(t) = 0$ V, then the applied voltage of the memristor M_i is $u_2(t)$ and exceeds the negative threshold voltage of memristor. Thus the memristance of M_i increases and $\mathrm{G_{M}}i$ decreases. In addition, if $x_{\mathrm{s}i}(t) = 0$ V, then the M_i disconnect from the SSMSNC, and no current flows in the MS_i. In the testing stage, the memristance of M_i is unchanged. The simulation results of the MS_i circuit are shown in Fig. 11, where the initial memristance of the M_i is set to 8 $k\Omega$.

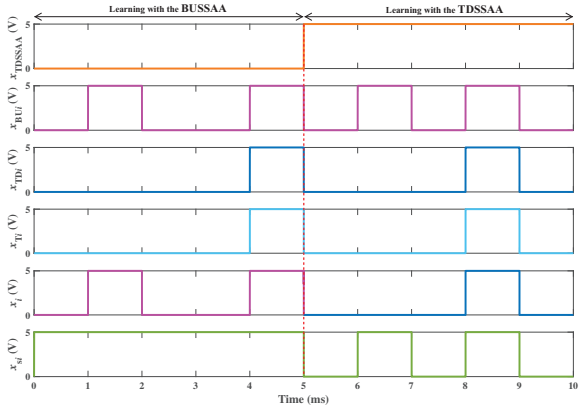


Fig. 12. The simulation results of the signal generation of $x_{\mathrm{s}i}(t)$ and $x_i(t)$. During the first 5 ms, the SSMSNC learns with BUSSAA due to $x_{\mathrm{TDSSAA}}(t) = 0$ V, thus $x_i(t)$ is the same as $x_{\mathrm{BU}i}(t)$ and $x_{\mathrm{s}i}(t) = 5$ V. During the last 5 ms, the SSMSNC learns with TDSSAA due to $x_{\mathrm{TDSSAA}}(t) = 5$ V, and $x_i(t)$ is the same as $x_{\mathrm{T}i}(t)$. $x_{\mathrm{T}i}(t) = 5$ V is obtained by $x_{\mathrm{TD}i}(t) = 5$ V and $x_{\mathrm{BU}i}(t) = 5$ V. When $x_{\mathrm{TDSSAA}}(t) = 5$ V and $x_{\mathrm{BU}i}(t) = 0$ V, $x_{\mathrm{s}i}(t) = 0$ V.

In the generation circuit of $x_i(t)$, on one hand, $\mathrm{N_{TD}}i$ is turned on and $\mathrm{N_{BU}}i$ is turned off in the pulse of $x_{\mathrm{TDSSAA}}(t)$, then $x_{\mathrm{TD}i}(t)$ and $x_{\mathrm{BU}i}(t)$ jointly input an AND logic gate and output $x_{\mathrm{T}i}(t)$ as $x_i(t)$. On the other hand, without the pulse of $x_{\mathrm{TDSSAA}}(t)$, $\mathrm{N_{BU}}i$ is turned on and $\mathrm{N_{TD}}i$ is turned off, then $x_i(t)$ is the same as $x_{\mathrm{BU}i}(t)$. In addition, $x_{\mathrm{TDSSAA}}(t)$ is inverted by the NOT logic gate and then input to the OR logic gate together with $x_{\mathrm{BU}i}(t)$, and the output signal is $x_{\mathrm{s}i}(t)$. The simulation results of the signal generation of $x_{\mathrm{s}i}(t)$ and $x_i(t)$ are shown in Fig. 12.

In the output module, the output voltage $x_\mathrm{o}(t)$ of the SSMSNC can be expressed as:

$$\begin{aligned} x_\mathrm{o}(t) = &\sum_{r=1}^{P} \Big(\frac{R_\mathrm{f} R_2}{M_r(t) R_1} X_{\mathrm{s}r}(t) \Big) u_1(t) \\ &+ \sum_{q=1}^{Q} \Big(\frac{R_\mathrm{f} R_2}{M_q(t) R_1} X_{\mathrm{s}q}(t) \Big) u_2(t), \end{aligned} \tag{10}$$

where P is the number of MSs with the applied voltage $u_1(t)$, Q is the number of MSs with the applied voltage $u_2(t)$, and $P + Q = n$. $X_{\mathrm{s}i}(t) = \frac{x_{\mathrm{s}i}(t)}{A}$, where A is the amplitude of spikes in $x_{\mathrm{s}i}(t)$. If the memristance increases, then the synaptic weight decreases and $x_\mathrm{o}(t)$ decreases. If the memristance decreases, then the synaptic weight increases and $x_\mathrm{o}(t)$ increases. In the learning stage, since $x_\mathrm{c}(t) = 5$ V, then $u_1(t) = u_2(t) = E_\mathrm{L}$. Thereby the voltage of $x_\mathrm{o}(t)$ is accumulated by all MSs. In the testing stage, since $x_\mathrm{c}(t) = 0$ V, then $u_1(t) = E_\mathrm{T}$ and $u_2(t) = 0$ V. Thereby the voltage of $x_\mathrm{o}(t)$ will only be accumulated by the MSs under the condition of $x_i(t) = 5$ V. In addition, if $x_{\mathrm{s}i}(t) = 5$ V, the MS_i is turned on and participates in the voltage accumulation of $x_\mathrm{o}(t)$. If $x_{\mathrm{s}i}(t) = 0$ V, the MS_i is turned off and out of action.

The simulation results of the SSMSNC for the TDSSAA learning are shown in Fig. 13, where the number of MSs in the SSMSNC is set to $n = 5$, R_f is set to 100 Ω and the initial memristances of the memristors are set randomly. In each learning period, $x_{\mathrm{TD}1}(t) = 5$ V, $x_{\mathrm{TD}2}(t) = 5$ V,

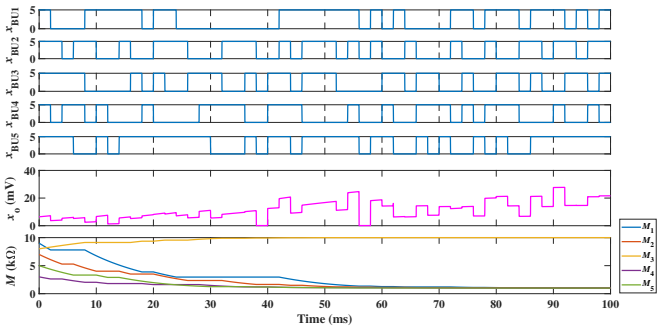


Fig. 13. The TDSSAA learning of the SSMSNC with five MSs.

$x_{\mathrm{TD3}}(t) = 0$ V, $x_{\mathrm{TD4}}(t) = 5$ V, $x_{\mathrm{TD5}}(t) = 5$ V. Meanwhile, the patterns of $x_{\mathrm{BU1}}(t)$, $x_{\mathrm{BU2}}(t)$, $x_{\mathrm{BU3}}(t)$, $x_{\mathrm{BU4}}(t)$, $x_{\mathrm{BU5}}(t)$ are set randomly. Synaptic weights are modified during learning periods and output $x_{\mathrm{o}}(t)$ of the SSMSNC. The simulation results show that M_3 gradually increases and reaches the maximum value, while M_1, M_2, M_4 and M_5 gradually decrease and reach the minimum value. Then, when the pattern of the BU attention spike signals is the same as the TD attention spike signals at 90 ms, the output voltage of $x_{\mathrm{o}}(t)$ is the maximum.

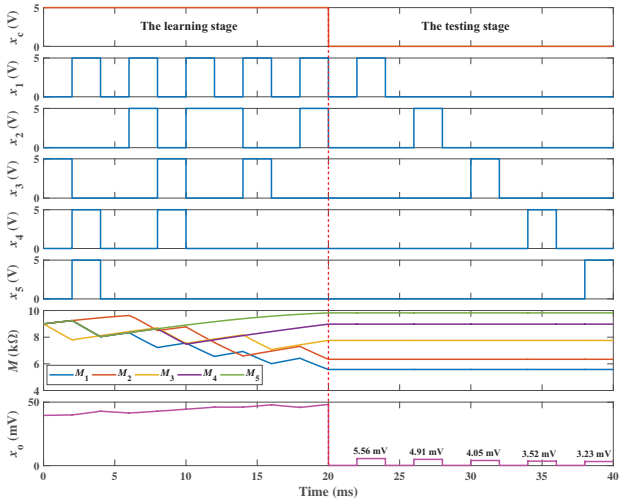


Fig. 14. The BUSSAA learning and testing of the SSMSNC with five MSs. In the learning stage, five MSs respectively learn under decreasing numbers of spikes and the memristances of the five memristors are modified. In the testing stage, $x_1(t), x_2(t), x_3(t), x_4(t), x_5(t)$ have a spike at 22 ms, 26 ms, 30 ms, 34 ms and 38 ms, respectively.

The simulation results of the SSMSNC for the BUSSAA learning and testing are shown in Fig. 14, where the number of the MS is set to $n = 5$, R_{f} is set to 1 $k\Omega$ and the initial memristances of all memristors are set to 9 $k\Omega$. In the learning stage, spikes with decreasing numbers occur in $x_1(t), x_2(t), x_3(t), x_4(t), x_5(t)$, respectively. The simulation results show that the weight of the MS_1 is the largest after the BUSSAA learning. In the testing stage, the voltage of $x_{\mathrm{o}}(t)$ for the MS_1 is the largest due to the largest synaptic weight. Therefore, the voltage amplitude of $x_{\mathrm{o}}(t)$ in the testing stage can be used as the basis for classification.

IV. Attention Encoding Circuit for the SSMSNC

Recently, some efficient spike encoding methods have been proposed, such as rate encoding method [51], hybrid spike encoding method [52] and direct input encoding method [46]. Among these, the rate coding method requires a large number of time steps to obtain high accuracy. The mapping between image pixel intensities and individual neuron firing times in the hybrid spike encoding method lead to inconvenience for circuit implementation. The direct input encoding method feeds the analog pixel values directly into the first layer of the SNN, but the first layer in direct coded SNNs requires multiply-and-accumulates (MAC), which results in an expensive cost. Although an SRAM-based processing-in-memory (PIM) architecture was proposed in [5] to alleviate the cost of the first layer in direct coded SNNs, the software-hardware co-design method is less simple than the circuit-based encoding method. Therefore, in this section we propose a BU attention encoding circuit and a TD attention encoding circuit for the SSMSNC.

A. BU Attention Encoding Circuit

BU attention is the key for the SSMSNC to capturing external information. The realization of the TDSSAA and BUSSAA in the SSMSNC is premised on the BU attention spike signal. Therefore, a BU attention encoding circuit (BUAEC) is proposed in this section.

The size of the captured image is set to $N \times N$, and the BU attention field is set to $m \times m$. The image is divided into $(\frac{N}{m})^2$ sub-blocks by the BU attention field. The gray value of the k pixel in the jth sub-block can be converted into a voltage value as follows:

$$V_{\mathrm{p}j\_k} = \gamma_j (p_{j\_k} - p_{\mathrm{Im}}), \tag{11}$$

where $V_{\mathrm{p}j\_k}$ is the converted voltage of the kth pixel in the jth sub-block, $k = 1, 2, \ldots, m^2$ and $j = 1, 2, \ldots, (\frac{N}{m})^2$. p_{j_k} is the gray value of the kth pixel in the jth sub-block, and $p_k \in [0, 255]$. p_{Im} is the mean gray value of the image. γ_j is the conversion factor of the jth sub-block. Therefore, m^2 gray values of pixels in each sub-block can be converted into the corresponding voltage values. The difference operation between p_{j_k} and p_{Im} can eliminate the effect of the mean gray of the image and make the BU attention only focus on the dominant features in the field of view. It should be noted that the size of the BU attention field can be adjusted with the size of the image, and it is necessary to ensure that the size of the image is an integer multiple of the size of the BU attention field.

Fig. 15(a) shows a handwritten digit image of size 28 × 28, which is obtained from the MNIST dataset [53]. The image is divided into 16 sub-blocks by the 4 × 4 BU attention field, and the size of each sub-block is 7 × 7 with 49 pixels. As shown in Fig. 15(b), the gray values of pixels in the jth sub-block are converted into voltages with $\gamma_j = 0.0001$ as the voltage amplitude of the input signals $v_{\mathrm{p1}}(t), v_{\mathrm{p2}}(t), \ldots, v_{\mathrm{p49}}(t)$ and transmitted to the jth BUAEC. The spike duration in input signals is set to 1 ms. The OPAMP $\mathrm{U}_{\mathrm{b}j}$ and the resisters $\mathrm{R}_{\mathrm{b}j}$, $\mathrm{R}_{\mathrm{b}j\_1}$, $\mathrm{R}_{\mathrm{b}j\_2}$, …, $\mathrm{R}_{\mathrm{b}j\_49}$ form an inverting summation OPAMP circuit. Therefore, the gray values of all pixels in

the jth sub-block can be synchronously converted into a corresponding sub-block voltage signal. The output signal $v_{\mathrm{b}j}(t)$ of the jth sub-block can be expressed as:

$$v_{\mathrm{b}j}(t) = -\sum_{k1=1}^{49} \frac{R_{\mathrm{b}j}}{R_{\mathrm{b}j\_k1}} v_{\mathrm{p}j\_k1}(t). \tag{12}$$

Then $v_{\mathrm{b}j}(t)$ is compared in the comparison OPAMP circuit consisting of the OPAMP $\mathrm{U}_{\mathrm{a}j}$ and the resisters $\mathrm{R}_{\mathrm{a}j\_1}$ and $\mathrm{R}_{\mathrm{a}j\_2}$. If the voltage of $v_{\mathrm{b}j}(t)$ exceeds 0 V, a spike is generated in $x_{\mathrm{BU}j}(t)$, otherwise no BU attention spike occurs. After all sub-blocks of the image complete the voltage conversion synchronously, $(\frac{N}{m})^2$ converted sub-block voltages are obtained and then input into $(\frac{N}{m})^2$ MSs of the SSMSNC in parallel. The number of MSs in the SSMSNC is equal to the number of sub-blocks.

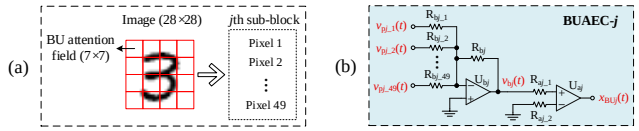


Fig. 15. (a) The handwritten digit image divided into 16 sub-blocks by the BU attention field, and each sub-block containing 49 pixels. (b) The circuit structure of the jth BUAEC.

Fig. 16 shows the simulation results of $v_{\mathrm{b}j}(t)$ and $x_{\mathrm{BU}j}(t)$ after encoding the handwritten digit image shown in Fig. 15(a), where $x_{\mathrm{BU3}}(t)$ $x_{\mathrm{BU6}}(t)$, $x_{\mathrm{BU7}}(t)$, $x_{\mathrm{BU10}}(t)$, $x_{\mathrm{BU11}}(t)$, $x_{\mathrm{BU14}}(t)$, $x_{\mathrm{BU15}}(t)$ generate a BU attention spike. In the simulation, all resistors in BUAECs are set to 1 $k\Omega$. The positive supply of $\mathrm{U}_{\mathrm{b}j}$ and $\mathrm{U}_{\mathrm{a}j}$ is +5 V, the negative supply of $\mathrm{U}_{\mathrm{b}j}$ is -5 V, and the negative supply of $\mathrm{U}_{\mathrm{a}j}$ is electrical grounding, where $j = 1, 2, \ldots, 16$.

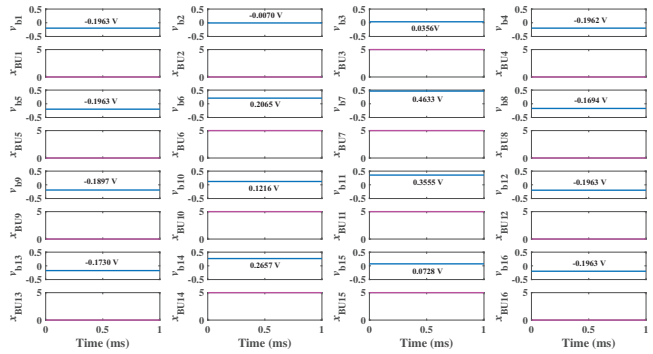


Fig. 16. The simulation results of 16 BUEACs encoding the handwritten digit image in Fig. 15(a).

B. TD Attention Encoding Circuit

Assume a total of r images of size $N \times N$ are processed by BUAECs with the BU attention field of size $m \times m$. Each image obtains $(\frac{N}{m})^2$ sub-block signals, which is the same as the number of n MSs in SSMSNCs. As shown in Fig. 17(a), $v_{\mathrm{b}j}(t)$ of each image are corresponding input into the jth TD attention encoding circuit (TDAEC-j) for encoding to obtain the TD attention spike signals, where $j = 1, 2, \ldots, n$. The circuit structure of TDAEC-j is introduced in Fig. 17(b). The OPAMP $\mathrm{U}_{\mathrm{c}j}$ and the resisters $\mathrm{R}_{\mathrm{c}j}$, $\mathrm{R}_{\mathrm{c}j\_1}$, $\mathrm{R}_{\mathrm{c}j\_2}$, …, $\mathrm{R}_{\mathrm{c}j\_r}$ form an inverting summation OPAMP circuit. The OPAMP $\mathrm{U}_{\mathrm{d}j}$ and the resisters $\mathrm{R}_{\mathrm{d}j\_1}$, $\mathrm{R}_{\mathrm{d}j\_2}$ form an inverting OPAMP circuit. The jth sub-block signals of all images are simultaneously transmitted to the TDAEC-j for superposition, and the superimposed signal $v_{\mathrm{d}j}(t)$ can be described as follows:

$$v_{\mathrm{d}j}(t) = \sum_{k2=1}^{r} \frac{R_{\mathrm{c}j} R_{\mathrm{d}j\_2}}{R_{\mathrm{c}j\_k2} R_{\mathrm{d}j\_1}} v_{\mathrm{b}j\_k2}(t). \tag{13}$$

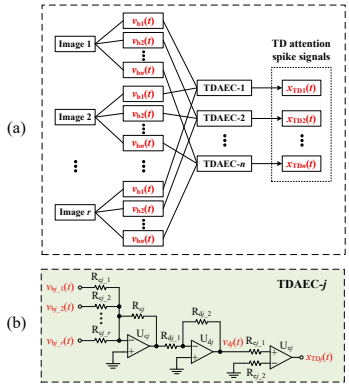


Fig. 17. (a) The schematic diagram of n TDAECs encoding r images to obtain n TD attention spike signals. (b) The circuit structure of the jth TDAEC.

In the simulation, all resistors in TDAECs are set to 1 $k\Omega$. The OPAMP $\mathrm{U}_{\mathrm{e}j}$ and the resisters $\mathrm{R}_{\mathrm{e}j\_1}$ and $\mathrm{R}_{\mathrm{e}j\_2}$ are composed of a comparison OPAMP circuit. If the voltage of $v_{\mathrm{e}j}(t)$ exceeds 0 V, a spike is generated in $x_{\mathrm{TD}j}(t)$, otherwise no TD attention spike occurs. The positive supply of $\mathrm{U}_{\mathrm{c}j}$, $\mathrm{U}_{\mathrm{d}j}$ and $\mathrm{U}_{\mathrm{e}j}$ is +5 V, the negative supply of $\mathrm{U}_{\mathrm{c}j}$ and $\mathrm{U}_{\mathrm{d}j}$ is -5 V, and the negative supply of $\mathrm{U}_{\mathrm{e}j}$ is electrical grounding.

V. Application of Image Classification

In this section, we construct a four-layer SNN circuit consisting of 20 SSMSNCs for supervised learning and classification on the MNIST dataset [53] and the Fashion-MNIST dataset [54]. The four-layer of SSMSNNC is divided as: input layer, TDSSA layer, BUSSA layer and output layer. The structure of the SSMSNNC is shown in Fig. 18. The size of sample images is set to 30 × 30, the size of the BU attention field is set to 6 × 6, thereby a SSMSNC contains 25 MSs.

The input layer is divided into the TDSSA input layer and the BUSSA input layer. The TDSSA input layer consists of 25 BUAECs to encode a part of sample images and transmits the encoded BU attention spike signals into the SSMSNCs in the TDSSA layer for TD selection. The TDSSA layer consists of 10 SSMSNCs with TD attention spike signals. The TD attention spikes resulted from TDAECs encoding of a small scale labeled class images. Meanwhile, $x_{\mathrm{TDSSAA}}(t)$ of the SSMSNCs are always maintained at 5 V in the TDSSA layer. The BUSSA input layer consists of 25 BUAECs to encode all unlabeled images and transmits the encoded BU attention spikes into SSMSNCs of the BUSSA layer for classification. The BUSSA layer consists of 10 SSMSNCs without TD attention spike signals. The training data of the SSMSNCs in the BUSSA layer are the BU attention spike signals of selected images from the TDSSA layer. In the BUSSA layer,

$x_{\mathrm{TDSSAA}}(t)$ of SSMSNCs are always maintained at 0 V. In the learning stage of the TDSSA layer and the BUSSA layer, $x_{\mathrm{c}}(t)$ of SSMSNCs are kept at 5 V. After the training of SSMSNCs in the BUSSA layer is completed, the BUSSA layer performs classification on all unlabeled images in the testing stage with $x_{\mathrm{c}}(t) = 0$ V. Then the classification results are analyzed in the output layer.

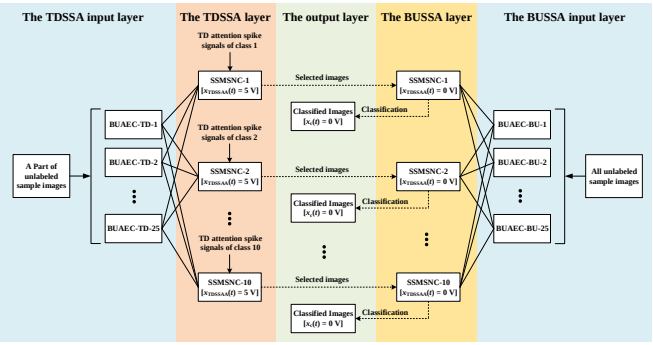


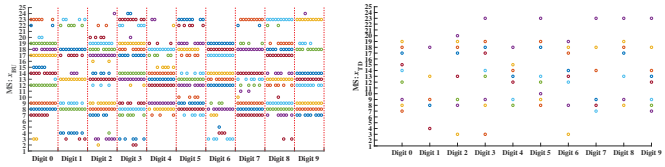
Fig. 18. The structure of the SSMSNNC.

In the simulations of the SSMSNNC, R_{f} of each SSMSNC is set to 100 Ω. The initial memristance of all memristors is set to 9 $k\Omega$. The resistances of all the resistors in BUAECs and TDAECs are set to 1 $k\Omega$. The other parameter settings are the same as Section III and IV. In the simulation of image learning, 10000 sample images are obtained from the MNIST training set and the Fashion-MNIST training set, respectively. In the simulation of image testing, 10000 images are obtained from the MNIST test set and the Fashion-MNIST test set, respectively.

A. The TDSSA layer and Image Selection

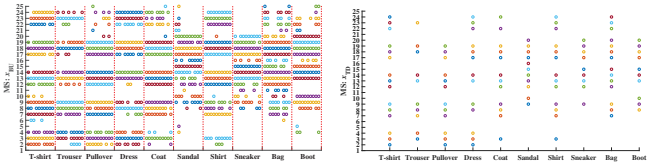
In the simulations of the TDSSA layer, the number of unlabeled images used for selection is set to 2000. Ten sample images of each digit in the MNIST training set are labeled as the labeled images, which are respectively encoded as the TD attention spike signals of each SSMSNC in the TDSSA layer. Each SSMSNC in the TDSSA layer corresponds to a class. The BU attention signals of the labeled image are shown in Fig. 19(a). The TD attention spikes encoded by 25 TDAECs on the labeled images are shown in Fig. 19(b). Similarly, the BU attention signals of the labeled images in the Fashion-MNIST training set are obtained as shown in Fig. 20(a). The TD attention spikes encoded by 25 TDAECs on the labeled images are shown in Fig. 20(b). Then 2000 unlabeled images of the MNIST training set and 2000 unlabeled images of the Fashion-MNIST training set are encoded by the BUAECs and input synchronously into the 10 SSMSNCs of the TDSSA layer for TD selection, respectively.

The learning time for each image is set to 1 ms in a learning epoch, and 2000 unlabeled images are learned by each SSMSNC in the TDSSA layer during 100 s for 50 learning epochs. The final amplitude of the output signal $x_{\mathrm{o}}(t)$ after learning for each image is used to compare with the neuron threshold of the corresponding SSMSNC. The number of spikes generated in the output spike signal $s_{\mathrm{o}}(t)$



(a) (b)

Fig. 19. (a) The BU attention spikes of the 10 labeled images for each digit of the MNIST dataset. The circles represent attention spikes. The ordinate represents 25 MSs of the SSMSNC for receiving the corresponding BU attention spike signals. (b) The schematic diagram of the TD attention spikes signals in the TDSSA layer. The abscissa represents digits from 0 to 9.



(a) (b)

Fig. 20. (a) The BU attention spikes of the 10 labeled images for each class of the Fashion-MNIST dataset. (b) The schematic diagram of the TD attention spikes signals in the TDSSA layer. Each class is described on the abscissa.

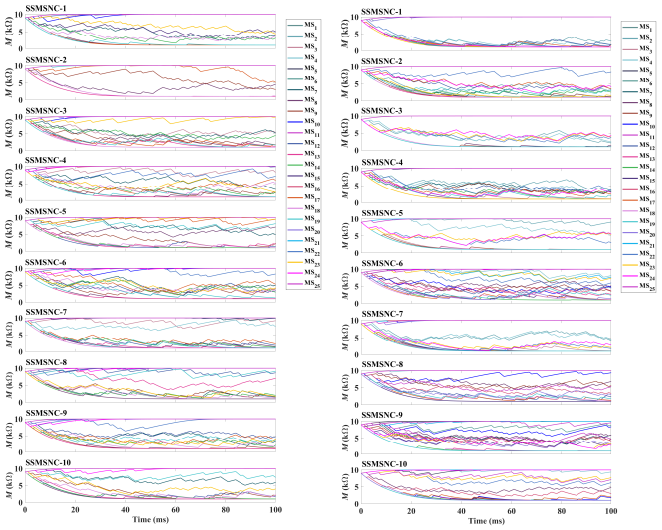
of each SSMSNC in the TDSSA layer can be adjusted by changing the neuron threshold of the corresponding SSMSNC. In the simulation, we set the number of selected images for each SSMSNC of the BUSSA layer to 100. By continuously adjusting the neuron threshold voltage of the SSMSNC, when the number of output spikes of the SSMSNC reaches 100, the corresponding unlabeled image is selected based on the spike position in $s_{\mathrm{o}}(t)$. Then the BU attention spike signals of the 100 selected images are used as the training data for the corresponding SSMSNC in the BUSSA layer.

B. The BUSSA layer and Image Classification

After each SSMSNC in the TDSSA layer selects 100 images from 2000 unlabeled images, the BU attention spike signals of the selected images are input into the corresponding SSMSNC in the BUSSA layer for training, and the training time for each image is 1 ms. Fig. 21(a) and (b) show the trained memristances of 25 MSs in each SSMSNC of the BUSSA layer when learning the selected images of the MNIST dataset and the Fashion-MNIST dataset, respectively. Fig. 22 and Fig. 23 show the output voltages of 10 SSMSNCs in the BUSSA layer obtained by classifying a set of test images during the testing stage. The simulation results show that when the test images are input, the output voltage amplitude of the corresponding class of SSMSNC in BUSSA layer is the highest, which indicates the SSMSNNC successfully classified the test images.

C. Classification Performance

To study the noise robustness of the SSMSNNC, we add different levels of Gaussian noise and salt-pepper noise to 10000 test sample images of the MNIST dataset and the Fashion-MNIST dataset, respectively. The intensity of Gaussian noise is represented by the variance σ^2. The mean of Gaussian noise is set to 0. The intensity of salt-pepper noise is represented



(a) (b)

Fig. 21. The memristive modification process of the 25 MSs in each SSMSNC of the BUSSA layer. (a) The training of images selected from the MNIST dataset. (b) The training of images selected from the Fashion-MNIST dataset.

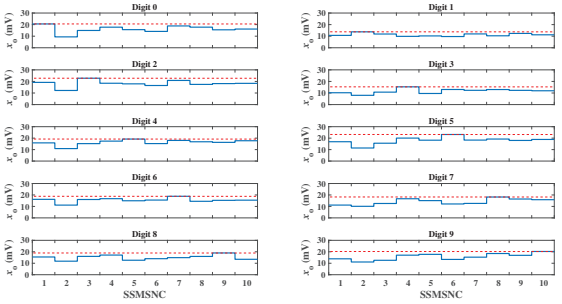


Fig. 22. The output voltage $x_{\mathrm{o}}(t)$ of 10 SSMSNCs in the BUSSA layer tested on a set of digit images from 0 to 9 in the MNIST test set.

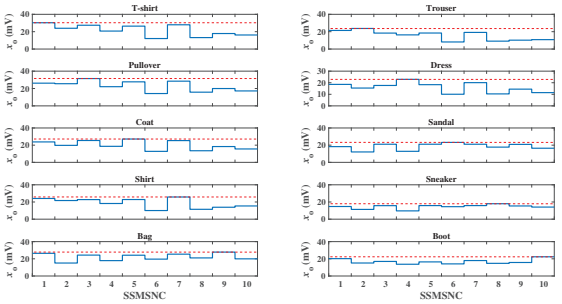


Fig. 23. The output voltage $x_{\mathrm{o}}(t)$ of 10 SSMSNCs in the BUSSA layer tested on a set of different classes of test images in the Fashion-MNIST test set.

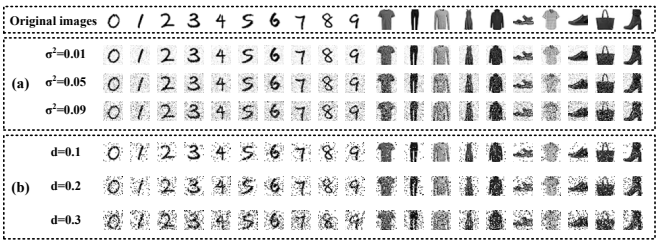


Fig. 24. The example of test images of the MNIST dataset and the Fashion-MNIST dataset with different noise. (a) The images with different variances of Gaussian noise. (b) The images with different densities of salt-pepper noise.

by the density d. The higher the classification accuracy of the algorithm on noisy images in the testing stage, the better the robustness of the algorithm.

Fig. 24(a) shows the images of the MNIST dataset and the Fashion-MNIST dataset with different variances of Gaussian noise. The gray value of the noise pixel is between 0 and 255. The larger the variance σ^2, the more noise points. The salt-pepper noise randomly changes some pixel values of the image to 0 or 255 as shown in Fig. 24(b). The larger the density d, the more pixels are changed. In the simulations, the size of sample images is set to 30 $\times$ 30, the size of the BU attention field is set to 3 $\times$ 3.

TABLE II
IMAGE CLASSIFICATION PERFORMANCE OF THE SSMSNNC UNDER DIFFERENT NOISE LEVELS

	Noise level	MNIST testing accuracy	Fashion-MNIST testing accuracy
No noise	-	98.1%	86.4%
Gaussian noise	$\sigma^2 = 0.01$	92.6%	76.1%
	$\sigma^2 = 0.05$	83.1%	63.7%
	$\sigma^2 = 0.09$	67.6%	51.5%
Salt-pepper noise	$d = 0.1$	91.8%	75.2%
	$d = 0.2$	72.5%	58.6%
	$d = 0.3$	49.3%	45.2%

In the classification of no noise images, the MNIST testing accuracy is 98.1% and the Fashion-MNIST testing accuracy is 86.4%. When the intensity of the Gaussian noise is small, the noise effect is eliminated in the BU attention encoding process in our work. When the variance σ^2 is large enough, the classification accuracy of the SSMSNNC is greatly affected, as shown in Table II. The salt-pepper noise is to convert the original pixel value of the test image to black (0) or white (255). In the process of the BU attention encoding, the conversion of pixel to black results in easier generation of BU attention spikes, and the conversion of pixel to white results in harder generation of BU attention spikes. The increase or decrease of BU attention spikes can affect the classification accuracy of the SSMSNNC. The classification accuracy in our work can be maintained at a high level when the salt-pepper noise intensity is weak. When the salt-pepper noise is dense enough, the classification accuracy of the SSMSNNC decreases greatly.

VI. Discussion

A. Power Consumption

The power consumption of the SSMSNNC is derived from SSMSNCs, BUAECs and TDAECs, where the SSMSNC is mainly composed of the MSs, the generating circuits of $x_i(t)$, the output circuit and the power supply circuit. The power consumption in Table III is obtained by PSPICE simulation reports of the SSMSNNC in Section V. During the learning stage, power consumption occurs in BUAECs and TDAECs when an image is input. Power consumption of the SSMSNCs in the TDSSA layer and the BUSSA layer occurs when attention spikes are input. In the testing stage, only the SSMSNCs of the BUSSA layer and the BUAECs of the BUSSA input layer work.

TABLE III
POWER CONSUMPTION OF SUBCIRCUITS IN THE SSMSNNC

	Circuit component	Power consumption
Per SSMSNC	Per MS circuit	0.15 mW
	Per generating circuit of $x_i(t)$	0.20 mW
	Output circuit	0.35 mW
	Power supply circuit	0.05 mW
Per BUAEC	36 inputs	0.59 mW
Per TDAEC	25 inputs	0.60 mW

B. Impacts of Parameters on Power Consumption and Classification Performance

Different sizes of BU attention fields result in different numbers of sub-blocks in images and MSs in the SSMSNC. Different numbers of sub-blocks affect the classification accuracy, and different numbers of MSs affect the power consumption of the SSMSNC. As shown in Table IV, the size of images is set to 30×30, the size of the BU attention field is set to 3×3, 5×5 and 10×10, respectively. Thus the number of sub-blocks in a image is 100, 36 and 9, respectively. The number of MSs in the SSMSNC is the same as the numbers of sub-blocks. In a SSMSNNC, the numbers of MSs of the SSMSNC in the TDSSA layer and the BUSSA layer are the same as the numbers of BUAECs in the TDSSA input layer and the BUSSA input layer, respectively.

TABLE IV
IMPACT OF PARAMETERS ON POWER CONSUMPTION AND CLASSIFICATION PERFORMANCE

$N \times N$	30×30		
$m \times m$	3×3	5×5	10×10
Number of sub-blocks in an image	100	36	9
Number of MSs in a SSMSNC	100	36	9
Number of BUAECs in a SSMSNNC	200	72	18
Power consumption of a SSMSNNC with 20 SSMSNC	771.4 mW	309.6 mW	94.2 mW
MNIST testing accuracy	98.1%	93.2%	65.6%
Fashion-MNIST testing accuracy	86.4%	80.9%	52.3%

The decrease in the size of the BU attention field leads to the increase in the number of MSs. The greater the number of MSs, the higher the power consumption of the SSMSNC, and the larger the number and power consumption of BUAECs. Thereby the total power consumption decreases as the size of the BU attention field increases and the number of MSs decreases. However, the increase in the size of the BU attention field and the decrease in the number of MSs lead to the decrease of classification accuracy. Therefore, the classification of a SSMSNNC with appropriate size of the BU attention field and number of MSs to achieve optimal performance needs to be considered.

C. Comparison with Other Memristive SNN Circuits

The comparison of our work with other memristive SNN circuits is shown in Table V. The memristive SNN circuit designed in this work adopts the proposed SSAA based on the selective attention mechanism, which is a selectively supervised learning method. Compared to other memristive SNN circuits with supervised learning methods, this work only requires labeled images that account for less than 5% of the training sample images to classify the MNIST dataset and the Fashion-MNIST dataset with superior classification accuracy, which greatly reduces the cost required for sample labeling and improves the efficiency of supervised learning. Compared to other memristive SNN circuits with unsupervised learning methods, although this work requires a small scale of labeled images, it is better in classification performance.

D. Non-Ideality Analysis

Considering that the non-ideality (such as device variation) in the memristive devices will affect the classification accuracy of the SSMSNNC, the non-ideality in the memristive devices is modeled as a Gaussian variation in the memristance with $\frac{\sigma}{\mu}$ [55], [56]. The memristive devices with different $\frac{\sigma}{\mu}$ are used for training and testing of image classification in the SSMSNNC, and the classification results are shown in Table VI. In the simulation, the size of the BU attention field is set to 3×3. The simulation results show that the classification accuracy of the SSMSNNC on both the MNIST dataset and Fashion-MNIST dataset gradually decreases with the increase of $\frac{\sigma}{\mu}$, which means that the more significant the non-ideality in the memristive devices, the worse the classification performance.

TABLE VI
IMPACT OF MEMRISTORS WITH DIFFERENT NON-IDEALITIES ON CLASSIFICATION ACCURACY

$\frac{\sigma}{\mu}$	MNIST testing accuracy	Fashion-MNIST testing accuracy
10%	84.3%	71.7%
20%	61.5%	42.6%
30%	36.2%	27.5%

VII. Conclusion

In this paper, we report a selective supervised attention algorithm for SNs inspired by the selective attention mechanism of biological vision principles, and design a selective supervised spiking neuron circuit based on memristors, which has the advantages of algorithm-based rules and physical realizability. Algorithm performance analysis shows that the proposed algorithm has better learning performance compared with other

TABLE V
COMPARISON WITH OTHER MEMRISTIVE SNN CIRCUITS

Works	Duan et.al [37]	Chen et.al [36]	Zhao et.al [30]	Zhou et.al [38]	This work
Learning method	Supervised	Supervised	Unsupervised	Supervised	Selective supervised
Biological mechanism	Spatiotemporal dynamics and gain modulation	Synaptic complementation	Lateral inhibition and homeostasis	-	Selective attention
Learning algorithm	Simplified δ-rule	ReSuMe	STDP	BPTT	SSAA
Percentage of labeled samples in the training set	100%	100%	0%	100%	$<$5%
Maximum MNIST testing accuracy	83.2%	94.0%	91.7%	97.5%	98.1%
Maximum Fashion-MNIST testing accuracy	-	-	-	75.2%	86.4%

supervised algorithms. Meanwhile, we design selective attention encoding circuits to provide the hardware foundation for the image processing of the designed SN circuit. Furthermore, a selectively supervised memristive SNN circuit is designed. The simulation results show that the proposed memristive SNN circuit can perform accurate classification on the MNIST dataset and the Fashion-MNIST dataset after learning a small number of labeled samples and has excellent classification accuracy as well as robustness. Compared with other memristive SNN circuits, this work greatly reduces the time and cost of supervised learning and improves the real-time working ability of the SNN hardware in practical applications. In the future, we will further optimize the performance and design methods of the memristive SNN circuit to achieve higher robustness and higher classification accuracy.

REFERENCES

[1] A. Taherkhani, A. Belatreche, Y. Li, G. Cosma, L. P. Maguire, and T. McGinnity, "A review of learning in biologically plausible spiking neural networks," *Neural Netw.*, vol. 122, pp. 253–272, 2020.

[2] S. Dora and N. K. Kasabov, "Spiking neural networks for computational intelligence: An overview," *Big Data Cogn. Comput.*, vol. 5, no. 4, p. 67, 2021.

[3] Y. Kim and P. Panda, "Revisiting batch normalization for training low-latency deep spiking neural networks from scratch," *Front. Neurosci.*, arXiv preprint arXiv:2010.01729, 2021.

[4] Y. Kim and P. Panda, "Optimizing deeper spiking neural networks for dynamic vision sensing," *Neural Netw.*, vol. 144, pp. 686–698, 2021.

[5] G. Datta, S. Kundu, A. Jaiswal, and P. Beerel, "ACE-SNN: Algorithm-hardware co-design of energy-efficient & low-latency deep spiking neural networks for 3D image recognition," *Front. Neurosci.*, DOI: 10.3389/fnins.2022.815258, 2022.

[6] Y. Kim and P. Panda, "Visual explanations from spiking neural networks using inter-spike intervals," *Sci Rep*, vol. 11, no. 1, pp. 1-14, 2021.

[7] P. Huynh, M. Varshika, A. Paul, M. Işık, A. Balaji, and A. Das, "Implementing spiking neural networks on neuromorphic architectures: A review," arXiv preprint arXiv:2202.08897, 2022.

[8] E. A. Miranda and J. Suñé, "Memristors for neuromorphic circuits and artificial intelligence applications," *Materials*, vol. 13, no. 4, p. 938, 2020.

[9] L. Chua, "Memristor-the missing circuit element," *IEEE Transactions on Circuit Theory*, vol. 18, pp. 507–519, 1971.

[10] W. Cai, F. Ellinger, and R. Tetzlaff, "Neuronal synapse as a memristor: Modeling pair- and triplet-based STDP rule," *IEEE Trans. Biomed. Circuits Syst.*, vol. 9, no. 1, pp. 87–95, 2015.

[11] D. Strukov, G. Snider, D. Stewart, and S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, 2008.

[12] C. Yakopcic, T. M. Taha, G. Subramanyam, and R. E. Pino, "Generalized memristive device spice model and its application in circuit design," *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.*, vol. 32, no. 8, pp. 1201–1214, 2013.

[13] X. Wang, B. Xu, and L. Chen, "Efficient memristor model implementation for simulation and application," *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.*, vol. 36, no. 7, pp. 1226–1230, 2017.

[14] Y. Zhang, X. Wang, Y. Li, and E. G. Friedman, "Memristive model for synaptic circuits," *IEEE Trans. Circuits Syst. II-Express Briefs*, vol. 64, no. 7, pp. 767–771, 2016.

[15] H. Lin, C. Wang, Q. Hong, and Y. Sun, "A multi-stable memristor and its application in a neural network," *IEEE Trans. Circuits Syst. II-Express Briefs*, vol. 67, no. 12, pp. 3472–3476, 2020.

[16] W. Xie, C. Wang, and H. Lin, "A fractional-order multistable locally active memristor and its chaotic system with transient transition, state jump," *Nonlinear Dyn.*, vol. 104, no. 4, pp. 4523–4541, 2021.

[17] A. Ankit, A. Sengupta, and K. Roy, "Trannsformer: Neural network transformation for memristive crossbar based neuromorphic system design," in *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2017, pp. 533–540.

[18] H. Lin, C. Wang, C. Xu, X. Zhang, and H. H. C. Iu, "A memristive synapse control method to generate diversified multi-structure chaotic attractors," *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.*, DOI: 10.1109/TCAD.2022.3186516, 2022.

[19] C. Xu, C. Wang, J. Jiang, J. Sun, and H. Lin, "Memristive circuit implementation of context-dependent emotional learning network and its application in multi-task," *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.*, vol. 41, no. 9, pp. 3052–3065, 2022.

[20] H. Lin, C. Wang, C. Chen, Y. Sun, C. Zhou, C. Xu, and Q. Hong, "Neural bursting and synchronization emulated by neural networks and circuits," *IEEE Trans. Circuits Syst. I-Regul. Pap.*, vol. 68, no. 8, pp. 3397–3410, 2021.

[21] C. Xu, C. Wang, Y. Sun, Q. Hong, Q. Deng, and H. Chen, "Memristor-based neural network circuit with weighted sum simultaneous perturbation training and its applications," *Neurocomputing*, vol. 462, pp. 581–590, 2021.

[22] H. Lin, C. Wang, L. Cui, Y. Sun, C. Xu, and F. Yu., "Brain-like initial-boosted hyperchaos and application in biomedical image encryption," *IEEE Trans. Ind. Inform.*, vol. 18, no. 12, pp. 8839–8850, 2022.

[23] R. Yan, Q. Hong, C. Wang, J. Sun, and Y. Li, "Multilayer memristive neural network circuit based on online learning for license plate detection," *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.*, vol. 41, no. 9, pp. 3000–3011, 2022.

[24] C. Pan, Q. Hong, and X. Wang, "A novel memristive chaotic neuron circuit and its application in chaotic neural networks for associative memory," *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.*, vol. 40, no. 3, pp. 521–532, 2021.

[25] H. Lin, C. Wang, Y. Sun, and T. Wang, "Generating n-scroll chaotic attractors from a memristor-based magnetized hopfield neural network," *IEEE Trans. Circuits Syst. II-Express Briefs*, DOI: 10.1109/TCSII.2022.3212394, 2022.

[26] H. An, M. S. Al-Mamun, M. K. Orlowski, L. Liu, and Y. Yi, "Three-dimensional neuromorphic computing system with two-layer and low-variation memristive synapses," *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.*, vol. 41, no. 3, pp. 400–409, 2022.

[27] M. Hu, Y. Chen, J. J. Yang, Y. Wang, and H. H. Li, "A compact memristor-based dynamic synapse for spiking neural networks," *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.*, vol. 36, no. 8, pp. 1353–1366, 2017.

[28] A. Ankit, A. Sengupta, P. Panda, and K. Roy, "RESPARC: A reconfigurable and energy-efficient architecture with memristive crossbars for deep spiking neural networks," in *Proceedings of the 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2017, pp. 1–6.

[29] A. Shukla and U. Ganguly, "An on-chip trainable and the clock-less spiking neural network with 1r memristive synapses," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 4, pp. 884–893, 2018.

[30] Z. Zhao, L. Qu, L. Wang, Q. Deng, N. Li, Z. Kang, S. Guo, and W. Xu, "A memristor-based spiking neural network with high scalability and learning efficiency," *IEEE Trans. Circuits Syst. II-Express Briefs*, vol. 67, no. 5, pp. 931–935, 2020.

[31] X. Wang, X. Lin, and X. Dang, "Supervised learning in spiking neural networks: A review of algorithms and evaluations," *Neural Netw.*, vol. 125, pp. 258–280, 2020.

[32] F. Ponulak and A. Kasiński, "Supervised learning in spiking neural networks with RESUME: sequence learning, classification, and spike shifting," *Neural Comput.*, vol. 22, no. 2, pp. 467–510, 2010.

[33] Y. Xu, X. Zeng, and S. Zhong, "A new supervised learning algorithm for spiking neurons," *Neural Comput.*, vol. 25, no. 6, pp. 1472–1511, 2013.

[34] X. Lin, X. Wang, and X. Dang, "A new supervised learning algorithm for spiking neurons based on spike train kernels," *Acta Electronica Sinica*, vol. 44, no. 12, pp. 2877–2886, 2016.

[35] Y. Nishitani, Y. Kaneko, and M. Ueda, "Supervised learning using spike-timing-dependent plasticity of memristive synapses," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 2999–3008, 2015.

[36] Y. Chen, Y. Zhou, F. Zhuge, B. Tian, M. Yan, Y. Li, Y. He, and X. S. Miao, "Graphene–ferroelectric transistors as complementary synapses for supervised learning in spiking neural network," *npj 2D Mater. Appl.*, vol. 3, no. 1, pp. 1–9, 2019.

[37] Q. Duan, Z. Jing, X. Zou, Y. Wang, K. Yang, T. Zhang, S. Wu, R. Huang, and Y. Yang, "Spiking neurons with spatiotemporal dynamics and gain modulation for monolithically integrated memristive neural networks," *Nat. Commun.*, vol. 11, no. 1, p. 3399, 2020.

[38] P. Zhou, J. K. Eshraghian, D. Choi, and S. S. Kang, "Spiceprop: Backpropagating errors through memristive spiking neural networks," *arXiv preprint arXiv:2203.01426*, 2022.

[39] F. Katsuki and C. Constantinidis, "Bottom-up and top-down attention: different processes and overlapping neural systems," *Neuroscientist*, vol. 20, no. 5, pp. 509–521, 2014.

[40] A. R. and A. J, "Study on computational visual attention system and its contribution to robotic cognition system," in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)*, 2020, pp. 278–283.
[41] A. Gazzaley and A. C. Nobre, "Top-down modulation: bridging selective attention and working memory," *TRENDS COGN. SCI.*, vol. 16, no. 2, pp. 129–135, 2012.
[42] C. Yang, X. Wang, and Z. Zeng, "Full-circuit implementation of transformer network based on memristor," *IEEE Trans. Circuits Syst. I-Regul. Pap.*, vol. 69, no. 4, pp. 1395–1407, 2022.
[43] B. R. Burnham, M. Sabia, and C. Langan, "Components of working memory and visual selective attention," *J. Exp. Psychol.-Hum. Percept. Perform.*, vol. 40, no. 1, pp. 391–403, 2014.
[44] S. Schreiber, J. Fellous, D. Whitmer, P. Tiesinga, and T. J. Sejnowski, "A new correlation-based measure of spike timing reliability," *Neurocomputing*, vol. 52, pp. 925–931, 2003.
[45] N. Rathi, G. Srinivasan, P. Panda, and K. Roy, "Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation," *arXiv preprint arXiv:2005.01807*, 2020.
[46] N. Rathi and K. Roy, "DIET-SNN: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, DOI: 10.1109/TNNLS.2021.3111897, 2021.
[47] G. Datta and P. A. Beerel, "Can deep neural networks be converted to ultra low-latency spiking neural networks?" in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2022, pp. 718–723.
[48] D. Biolek, V. Biolková, and Z. Biolek, "SPICE model of memristor with nonlinear dopant drift," *Radioengineering*, vol. 18, no. 2, pp. 210–214, 2009.
[49] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "TEAM: Threshold adaptive memristor model," *IEEE Trans. Circuits Syst. I-Regul. Pap.*, vol. 60, no. 1, pp. 211–221, 2013.
[50] S. Kvatinsky, M. Ramadan, E. G. Friedman, and A. Kolodny, "VTEAM: A general model for voltage-controlled memristors," *IEEE Trans. Circuits Syst. II-Express Briefs*, vol. 62, no. 8, pp. 786–790, 2015.
[51] A. Sengupta, Y. Ye, R. Y. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: VGG and residual architectures," *Front. Neurosci.*, vol. 13, p. 95, 2019.
[52] G. Datta, S. Kundu, and P. Beerel, "Training energy-efficient deep spiking neural networks with single-spike hybrid input encoding," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.
[53] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, 2012.
[54] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.
[55] A. Bhattacharjee, Y. Kim, A. Moitra, and P. Panda, "Examining the robustness of spiking neural networks on non-ideal memristive crossbars," in *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, 2022, pp. 1–6.
[56] A. Bhattacharjee, L. Bhatnagar, Y. Kim, and P. Panda, "NEAT: Nonlinearity aware training for accurate, energy-efficient, and robust implementation of neural networks on 1T-1R crossbars," *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.*, vol. 41, no. 8, pp. 2625–2637, 2022.

Zekun Deng received the B.S. degree in communications engineering from Northeastern University, Shenyang, China, in 2014. He is currently pursuing the Ph.D. degree in College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His research interests include memristive neural network circuits, algorithm design of neural networks, and analog implementation of neuromorphic systems.

Chunhua Wang received the M.S. degree from Zhengzhou University, Zhengzhou, China, in 1994, and the Ph.D. degree from Beijing University of Technology, Beijing, China, in 2003. He is currently the Professor of College of Information Science and Engineering, Hunan University, Changsha, China. He is the Doctor tutor, director of advanced communication technology key laboratory of hunan universities, the member of academic committee of hunan university, the director of chaos and nonlinear circuit professional committee of circuit and system branch of China electronic society. Now, his research interests include memristor circuit, complex networks, chaotic circuit, chaos secure communication, current-mode circuit and neural networks based on memristor. He has presided over 8 national and provincial projects, and published more than 120 papers, among which more than 100 were retrieved by SCI.

Hairong Lin received M.S. and Ph.D. degree in information and communication engineering and computer science and technology from Hunan University, China, in 2015 and 2021, respectively. He is currently a postdoctoral research fellow at the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His main research interest includes neural system modeling, Chaotic dynamical analysis, nonlinear circuit and neuromorphic engineering. He has published more than 30 papers in related international journals, such as IEEE-TII, IEEE-TIE, IEEE-TCAD, IEEE-TCAS-I, IEEE-TCAS-II, Nonlinear Dynamics, etc.

Yichuang Sun (M'90–SM'99) received the B.Sc. and M.Sc. degrees from Dalian Maritime University, Dalian, China, in 1982 and 1985, respectively, and the Ph.D. degree from the University of York, York, U.K., in 1996, all in communications and electronics engineering.

Dr. Sun is currently Professor of Communications and Electronics, Head of Communications and Intelligent Systems Research Group, and Head of Electronic, Communication and Electrical Engineering Division in the School of Engineering and Computer Science of the University of Hertfordshire, UK. He has published over 330 papers and contributed 10 chapters in edited books. He has also published four text and research books: Continuous-Time Active Filter Design (CRC Press, USA, 1999), Design of High Frequency Integrated Analogue Filters (IEE Press, UK, 2002), Wireless Communication Circuits and Systems (IET Press, 2004), and Test and Diagnosis of Analogue, Mixed-signal and RF Integrated Circuits - the Systems on Chip Approach (IET Press, 2008). His research interests are in the areas of wireless and mobile communications, RF and analogue circuits, microelectronic devices and systems, and machine learning and deep learning.

Professor Sun was a Series Editor of IEE Circuits, Devices and Systems Book Series (2003-2008). He has been Associate Editor of IEEE Transactions on Circuits and Systems I: Regular Papers (2010-2011, 2016-2017, 2018-2019). He is also Editor of ETRI Journal, Journal of Semiconductors, and Journal of Sensor and Actuator Networks. He was Guest Editor of eight IEEE and IEE/IET journal special issues: High-frequency Integrated Analogue Filters in IEE Proc. Circuits, Devices and Systems (2000), RF Circuits and Systems for Wireless Communications in IEE Proc. Circuits, Devices and Systems (2002), Analogue and Mixed-Signal Test for Systems on Chip in IEE Proc. Circuits, Devices and Systems (2004), MIMO Wireless and Mobile Communications in IEE Proc. Communications (2006), Advanced Signal Processing for Wireless and Mobile Communications in IET Signal Processing (2009), Cooperative Wireless and Mobile Communications in IET Communications (2013), Software-Defined Radio Transceivers and Circuits for 5G Wireless Communications in IEEE Transactions on Circuits and Systems-II (2016), and the 2016 IEEE International Symposium on Circuits and Systems in IEEE Transactions on Circuits and Systems-I (2016). He has also been widely involved in various IEEE technical committee and international conference activities.