

Self-Evolutionary Neuron Model for Fast-Response Spiking Neural Networks

Anguo Zhang, Ying Han, Jing Hu, Yuzhen Niu, *Member, IEEE*, Yueming Gao*, *Member, IEEE*, Zhizhang (David) Chen*, *Fellow, IEEE*, and Kai Zhao

Abstract—We propose two simple and effective spiking neuron models to improve the response time of the conventional spiking neural network. The proposed neuron models adaptively tune the presynaptic input current depending on the input received from its presynapses and subsequent neuron firing events. We analyze and derive the firing activity homeostatic convergence of the proposed models. We experimentally verify and compare the models on MNIST handwritten digits and FashionMNIST classification tasks. We show that the proposed neuron models significantly increase the response speed to the input signal. Experiment codes are available at <https://github.com/anvien/Evol-SNN>.

Index Terms—Spiking Neural Network, Self-Evolutionary Neuron Model, Fast Response Network, Synaptic Plasticity.

I. INTRODUCTION

WITH the prevalence of the Internet of Things (IoT), artificial intelligence (AI) theory and technology have advanced rapidly. Among them, artificial-neural-networks (ANNs) based devices and systems have attracted considerable attention. However, due to the limited energy storage capability power consumption and computational efficiency have become a critical factor in designing ANN chips and systems. To this end, research and development efforts have been made, for instance, a lightweight neural framework is

proposed in [1], [2] and an on-demand usage method to avoid unnecessary ANN computation in [3].

Compared with the traditional ANNs that perform dense real-value computation, the spiking neural network (SNN) is more suitable for the AI devices in the edge computing scenarios for its event-driven and sparse-matrix computation. For example, TrueNorth of IBM's neuromorphic chip that apply SNN can run one million spiking neurons at the cost of only 63mW [4], while the traditional ANNs of the same scale require nearly 100,000 to 300,000 times more of the power. The Loihi [5] neuromorphic chip launched by Intel Labs consists of 128 neuromorphic cores which simulates 130K neurons and 130M synapses in real time. Loihi can solve optimization problems, compared with CPU-based solver, Loihi is over three orders of magnitude better in terms of energy-delay-product. Braindrop [6], a 28-nm fabricated process, integrates 4096 neurons in 0.65mm², it uses sparse encoding and weighted spike-rate summation to cut digital traffic drastically. Braindrop consumes energy per equivalent synaptic operation at 381fJ for typical network configurations. Therefore, SNNs are considered energy-friendly at the chip level, have been widely employed in chips such as GPU, FPGA, VLSI, etc. [7]–[11] and have also been used for low-power tasks such as neuromorphic vision and auditory sensors [12], [13].

Modern SNN, also referred to as the third generation ANN, was first proposed by Maass [14]. In contrast with the traditional ANNs, SNNs use discrete and sparse spikes to transmit and process information. SNNs can simulate arbitrary feed-forward sigmoidal ANNs [15] and have been proven to be computationally more efficient than the neurons with sigmoidal activation function. Modern SNNs have been extensively studied in structural and neuron model design [16], [17], learning algorithms [18]–[21], information coding [22], etc., and also have found a wide range of applications, such as object recognition [23], [24], image classification [13], and series data process [25]–[27].

In conventional SNNs, a spike stream of the inputs is applied, it stimulates the neurons in the next layer to generate new spike signals for the following neurons, and the process moves on. On the other hand, an SNN takes time to let output become homeostatic since it needs to get and then process the information from the output layer. An SNN may even take an excessively long time to respond and does not catch up with the input, reducing significantly computational efficiency and real-time performance.

To this end, this paper proposes two computational models

Manuscript received March 3, 2020; revised May 26, 2020. Corresponding author: Yueming Gao (fzugym@gmail.com) and Zhizhang (David) Chen (zz.chen@ieee.org)

This work is supported by the National Natural Science Foundation of China under grant No. U1505251 and No. 61672158, Chinese Ministry of Science and Technology under grant No. 2016YFE0122700, Fujian Provincial Department of Science and Technology under grant No. 2018J0011 and in part by the Natural Science Foundation of Fujian Province under grant 2019J02006

Anguo Zhang is with the College of Physics and Information Engineering, Fuzhou University, Fujian, P. R. China 350108. He is also with the Key Laboratory of Medical Instrumentation and Pharmaceutical Technology of Fujian Province, Fuzhou 350116, and the Research Institute of Ruijie, Ruijie Networks Co., Ltd, Fujian, P. R. China 350002.

Ying Han is with the School of Public Health, Xiamen University, Xiamen 361102, China

Jing Hu is with College of Information and Intelligent Transportation, Fujian Chuanzheng Communications College, Fuzhou 350007, China

Yuzhen Niu is with the Fujian Key Laboratory of Network Computing and Intelligent Information Processing, College of Mathematics and Computer Science, Fuzhou University, and also with the Key Laboratory of Spatial Data Mining and Information Sharing, Ministry of Education, Fujian, P. R. China 350108.

Yueming Gao is with the College of Physics and information Engineering, Fuzhou University, and the Key Laboratory of Medical Instrumentation and Pharmaceutical Technology of Fujian Province, Fujian, P. R. China 350108.

Zhizhang (David) Chen was with the College of Physics and Information Engineering, Fuzhou University, P. R. China 350108, and on leave from the Department of Electrical and Computer Engineering, Dalhousie University, Halifax, NS, Canada B3J1Z1.

Kai Zhao is with the Faculty of Science and Technology, University of Macau, Macau 999078, China.

for fast SNN responses. They are the two integrate-and-fire (IF) neuron based models, one being evolutionary spiking neuron (*Evol*) and the other being adaptive evolutionary spiking neuron (*Adap-Evol*). The proposed model are implemented asynchronously and locally. Here "asynchronous" means that the computation of the spiking neurons is event-triggered, and "locally" means that every neuron updates its state using only its presynaptic input current and neuronal dynamics. Consequently, the proposed neuronal operations can be easily executed without the need for external injections at a low computational cost. Experimental results on MNIST and FashionMNIST datasets confirm the effectiveness of the proposed two models.

In summary, the main contributions of this paper are three-fold:

1) We propose an evolutionary spiking neuron (*Evol*) based on conventional LIF neuron model. A further improved model, i.e., adaptive evolutionary spiking neuron (*Adap-Evol*) is also proposed. We prove through theoretical analysis that both models can reach homeostatic state, stabilize the output of the network, and maximize the transmission information rate between input and output spikes.

2) Through experiments of classification accuracy and matching time, it is demonstrated that the two models we proposed can speed up the network's response speed to the input image signal when performing image classification tasks, and at the same time can ensure that the classification accuracy is not lost. Especially the response speed of *Adap-Evol* is faster.

3) The experiment of neuronal firing activity proves that the *Evol* and *Adap-Evol* models can also reduce the rate of network firing activity, and the network can achieve faster response to input but with fewer spike signals and computational operations triggered. This helps to realize the low-power computation of SNN model if embedded on neuromorphic chip.

The remainder of this paper is organized as follows. In Section II, we briefly review some related works that deal with the response of SNNs. Section III introduces the proposed SNN models with *Evol* and *Adap-Evol* neurons. Section IV presents the theoretical aspects of our proposed models, and Section V presents experimental results. Finally, Section VI summarizes and concludes the paper.

II. RELATED WORKS

The most straightforward and intuitive way to improve the response speed of SNN is to increase the input firing rate and reduce the neuronal firing threshold. However, a high input rate leads to high power cost and memory cost [28]–[30]. A low firing threshold can reduce latency and improve speed, but may reduce accuracy and vice versa. Towards this end, research has been carried out to improve the SNN response speed while maintaining high computational accuracy and low power consumption.

[30] proposes model- and data-based normalization to regulate the firing rates of a spiking fully connected neural network (SFNN) and a spiking convolutional neural network

(SCNN). The normalization boosts the convergence of the spiking neuron firing activity, hence improving the speed and real-time performance. [31] proposes an information transmission method with burst spikes and a layer-wise hybrid neural coding scheme for deep SNNs. The method can improve the speed and power efficiency while maintaining reasonable accuracy for image classification. [32] proposes an adaptive SNN, where the neurons encode information in spike trains using asynchronous spiking sigma-delta coding. It is an order of magnitude faster and uses an order of magnitude fewer spikes than the conventional neural networks. [33] takes into account of neural conductance with structural plasticity, and it improves SNN inference speed. The structural plasticity method is a straightforward ANN without additional operations such as pooling, softmax, etc.. The conductance variation schemes are developed for excitation and inhibitory neurons.

Although the models or methods mentioned above have shown good performances, there still exist some problems or drawbacks. [30] requires global information about the network layers to normalize input weights for each neuron. In [31], the information transmission method and the layer-wise neural coding scheme must be well-designed and carefully implemented for the desired performance. The firing threshold plasticity in [33] is only proposed for straightforward fully-connected SNNs, which can not be applied to the pooling, convolution operations of convolutional neural networks.

III. THE PROPOSED SPIKING NEURON MODELS

In this section, we will present our proposed SNN models. First, we will briefly go through the basics of the conventional spiking neuron models, and then our proposed models with the associated computational algorithms.

A. The Conventional Spiking Neuron Model

As mentioned before, SNNs are considered to be the third generation of ANNs and have shown advantages in computational cost, power consumption, and neural plasticity [14], [15], [34]–[36]. Conventional ANNs feed input data into a network one set at one time, and process information layer by layer, before reaching the output layers. They first convert the input information into spike streams of event signals, and then neurons record these signals and create spikes to communicate information with other connected neurons. Output layer neurons collect firing evidence driven by incoming information and make decisions over time.

Spiking neuron models mimic biological neurons: they have the properties in analogy to physiological properties biological neurons, such as membrane potential v , membrane reset potential v_{reset} , membrane firing potential threshold v_{thr} , capacitance C_m and refractory period t_{ref} , etc. When $v > v_{thr}$, the neuron generates a current spike and transmits it to the dendrites of other neurons; v then immediately drops to v_{reset} and the neuron enters the refractory period. During the refractory period, the neuron does not receive any external current signal from its synapse.

A biological neuron membrane is a semi-permeable biofilm composed of phospholipid molecules. The membrane is non-conductive, aside from protein channels for specific ions; its

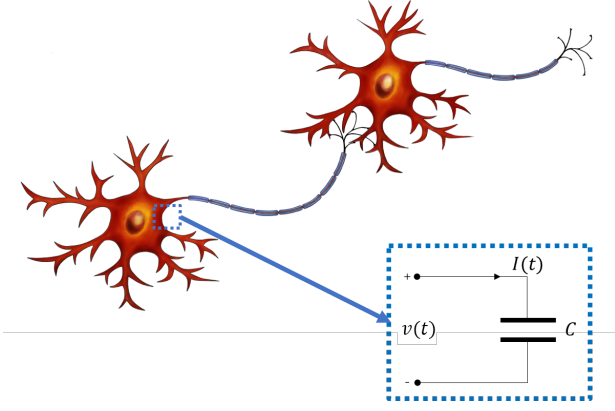


Fig. 1: The semi-permeable biological membrane of biological neurons which composed of phospholipid molecules, and its equivalent capacitor module.

physical properties are similar to a parallel-plate capacitor, as shown, where $v(t)$ is the voltage across the capacitor, $I(t)$ is the current flowing to the capacitor, and C is the capacitance. Fig.1 shows a simple diagram of a parallel-plate capacitor in analogy to a biological neuron. The relation among the voltage, current and capacitance can be expressed as

$$v(t) = \int \frac{1}{C} I(t) dt, \quad (1)$$

where $v(t)$ is the voltage across the capacitor, $I(t)$ is the current flowing to the capacitor, and C is the capacitance.

The capacitor can be adaptively discharged or charged by dynamically changing its capacitance based on the demand. The derivative of $v(t)$ is

$$\dot{v}(t) = \frac{1}{C(t)} I(t), \quad (2)$$

where $C(t)$ is the dynamic capacitance with respect to time t .

The leaky integrate-and-fire (leaky IF) [37] neuron has the similar expression to (2),

$$\frac{dv(t)}{dt} = -v(t) + \sum_{\omega_i \in \Omega} \omega_i I_i(t), \quad (3)$$

where Ω is the set of presynapses for the leaky IF neuron, $v(t)$ is the neuron membrane potential (voltage), $w_i \in \Omega$ is the connection weight between the neuron and its presynapse i , and $I_i(t)$ is the synaptic current input from the neuron's presynapses.

B. Our Proposed Spiking Neuron Model

We can consider (3) as a special case of the following systems,

$$\frac{dv(t)}{dt} = -v(t) + \frac{1}{C(t)} \sum_i \omega_i I_i(t), \quad (4)$$

with $C(t) = 1$.

Define

$$R(t) = \Theta(t - \hat{t} - t_{ref}), \quad (5)$$

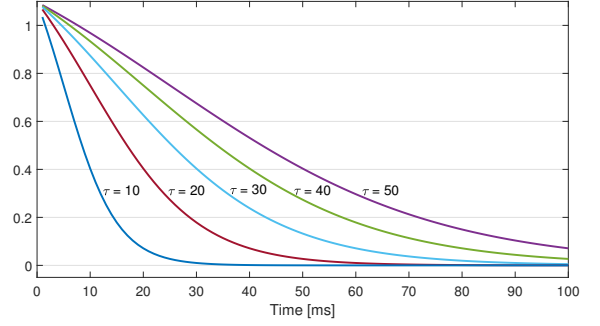


Fig. 2: Variation of $\epsilon(t)$ with respect to time t for different time constants τ .

where $\Theta(\cdot) = 1$ for non-negative arguments, and $\Theta(\cdot) = 0$ otherwise. \hat{t} is the neuron's last firing time, and t_{ref} is the refractory period. Let $E(t) = 1/C(t)$ and the total external input current

$$I(t) = \sum_i \omega_i I_i(t), \quad (6)$$

where $E(t)$ is greater than 0. Then we define the evolutionary tuning law for (4) as

$$\frac{dE(t)}{dt} = \nabla E(t) = \eta \left(\frac{\epsilon}{E(t)} + R(t) \frac{I(t)}{v_{thr} - v_{reset}} (\gamma - (1 + \gamma)O(t)) \right), \quad (7)$$

where η is the update rate, which indicates the strength of the generated spike signal; ϵ is a positive proportion factor; γ is a positive design constant; $O(t)$ is the spike output in response to the total presynaptic input current,

$$O(t) = \sum_f \delta(t - t^{(f)}), \quad (8)$$

where $t^{(f)}$ is the firing time for the spike signal generated by the neuron, and $\delta(\cdot)$ is a Dirac delta function, i.e., $\delta(t - t^{(f)}) = 1$ if $t = t^{(f)}$, and zero otherwise.

From (7), we can know that higher ϵ can effectively increase the rate of change of $E(t)$, and vice versa. Therefore, rather than the conventional approach of defining ϵ as a fixed positive value, we define a time-varying ϵ ,

$$\epsilon(t) = 1 - \frac{1}{1 + \exp(-\frac{2t}{\tau} + 1)}, \quad (9)$$

where t is the relative time since the new signal is input to SNN, and τ is the decay time constant. Fig. 2 shows the variation of $\epsilon(t)$ with respect to t for different τ .

We now propose to use (7) to regulate the neuron model (4), leading to one of our proposed models names as *evolution (Evol) SNN*. When we use both (7) and (9) to regulate the neuron model (4), we obtain the other proposed model named as *adaptive evolution (Adap-Evol) SNN*. The associated computing algorithms are shown in Algorithm 1 and 2.

Algorithm 1: Algorithm of evolutionary spiking neuron model (*Evol*)

Input: Inference Images $\mathcal{X} = \{x_i\}_{i=1}^n$.
Output: Predicted label $C = \{c\}_{i=1}^n$ for each image.
 Setting the parameters k , β and η , initialize C ;
for time t **in** $[1, 2, \dots, T]$ **do**
 for layer **in** layers **do**
 for neuron **in** layer.neurons **do**
 Calculate input current $I(t) = \sum_i^\Omega \omega_i I_i(t)$;
 Calculate $\nabla v(t)$ by Eq.(4);
 Update the membrane potential
 $v(t) = v(t-1) + \nabla v(t)$;
 Calculate $\nabla E(t)$ by Eq.(7);
 Update $E(t) = E(t-1) + \nabla E(t)$,
 $C(t) = \frac{1}{E(t)}$;
 end
 end
 Collect the spike signal from the neurons of output layer;
 Obtain the classification result at time t .
end

Algorithm 2: Algorithm of adaptive evolutionary spiking neuron model (*Adap-Evol*)

Input: Inference Images $\mathcal{X} = \{x_i\}_{i=1}^n$.
Output: Predicted label $C = \{c\}_{i=1}^n$ for each image.
 Setting the parameters k , β , η and the time constant τ , initialize C ;
for time t **in** $[1, 2, \dots, T]$ **do**
 for layer **in** layers **do**
 for neuron **in** layer.neurons **do**
 Calculate input current $I(t) = \sum_i^\Omega \omega_i I_i(t)$;
 Calculate $\nabla v(t)$ by Eq.(4);
 Update the membrane potential
 $v(t) = v(t-1) + \nabla v(t)$;
 Update $\epsilon(t)$ by Eq.(9);
 Calculate $\nabla E(t)$ by Eq.(7);
 Update $E(t) = E(t-1) + \nabla E(t)$,
 $C(t) = \frac{1}{E(t)}$;
 end
 end
 Collect the spike signal from the neurons of output layer;
 Obtain the classification result at time t .
end

IV. THEORETICAL ASPECTS OF THE PROPOSED SNN MODELS

In this section, we will present the theoretical aspects of the proposed models. The results lay the foundations and give the guidelines for their performance enhancements and applications.

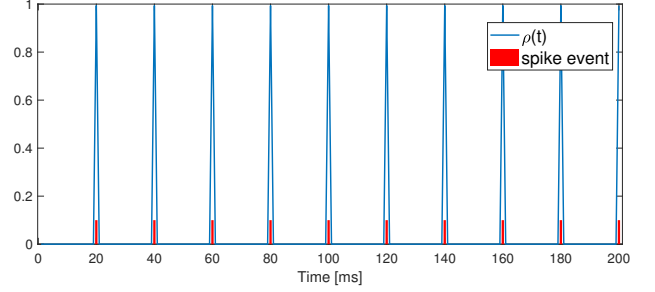


Fig. 3: The value of the firing probability ρ in (10) at a fixed input current corresponds to the firing events.

A. Maximizing Mutual Information between Input Current and Output Spikes

Define an auxiliary notation of the firing probability ρ at time t is related to v and refractory state R ,

$$\rho(t) = 1 - \exp\left(-\frac{v(t) - v_{reset}}{v_{thr} - v_{reset}} mR(t)\right), \quad (10)$$

where $v_{thr} - v_{reset} > 0$ and $m > 1$ is a positive constant. The larger the membrane potential $v(t)$, the higher the auxiliary firing probability. Fig.3 shows the firing probability ρ at a fixed input current corresponds to the firing events, we can see that if the spike event is triggered, ρ is close to 1, otherwise ρ approaches 0. Hence

$$\frac{\partial \rho}{\partial v} = \frac{mR(t)}{v_{thr} - v_{reset}} (1 - \rho), \quad (11)$$

As stated in [38]–[41], maximizing the information transmission of a single spiking neuron strictly locally is to optimize the neuron's output being Weibull distributed (or exponentially distributed, as a special case of Weibull distribution). Since the probability density function for the two-parameter Weibull distribution can be expressed as

$$f_{Weib}(y; \alpha, k) = \frac{k}{\alpha} \left(\frac{y}{\alpha}\right)^{k-1} \exp\left(-\left(\frac{y}{\alpha}\right)^k\right), \quad (12)$$

the Kullback-Leiber divergence (KLD) for the firing probability density function $f_\rho(\rho)$ and the desired Weibull-like distribution function $f_{Weib}(\rho; \alpha, k)$ can be expressed as

$$\begin{aligned} D_{KL}(f_\rho || f_{Weib}) &= \int f_\rho(\rho) \log(f_\rho(\rho)) d\rho - \ln\left(\frac{k}{\alpha^k}\right) \\ &\quad - (k-1) \int f_\rho(\rho) \ln(\rho) d\rho + \frac{1}{\alpha^k} \int f_\rho(\rho) \rho^k d\rho \\ &= -H(\rho) + \frac{1}{\alpha^k} \Xi(\rho^k) - (k-1) \mathbb{E}[\ln(\rho)] - \ln\left(\frac{k}{\alpha^k}\right), \end{aligned} \quad (13)$$

where $\mathbb{E}[\cdot]$ is mathematical expectation.

From the neuronal dynamics in (4),

$$\begin{aligned} \frac{\partial v}{\partial E} &= I \\ \frac{\partial v}{\partial t} &= -v + EI \\ \frac{\partial E}{\partial t} &= \frac{1}{\partial v / \partial E} \cdot \frac{\partial v}{\partial t} = -\frac{1}{I} (v - EI), \end{aligned} \quad (14)$$

and hence the derivative of ρ with respect to E is

$$\begin{aligned}\frac{\partial \rho}{\partial E} &= \frac{\partial \rho}{\partial v} \cdot \frac{\partial v}{\partial t} \cdot \frac{1}{\partial E / \partial t} \\ &= \frac{R}{v_{thr} - v_{reset}} (1 - \rho) (-v + EI) \frac{-I}{v - EI} \\ &= \frac{IR}{v_{thr} - v_{reset}} (1 - \rho),\end{aligned}\quad (15)$$

Since [42]

$$\begin{aligned}H(y) &= - \int f_\rho(\rho) \log(f_\rho(\rho)) d\rho \\ &= \mathbb{E} \left[\ln \left(\frac{\partial \rho}{\partial v} \right) \right] - \mathbb{E} \left[\ln (f_v(v)) \right],\end{aligned}\quad (16)$$

the KLD derivative with respect to E is

$$\begin{aligned}\frac{\partial D_{KL}}{\partial E} &= -\frac{1}{E} + \mathbb{E} \left[\left(\frac{1}{1 - \rho} + \frac{k}{\alpha^k} \rho^{k-1} - \frac{k-1}{\rho} \right) \frac{\partial \rho}{\partial E} \right] \\ &= -\frac{1}{E} + \mathbb{E} \left[\frac{mIR}{v_{thr} - v_{reset}} \left(1 + (1 - \rho) \left(\frac{k}{\alpha^k} \rho^{k-1} - \frac{k-1}{\rho} \right) \right) \right],\end{aligned}\quad (17)$$

which can be simplified by setting $k = 1$ and $\gamma = -1 - \frac{1}{\alpha}$,

$$\begin{aligned}\frac{\partial D_{KL}}{\partial E} &= -\frac{1}{E} + \frac{m}{v_{thr} - v_{reset}} \mathbb{E} \left[IR \left(-\gamma + (\gamma + 1)\rho \right) \right] \\ &= -\frac{1}{E} - \frac{m}{v_{thr} - v_{reset}} \mathbb{E} \left[IR \left(\gamma - (\gamma + 1)\rho \right) \right],\end{aligned}\quad (18)$$

Thus, the gradient descent learning rule for adjusting E is

$$\begin{aligned}E &\leftarrow E - \Delta E = E - \eta \left(\frac{\partial D_{KL}}{\partial E} \right) \\ &= E + \eta \left(\frac{\epsilon}{E} + \frac{mIR}{v_{thr} - v_{reset}} (\gamma - (\gamma + 1)\rho) \right),\end{aligned}\quad (19)$$

where $\epsilon = 1$ for *Evol* and is time-varying for *Adap-Evol*. If we set $t_{ref} = 0$, then $R(t) = 1$, and (19) can be expressed as \dots Since $\rho(t)$ is continuous on $[0, 1]$, we can substitute it using $O(t)$ defined in (8),

$$E \leftarrow E + \eta \left(\frac{\epsilon}{E} + \frac{mIR}{v_{thr} - v_{reset}} (\gamma - (\gamma + 1)O) \right), \quad (20)$$

which is equivalent to the proposed *Evol* model of (7). Combining (20) with time-varying $\epsilon(t)$ from (9) gives the proposed *Adap-Evol* discussed above.

The self-evolutionary rules for E derived above mean that mutual information between the spiking neuron firing event and input current can be maximized by adjusting E for a suitable learning rate η . Thus, mutual information between input current and output spikes for the whole SNN is also maximized since that all neurons are self-evolving.

B. Homeostatic State Convergence

According to the neuron model (4), it is assumed that if the input current $I_i (i \in \Omega)$ of each presynapse is fixed, the whole external input stimulus of the neuron is only related to its state variable $E(t) = 1/C(t)$. Therefore, if $E(t)$ can finally converge to a fixed value, all neurons, that is, the entire SNN will exhibit a fixed firing activity pattern, i.e., the network reaches homeostatic. Only when the network reaches a homeostatic state, its output is consistent and stable, and can be adopted as the final result. In this sub-section, we will prove that $E(t)$ will eventually converge with our proposed models *Evol* and *Adap-Evol*.

From (4), $v(t)$ decreases as absolute $C(t)$ increases; here we ignore presynaptic current input changes. We now consider two extreme cases for (7) with positive initial $E(t)$: one is neuron output $O(t) = 1$, i.e., the neuron fires for each discrete time step; the other is $O(t) = 0$, i.e., the neuron remains inactive for all discrete time steps.

Case 1: $O(t) = 1$. Since $I_i(t), i \in \Omega$ is either zero or positive, $I(t) \geq 0$. Thus, $(\gamma - (1 + \gamma)O(t))I(t)R(t) \leq 0$, and from (7),

$$\nabla E(t) > 0, \text{ if } E(t) < \frac{\epsilon(v_{thr} - v_{reset})}{(-\gamma + (1 + \gamma)O(t))mI(t)R(t) + \sigma}$$

and

$$\nabla E(t) < 0, \text{ if } E(t) > \frac{\epsilon(v_{thr} - v_{reset})}{(-\gamma + (1 + \gamma)O(t))mI(t)R(t) + \sigma}$$

with positive γ , and σ being small positive numbers to prevent the denominator = 0. Therefore, $E(t)$ will converge as (or track)

$$\frac{\epsilon(v_{thr} - v_{reset})}{(-\gamma + (1 + \gamma)O(t))mI(t) + \sigma}$$

Fig. 4 shows $E(t)$ with respect to time with different constant $I(t)$ (Fig. 4(a)) and random $I(t)$ with different means (4(b)). In other words, when the presynaptic input current has fixed mean, and the neuron fires for each time step in the early stage, it will continue to fire for subsequent time steps.

Case 2: $O(t) = 0$. Since $(\gamma - (1 + \gamma)O(t))mI(t)R(t)/(v_{thr} - v_{reset}) \geq 0$, gradient $\nabla E(t)$ will always be larger than 0, hence $E(t)$, i.e., $1/C(t)$, monotonically increases over $[0, +\infty)$. From (4) and (7) the neuron membrane potential will increase more rapidly than that in the case where $C(t)$ is constant, generating firing quicker, and hence responding faster to the input signal.

If a neuron does not produce a spike (i.e., $O(t) = 0$), then *Evol* and *Adap-Evol* rules increase the neuron membrane potential sensitivity to the input current. When the neuron produces a spike, i.e., $O(t) = 1$, it returns to case 1. Even when the neuron membrane potential sensitivity to input current increase, the neuron will not necessarily fire in a short period, because the neurons require presynaptic input currents that are not too small.

V. EXPERIMENT RESULTS

During the SNN inference process for image classification, spike streams are generated from the original images are

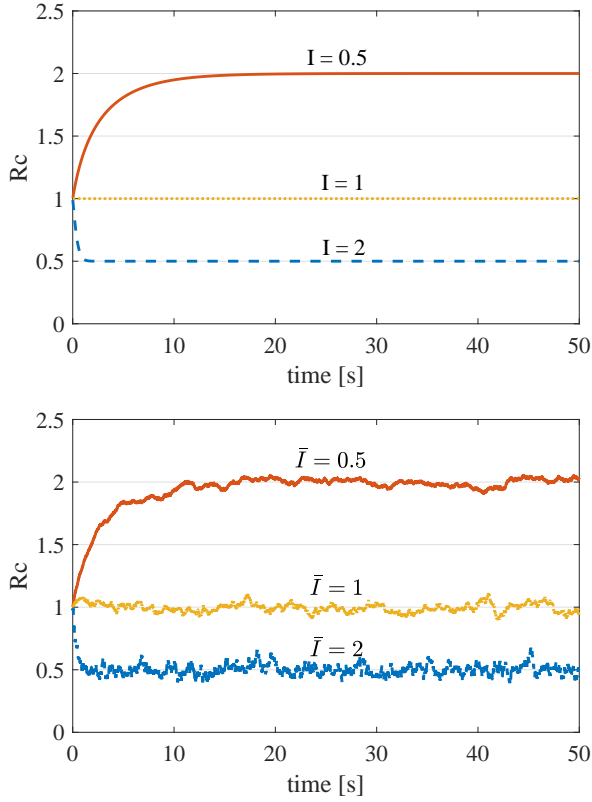


Fig. 4: Variation of $E(t)$ over time with different values of the presynaptic input current $I(t)$.

continuously input to the network input layer. Neurons in each SNN layer then generate spike signals for neurons in subsequent layers, and the output layer collects the spike response to reconstruct the original image class. What needs to be pointed out is we use the total count of spike of the output neurons to determine the class label at the end of simulation time. The network spike response obtained by the output layer may differ over time, and hence SNN classification accuracy may also change. SNN accuracy will only converge once the neuronal state becomes homeostatic. Thus, not only classification accuracy, but also response time (latency) and the total number of synaptic events (spikes) should be considered when evaluating SNN recognition performance [43]. We compare SFNN/Evol vs SFNN/Adap-Evol-SFNN vs SCNN/Evol with SCNN/Adap-Evol-SCNN performance metrics on MNIST and Fashion MNIST datasets. They are elaborated as follows.

A. Datasets

We use MNIST and FashionMNIST [44] datasets for testing our proposed models. FashionMNIST is a benchmark dataset for machine learning algorithms, intended as a direct drop-in replacement for the original MNSIT dataset. It comprises 60000 training and 10000 test examples, each being a 28x28 grayscale image associated with a label from 10 classes: T-shirt, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, and Ankle boot.

We use ANN-to-SNN conversion to obtain network hyperparameters for final SNN, i.e., the connection weights.

Therefore, image pixel values can be directly input into the corresponding ANN after being normalized. However, SNN can only receive discrete spike signal inputs, hence the image data is first converted into spike signals using Poisson distributed spike encoding [30], [45]. Fig. 5 shows the example of Poisson sampling to generate spike streams for SNN input.

B. Specifics about Spiking Networks with Evolutionary Neuron Model

We construct a fully-connected feed-forward (FNN) and convolutional (CNN) neural network, as the fundamental computing frameworks for the proposed evolutionary neuron models. These two ANN types, particularly CNN, have been widely used for image classification. Both FNN and CNN models are trained on the MNIST and FashionMNIST datasets.

However, direct training of an event-based spiking network, i.e., an SNN, can be complicated and challenging due to undifferentiable discrete and asynchronous neuronal spike events. Therefore, we use the conversion of a trained ANN into an SNN of equivalent structure, adopting conversion methods for network computing operations, such as convolution, average pooling, max pooling, fully connection propagation, ReLU activation, softmax activation, etc [28]–[30], [46], [47]. We also use ANN-to-SNN conversion to obtain the fundamental network weights and replace rate-based neural activation with the proposed evolutionary and adaptive evolutionary spiking neuron models. Thus, the steps to construct the SFNN and SCNN are as follows.

- 1) Implement FNN and CNN with ReLU activation function and set no bias in each layer;
- 2) Train the FNN and CNN using error backpropagation algorithm;
- 3) Map trained ANN weights to corresponding SNNs with the same structure with IF neurons for the proposed evolutionary neuron models.

We use two network architectures to verify the proposed evolutionary neuron model and their efficiencies. We first constructed a 784-1200-1200-10 FNN, i.e., input dimension = 784 (28×28 pixel input images) and output dimension = 10 (number of image classes). The FNN included two hidden layers with 1200 neurons per layer. For the convolutional neural network, we construct a 28x28-12c5-2s-64c5-2s-10o CNN. It has 12 5×5 convolutional kernels after the input layer. It generates feature maps that are then convoluted with 64 5×5 convolutional kernels, followed by two 2×2 average pooling steps. The resulting 64 feature maps are flattened and fully linked to the final ten output neurons.

C. Results with MNIST Dataset

1) *Classification Accuracy and Convergence Time:* Figs. 6 and 7 show classification accuracy for comparisons among SFNN, Evol-SFNN, Adap-Evol-SFNN and SCNN, Evol-SCNN, Adap-Evol-SCNN, with running time step = 1 ms. Each subgraph in Figs.6 and 7 compares SFNN or SCNN classification accuracy, respectively, with the same input frequency.

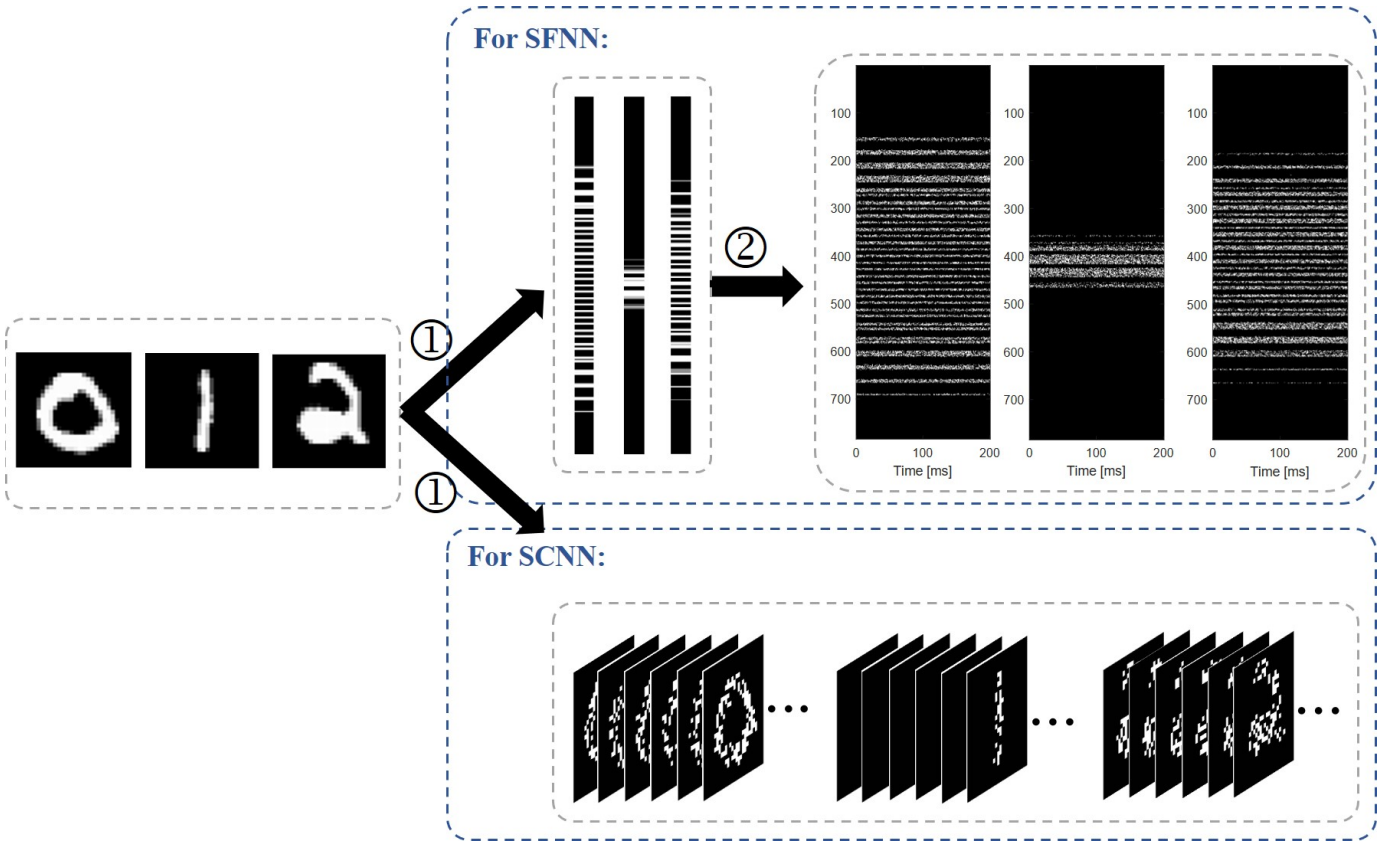


Fig. 5: Poisson sampling to generate spike streams of SNN input. For SFNN, it needs two steps, ①: Vectorize the 28*28 image pixels into 784 input neurons, image pixel is normalized to range [0, 1]. ②: Generate spike streams with two states of firing or no firing by using Poisson sampling, where the firing frequency of spike stream is proportional to the image intensity. The “white dot” in spike streams denotes a spike event at current time step, while the “black dot” denotes no spike generated. For SCNN, it needs the step as ② of SFNN, the difference is that at each time step, the method generate a 2D stream with two states of firing or no firing.

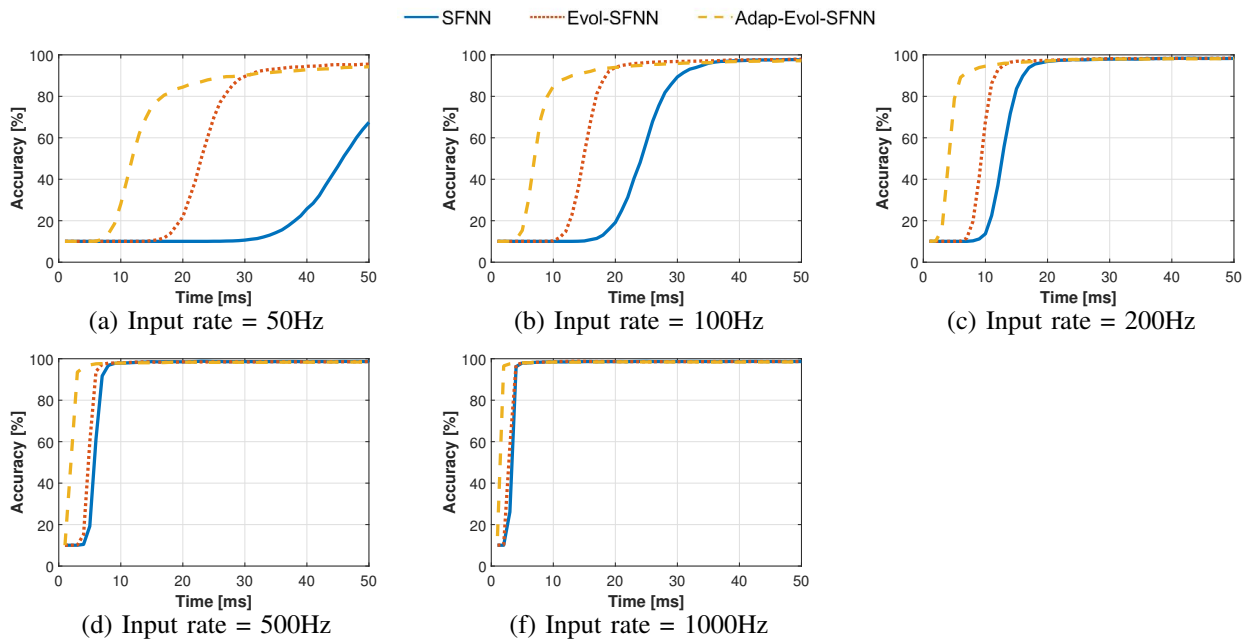


Fig. 6: Classification accuracy for SFNN, Evol-SFNN, and Adap-Evol-SFNN over time for different input firing rates.

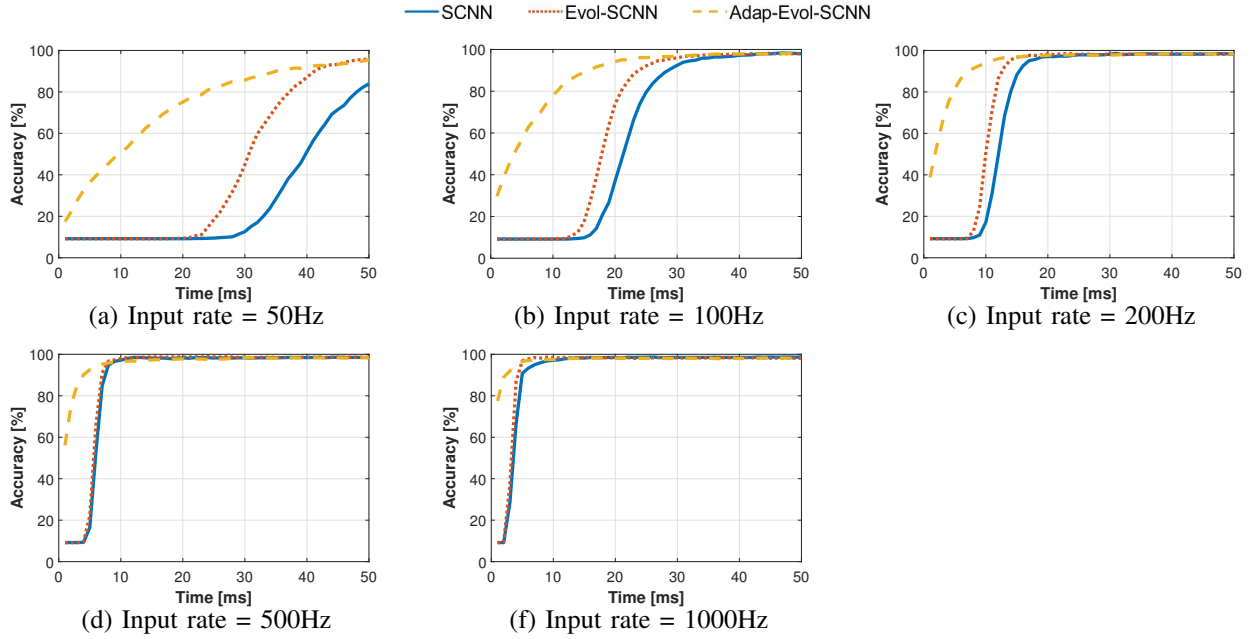


Fig. 7: Classification accuracy for SCNN, Evol-SCNN, and Adap-Evol-SCNN over time for different input firing rates.

Classification accuracy error rates $\approx 90\%$ initially, but reduce rapidly after a short time to collect output signals from the output layer, and eventually stabilize at a relatively small error. The stable classification error rate means the SNN neurons have converged to a homeostatic state. Faster convergence indicates less time to reach homeostasis state and produce stable output, which is beneficial for low latency real-time SNNs.

The proposed Evol SNN and Adap-Evol SNN improve the network response speed while retaining high classification accuracy for low (50 Hz) or high (1000 Hz) input rates, hence shortening response time to the input. Adapt-Evol significantly improves network responsivity compared with Evol, particularly for low input rate. Indeed, Adap-Evol-SCNN classification accuracy is larger than 10% even for the initial time step, indicating that Adap-Evol-SCNN achieved zero-latency response for at least some input signals (spike signals obtained by Poisson sampling).

2) *Neuron Firing Activity*: Figs. 8 and 9 show neuron firing activity statistics for the main SFNN and SCNN network layers while classifying an image. We only counted neuron firing activity from initial input signal reception to when classification accuracy begins to converge. Therefore, the statistics may differ for different networks, e.g. SFNN and SCNN exhibit longest, Evol-SFNN and Evol-SCNN slightly shorter, and Adap-Evol-SFNN and Adap-Evol-SCNN shortest time.

Firing activity depends on the number of spike events generated during a period. Input signal spike densities should be approximately equal for the same input rate, but input spike count differs. As seen from Fig. 8 and Fig. 9, the proposed Adap-Evol-SFNN and Adapt-Evol-SCNN have the least amount of spikes, and SFNN and SCNN the most. However, spike events of SFNN inner layer neuron are not

significantly different even though the input rates differ widely (200 and 500 Hz, respectively); it holds for all SFNN variants considered. Spike event sums for Layer 2 and 3 and output are also approximately equal between the two input rates, although Evol-SFNN is somewhat less than the other two networks.

The Adap-Evol-SFNN inhibits the previously hidden layer (Layer 2) but enhances the next hidden layer (Layer 3) firing activity compared with SFNN and Evol-SFNN. It has the advantage that more spike events directly stimulate output layer neurons; hence the output neurons can respond to input signals faster.

The SCNN, Evol-SCNN, and Adap-Evol-SCNN firing activities are similar to the SFNN cases (Fig. 9), i.e., less neuronal spike events for Evol-SCNN inner layers than for SCNN. However, Adap-Evol-SCNN appears to move spike events from the previous layer to the next layer, increasing output neuron response speed.

D. Results with FashionMNIST Dataset

We quantitatively evaluated SNN, the proposed Evol-SNN, and the proposed Adap-Evol-SNN performances on the FashionMNIST dataset, including simulation time required to reach 99% of minimum classification error, i.e., matching time (MT) [32]. Final classification accuracy (FA) is also computed, representing the stable precision when the SNN reached homeostasis. The smaller MT, the faster of relatively reliable output. The larger FA, the more accurate the output results.

Tab.I compares FA and MT among SNN, the proposed Evol-SNN, the proposed Adap-Evol-SNN at different input firing rates. Adap-Evol-SFNN and Adap-Evol-SCNN have the lowest MT at all input firing rates, suggesting that the proposed Adap-Evol mechanism accelerates SNN response speed. In contrast, the Evol-SNN or Adap-Evol-SNN have the highest FA at most input firing rates. Thus, Evol and

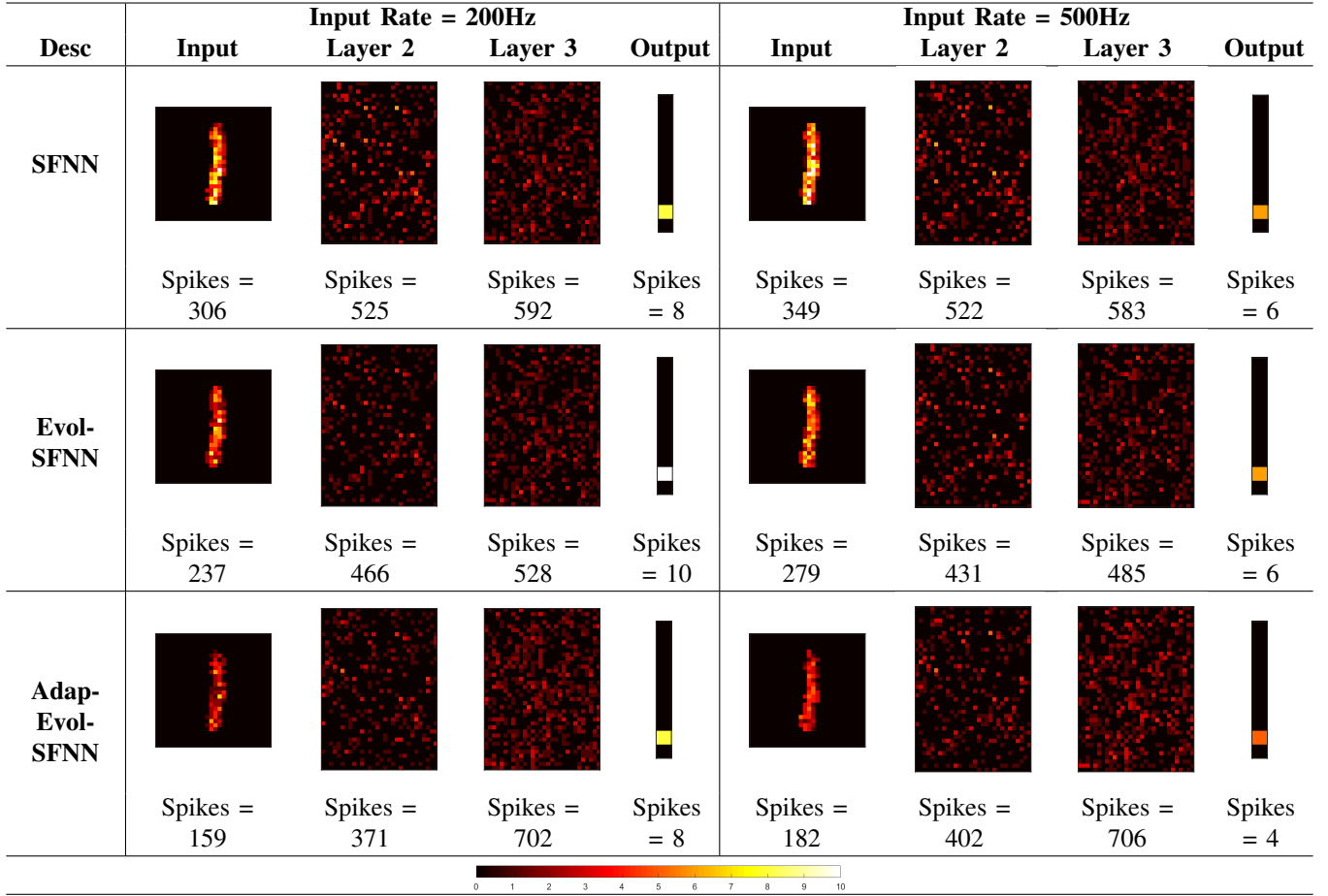


Fig. 8: Firing activities for SFNN, Evol-SFNN, and Adap-Evol-SFNN, where Layer 2 and Layer 3 refer to the 1st and 2nd hidden layers, respectively.

particularly Adap-Evol slightly improve classification accuracy while simultaneously improving response speed.

E. Evolution of Variables $dv(t)$ And $C(t)$ over Time

Based on (4), $dv(t)$ represents IF neuron firing activity, the larger $dv(t)$, the higher rate or frequency of the neuronal spike events. We use SFNN, Evol-SFNN and Adap-Evol-SFNN to infer a handwritten digit 2 of MNIST, respectively. Fig.10 compares neuronal $dv(t)$ variations over time for fixed input firing rate = 200Hz, where Layer 2 and Layer 3 refer to the 1st and 2nd hidden layer respectively, and Output denotes output layer neurons. Almost all the hidden neurons have $dv(t) \approx 0$, although Adap-Evol-SFNN significantly increases $dv(t)$ difference among hidden neuron. However, this increase is somewhat less than that for Evol-SFNN. Differences between neuron $dv(t)$ become larger over time for both Evol-SFNN and Adap-Evol-SFNN, with positive and negative $dv(t)$ occurring.

On the other hand, output neuron 3 has larger $dv(t)$ than the other 9 neurons for all SFNN, the proposed Evol-SFNN, and Adap-Evol-SFNN, since neuron 3 is the expected index for digit 2. Output layer neuron of Adapt-Evol-SFNN has much larger $dv(t)$ than Evol-SFNN and SFNN (approximately 5000 vs. 100 and 1, respectively). It means that Adap-Evol-

SFNN output neurons can acquire more spike stimuli, i.e., more synaptic currents are input to this neuron. Therefore, its output neuron can collect sufficient information to classify the input signal significantly earlier than those of SFNN and Evol-SFNN.

Fig. 11 shows $C(t)$ variations of Evol-SFNN and Adap-Evol-SFNN while inferring a handwritten digit 2. Layer 2 and 3 results confirm that the ranges of $C(t)$ variations differ between Evol-SFNN and Adap-Evol-SFNN. Although both networks exhibit downward $C(t)$ from 0 to 100 ms, Adap-Evol-SFNN decreases faster than Evol-SFNN. Neuron index 3 has the smallest $C(t)$ in all cases, which is consistent with neuron index 3 has the largest $dv(t)$ among all output neurons (see Fig. 10).

VI. CONCLUSION

In contrast to the conventional second-generation rate-based ANNs, SNNs use spike event streams to process information. As a result, they require a shorter time to collect sufficient spike information to produce reliable output even during the inference stage. This paper proposed the new self-evolutionary (*Evol*) and subsequent adaptive self-evolutionary (*Adap-Evol*) SNN models. Both models dynamically regulate neuron membrane conductivity based on neuronal spiking activity and

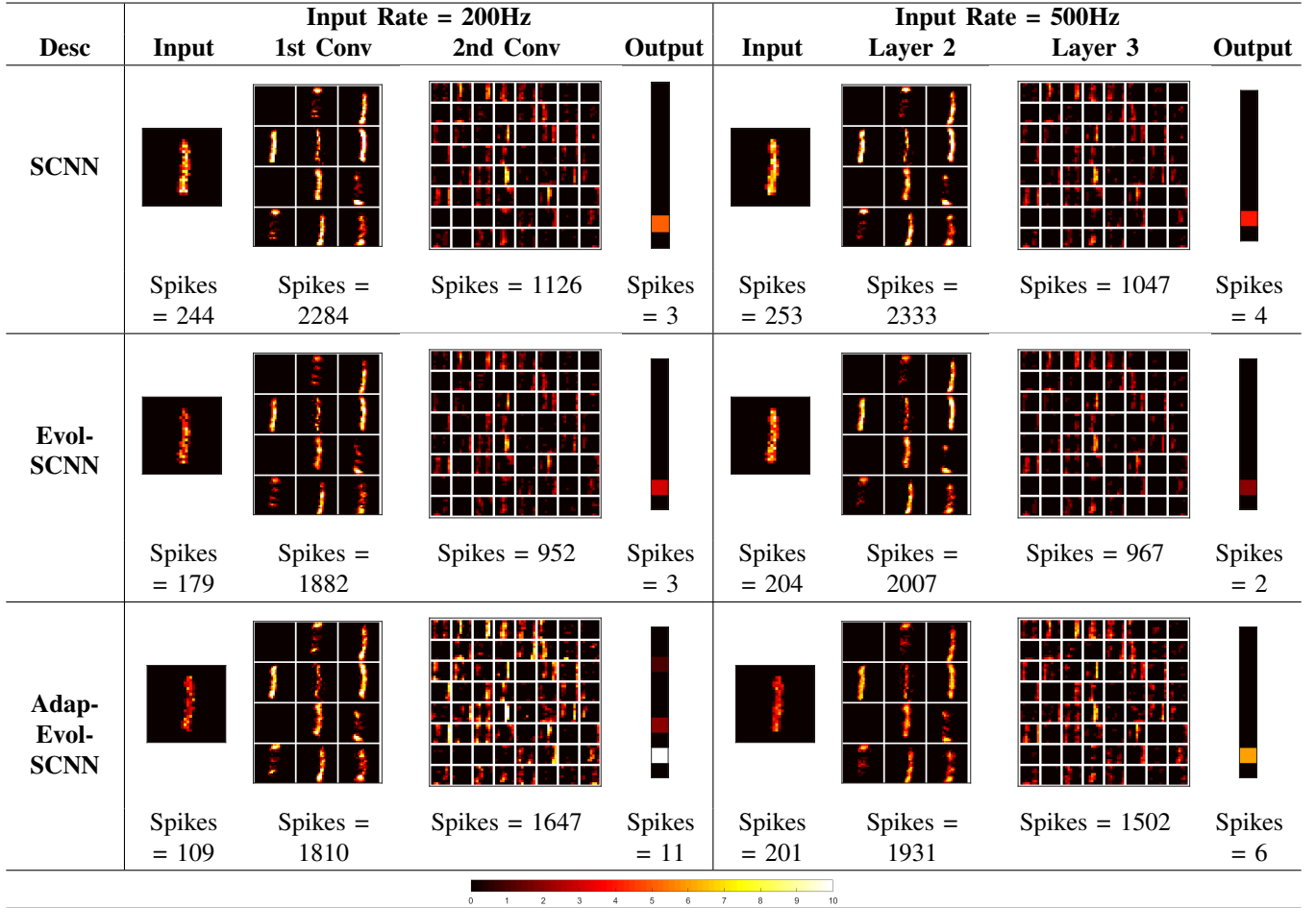


Fig. 9: Firing activities for SCNN, Evol-SCNN, and Adap-Evol-SCNN, where 1st Conv and 2nd Conv refer to the first and second convolutional layers, respectively.

TABLE I: Performance comparisons among SNN, Evol-SNN and Adap-Evol-SNN at different input firing rates on Fashion-MNIST.

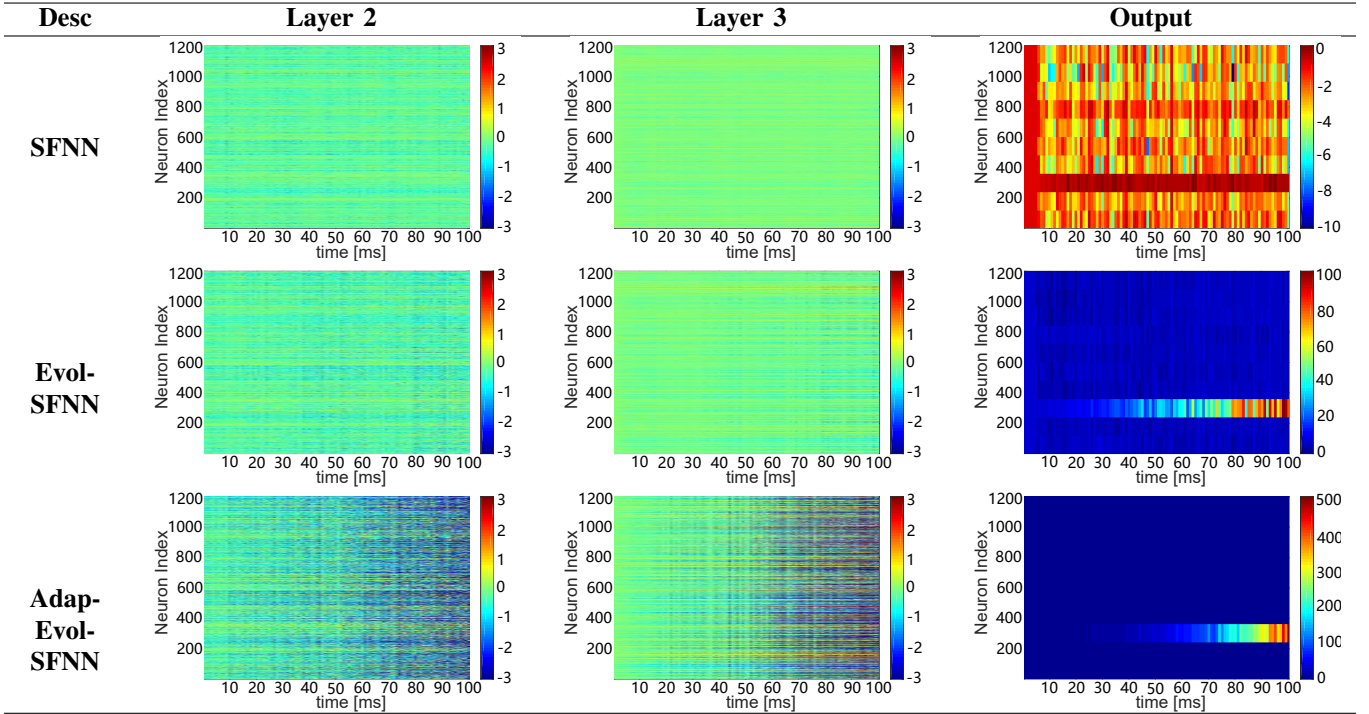
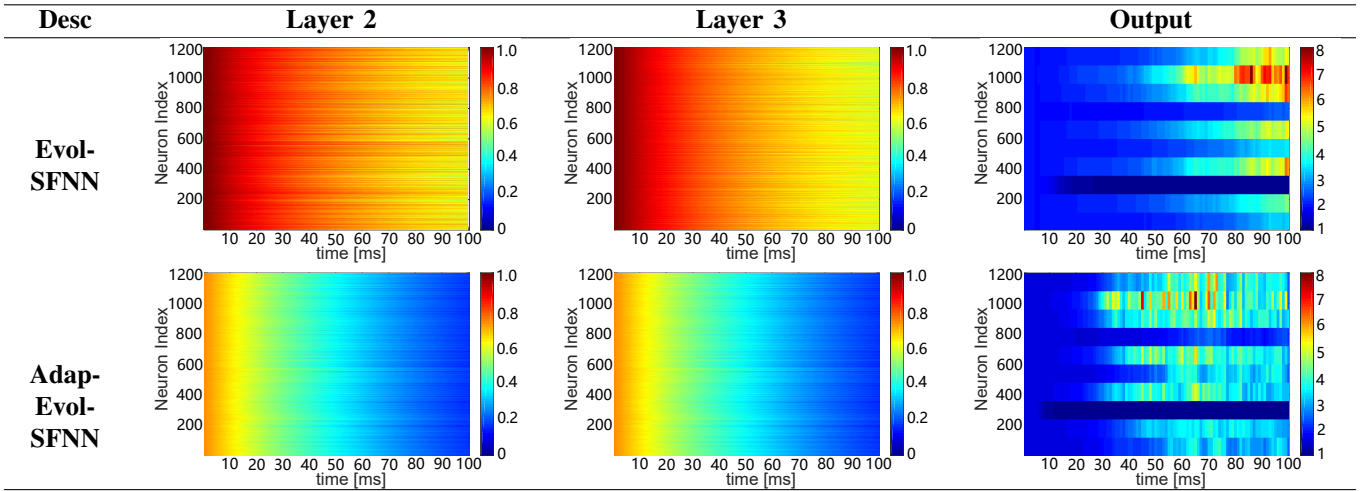
Model	Metric	50Hz	100Hz	200Hz	500Hz	1000Hz
SFNN	FA [%]	88.5	89.4	90.5	91.2	90.6
	MT [ms]	385	252	167	76	42
Evol-SFNN	FA [%]	88.4	89.6	90.6	91.3	90.5
	MT [ms]	247	163	116	66	38
Adap-Evol-SFNN	FA [%]	89.9	89.7	91.2	91.2	91.4
	MT [ms]	172	125	78	45	28
SCNN	FA [%]	90.2	89.9	91.2	90.5	91.2
	MT [ms]	391	288	151	66	49
Evol-SCNN	FA [%]	89.4	90.6	90.4	91.1	91.0
	MT [ms]	233	142	109	53	34
Adap-Evol-SCNN	FA [%]	90.1	90.3	91.0	91.3	91.6
	MT [ms]	184	119	82	52	26

external current input. After the input signal is input to the SNN, the proposed models can quickly activate neuronal homeostasis by increasing neuron responses to the input signal. We validate the effectiveness of the proposed approaches with MNIST and FashionMNIST datasets.

Theoretical analysis and experimental results confirm significant improvement in terms of reducing output delay to reach accuracy convergence even without the need for increased spike firing rate with our two proposed models.

REFERENCES

- [1] S. Y. Nikouei, Y. Chen, S. Song, R. Xu, B. Choi, and T. R. Faughnan, "Real-time human detection as an edge service enabled by a lightweight cnn," in *2018 IEEE International Conference on Edge Computing (EDGE)*, 2018, pp. 125–129.
- [2] C. Li, X. Ma, Z. An, and Y. Xu, "A deep neural network compression algorithm based on knowledge transfer for edge device," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, 2018, pp. 334–335.
- [3] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge ai: On-demand accelerating deep neural network inference via edge computing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 447–457, 2020.
- [4] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo,

Fig. 10: Evolution of $dv(t)$ from (4) for fully connected SNNs.Fig. 11: Evolution of $C(t)$ from (4) for fully connected SNNs.

- I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, aug 2014.
- [5] M. Davies, N. Srinivasa, T. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y. Weng, A. Wild, Y. Yang, and H. Wang, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.
- [6] A. Neckar, S. Fok, B. V. Benjamin, T. C. Stewart, N. N. Oza, A. R. Voelker, C. Eliasmith, R. Manohar, and K. Boahen, "Braindrop: A mixed-signal neuromorphic architecture with a dynamical systems-based programming model," *Proceedings of the IEEE*, vol. 107, no. 1, pp. 144–164, 2019.
- [7] S. Fusi, P. Del Giudice, and D. Amit, "Neurophysiology of a VLSI spiking neural network: LANN21," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*. IEEE, 2002, pp. 121–126 vol.3.
- [8] J. Schemmel, A. Grubl, K. Meier, and E. Mueller, "Implementing Synaptic Plasticity in a VLSI Spiking Neural Network Model," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE, 2008, pp. 1–6.
- [9] A. Afifi, A. Ayatollahi, and F. Raissi, "Implementation of biologically plausible spiking neural network models on the memristor crossbar-based CMOS/nano circuits," *ECCTD 2009 - European Conference on Circuit Theory and Design Conference Program*, pp. 563–566, 2009.
- [10] R. Wang, G. Cohen, K. M. Stiefel, T. J. Hamilton, J. Tapson, and A. van Schaik, "An FPGA Implementation of a Polychronous Spiking Neural Network with Delay Adaptation," *Frontiers in neuroscience*, vol. 7, no. February, p. 14, 2013.
- [11] K. Minkovich, C. M. Thibault, M. J. O'Brien, A. Nogin, Y. Cho, and N. Srinivasa, "Hrslsim: A high performance spiking neural network simulator for gpgpu clusters," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 316–331, feb 2014.

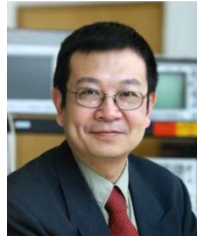
- [12] F. Naveros, N. R. Luque, J. A. Garrido, R. R. Carrillo, M. Anguita, and E. Ros, "A Spiking Neural Simulator Integrating Event-Driven and Time-Driven Computation Schemes Using Parallel CPU-GPU Co-Processing: A Case Study," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 7, pp. 1567–1574, jul 2015.
- [13] E. Stamatias, M. Soto, T. Serrano-Gotarredona, and B. Linares-Barranco, "An Event-Driven Classifier for Spiking Neural Networks Fed with Synthetic or Dynamic Vision Sensor Data," *Frontiers in Neuroscience*, vol. 11, no. JUN, jun 2017.
- [14] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.
- [15] —, "Fast sigmoidal networks via spiking neurons," *Neural Computation*, vol. 9, no. 2, pp. 279–304, feb 1997.
- [16] K. K. Parhi and N. K. Unnikrishnan, "Brain-Inspired Computing: Models and Architectures," *IEEE Open Journal of Circuits and Systems*, vol. 1, no. October, pp. 185–204, 2020.
- [17] B. Han, G. Srinivasan, and K. Roy, "RMP-SNN: Residual membrane potential network for enabling deeper high-accuracy and low-latency spiking neural network," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 13 555–13 564, 2020. [Online]. Available: <http://arxiv.org/abs/2003.01811>
- [18] M. Zhang, H. Qu, A. Belatreche, and X. Xie, "Empd: An efficient membrane potential driven supervised learning algorithm for spiking neurons," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 2, pp. 151–162, 2018.
- [19] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *Frontiers in Neuroscience*, vol. 12, no. MAY, may 2018. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5974215/><https://www.frontiersin.org/article/10.3389/fnins.2018.00331/full>
- [20] M. Zhang, H. Qu, X. Xie, and J. Kurths, "Supervised learning in spiking neural networks with noise-threshold," *Neurocomputing*, vol. 219, pp. 333–349, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231216310724>
- [21] C. Hong, X. Wei, J. Wang, B. Deng, H. Yu, and Y. Che, "Training Spiking Neural Networks for Cognitive Tasks: A Versatile Framework Compatible with Various Temporal Codes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 4, pp. 1285–1296, 2020.
- [22] B. Petro, N. Kasabov, and R. M. Kiss, "Selection and Optimization of Temporal Spike Encoding Methods for Spiking Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 2, pp. 358–370, 2020.
- [23] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "STDP-based spiking deep convolutional neural networks for object recognition," *Neural Networks*, vol. 99, pp. 56–67, 2018.
- [24] Y. Cao, Y. Chen, and D. Khosla, "Spiking deep convolutional neural networks for energy-efficient object recognition," *International Journal of Computer Vision*, vol. 113, no. 1, pp. 54–66, 2015.
- [25] N. Kasabov and E. Capecchi, "Spiking neural network methodology for modelling, classification and understanding of EEG spatio-temporal data measuring cognitive processes," *Information Sciences*, vol. 294, pp. 565–575, feb 2015.
- [26] J. P. Dominguez-Morales, Q. Liu, R. James, D. Gutierrez-Galan, A. Jimenez-Fernandez, S. Davidson, and S. Furber, "Deep Spiking Neural Network model for time-variant signals classification: A real-time speech recognition approach," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 2018-July. IEEE, jul 2018, pp. 1–8.
- [27] A. Zhang, W. Zhu, and J. Li, "Spiking echo state convolutional neural network for robust time series classification," *IEEE Access*, vol. 7, pp. 4927–4935, 2019.
- [28] B. Rueckauer and S. C. Liu, "Conversion of analog to spiking neural networks using sparse temporal coding," *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 2018-May, 2018.
- [29] P. O'Connor, D. Neil, S. C. Liu, T. Delbruck, and M. Pfeiffer, "Real-time classification and sensor fusion with a spiking deep belief network," *Frontiers in Neuroscience*, vol. 7, no. 7 OCT, pp. 1–13, 2013.
- [30] P. U. Diehl, D. Neil, J. Binas, M. Cook, S. C. Liu, and M. Pfeiffer, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 2015-September, 2015.
- [31] S. Park, S. Kim, H. Choe, and S. Yoon, "Fast and Efficient Information Transmission with Burst Spikes in Deep Spiking Neural Networks," no. 1, sep 2018.
- [32] D. Zambrano, "Fast and Efficient Asynchronous Neural Computation with Adapting Spiking Neural Networks," no. December, 2016.
- [33] R. Spiess, R. George, M. Cook, and P. U. Diehl, "Structural Plasticity Denoises Responses and Improves Learning Speed," *Frontiers in Computational Neuroscience*, vol. 10, 2016.
- [34] W. Maass, "Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons," *Advances in Neural Information Processing Systems*, vol. 9, p. 211, 1997.
- [35] M. Galarreta and S. Hestrin, "A network of fast-spiking cells in the neocortex connected by electrical synapses," *Nature*, vol. 402, no. 6757, pp. 72–75, 1999.
- [36] A. Aertsen, M. Diesmann, and M.-O. Gewaltig, "Stable propagation of synchronous spiking in cortical neural networks : Abstract : Nature," *Nature*, vol. 402, no. 6761, pp. 529–533, 1999.
- [37] G. Holt and C. Koch, "Shunting inhibition does not have a divisive effect on firing rates," *Neural Computation*, vol. 9, no. 5, pp. 1001–1013, 1997.
- [38] P. Joshi and J. Triesch, "Rules for information maximization in spiking neurons using intrinsic plasticity," in *2009 International Joint Conference on Neural Networks*, vol. 8. IEEE, jun 2009, pp. 1456–1461.
- [39] K. Neumann and J. J. Steil, "Intrinsic plasticity via natural gradient descent," *ESANN 2012 proceedings, 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 555–560, 2012.
- [40] C. Li and Y. Li, "A spike-based model of neuronal intrinsic plasticity," *IEEE Transactions on Autonomous Mental Development*, vol. 5, no. 1, pp. 62–73, mar 2013.
- [41] A. Zhang, H. Zhou, X. Li, and W. Zhu, "Fast and robust learning in Spiking Feed-forward Neural Networks based on Intrinsic Plasticity mechanism," *Neurocomputing*, vol. 365, pp. 102–112, nov 2019. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0925231219309348>
- [42] E. T. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, pp. 1129–1159, 1995.
- [43] Q. Liu, G. Pineda-García, E. Stamatias, T. Serrano-Gotarredona, and S. B. Furber, "Benchmarking Spike-Based Visual Recognition: A Dataset and Evaluation," *Frontiers in Neuroscience*, vol. 10, no. November, nov 2016.
- [44] H. Xiao, K. Rasul, and R. Vollgraf, "(2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.
- [45] S. Roy and A. Basu, "An online unsupervised structural plasticity algorithm for spiking neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 4, pp. 900–910, 2017.
- [46] B. Rueckauer, I. A. Lungu, Y. Hu, M. Pfeiffer, and S. C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers in Neuroscience*, vol. 11, no. DEC, pp. 1–12, 2017.
- [47] S. Kim, S. Park, B. Na, and S. Yoon, "Spiking-YOLO: Spiking Neural Network for Real-time Object Detection," *Arxiv*, mar 2019.



Anguo Zhang was born in Hefei city, Anhui province, China in 1990. He received his bachelor's degree and master's degree in control engineering from Chongqing University, Chongqing in 2012 and 2016, respectively. Since 2018, he has been a researcher and senior engineer in the Research Institute of Ruijie, Ruijie Networks Co., Ltd. He is also pursuing a Ph.D. in Communication and Information Systems at Fuzhou University. His research interest includes machine learning, artificial neural networks, control theory and applications.

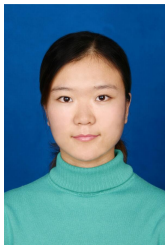


Ying Han (1992-) received the B.S. degree in Medicine from the Fujian Medical University, Fuzhou, in 2015 and M.S. degree in Nursing from Fujian Medical University in 2018. She is currently a doctoral student in Xiamen University majoring in biology. Her previous research was in nursing of post-operation for renal transplantation patients and management in manpower resources of hospital.



Zhizhang (David) Chen (S'92-M'92-SM'96-F'10) received the B.Eng. Degree in radio engineering from Fuzhou University, Fuzhou, China, the master's degree in radio engineering from Southeast University, Nanjing, China, and the Ph.D. degree in electrical engineering from the University of Ottawa, Ottawa, ON, Canada.

He was an NSERC Post-Doctoral Fellow with McGill University, Montreal, QC, Canada, in 1993. He was with Fuzhou University, on leave from Dalhousie University, Halifax, NS, Canada, where he is a Professor. He has been an Adjunct or Visiting Professor with the University of Nottingham, Nottingham, U.K., École Nationale Supérieure des Télécommunications de Bretagne of France, Brest, France, Shanghai Jiao Tong University, Beijing, China, the University of Aeronautics and Astronautics, Nanjing, Fuzhou University, Hong Kong University of Science and Technology, Hong Kong, and the University of Electronic Science and Technology of China, Chengdu, China. He is the Fellow of the Canadian Academy of Engineering and the Engineering Institute of Canada. He has authored or co-authored over 450 journal and conference papers in applied electromagnetics, RF/microwave electronics, antennas, and emerging wireless technologies.



Jing Hu was born in Pingdu city, Shandong province, she received her master's degree in computer science and technology from Fuzhou University in 2017. She is currently a full-time associate professor in the College of Information and Intelligent Transportation, Fujian Chuanzheng Communications College. Her research interests include data mining, machine learning and artificial neural networks.



Yuzhen Niu received her Ph.D. in Computer Science from the Shandong University of China in 2010. She was a Post-Doctoral Researcher with the Department of Computer Science, Portland State University, Portland, OR. She is currently a Professor with the College of Mathematics and Computer Science, Fuzhou University, China. Her current research interests include image and video processing, computer vision, artificial intelligence, and multimedia.



Kai Zhao received the Ph.D. degree in Control Theory and Control Engineering from Chongqing University, Chongqing, China, in 2019. He is currently a Post-Doctoral Fellow with the Faculty of Science and Technology, University of Macau, Macau.

His research interests include intelligent control, adaptive control, robotic control, and constrained nonlinear systems.



Yueming Gao received the Ph.D. degree in electrical engineering from Fuzhou University, Fuzhou, China, in 2010. Since 2004, he has been involved in research in the areas of bio-electromagnetism and biomedical circuits and systems. He is currently a professor in the school of physical and information engineering, Fuzhou University.