Tuning Convolutional Spiking Neural Network With Biologically Plausible Reward Propagation

Tielin Zhang, Shuncheng Jia, Xiang Cheng, and Bo Xu

***Abstract*— Spiking neural networks (SNNs) contain more biologically realistic structures and biologically inspired learning principles than those in standard artificial neural networks (ANNs). SNNs are considered the third generation of ANNs, powerful on the robust computation with a low computational cost. The neurons in SNNs are nondifferential, containing decayed historical states and generating event-based spikes after their states reaching the firing threshold. These dynamic characteristics of SNNs make it difficult to be directly trained with the standard backpropagation (BP), which is also considered not biologically plausible. In this article, a biologically plausible reward propagation (BRP) algorithm is proposed and applied to the SNN architecture with both spiking-convolution (with both 1-D and 2-D convolutional kernels) and full-connection layers. Unlike the standard BP that propagates error signals from postsynaptic to presynaptic neurons layer by layer, the BRP propagates target labels instead of errors directly from the output layer to all prehidden layers. This effort is more consistent with the top-down reward-guiding learning in cortical columns of the neocortex. Synaptic modifications with only local gradient differences are induced with pseudo-BP that might also be replaced with the spike-timing-dependent plasticity (STDP). The performance of the proposed BRP-SNN is further verified on the spatial (including MNIST and Cifar-10) and temporal (including TIDigits and DvsGesture) tasks, where the SNN using BRP has reached a similar accuracy compared to other state-of-the-art (SOTA) BP-based SNNs and saved 50% more computational cost than ANNs. We think that the introduction of biologically plausible learning rules to the training procedure of biologically realistic SNNs will give us more hints and inspiration toward a better understanding of the biological system's intelligent nature.**

Manuscript received August 17, 2020; revised December 29, 2020 and April 9, 2021; accepted May 24, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2020AAA0104305, in part by the National Natural Science Foundation of China under Grant 61806195, in part by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDB32070100 and Grant XDA27010404, and in part by the Beijing Brain Science Project under Grant Z181100001518006. *(Tielin Zhang and Shuncheng Jia are co-first authors.) (Corresponding authors: Tielin Zhang; Bo Xu.)*

Tielin Zhang, Shuncheng Jia, and Xiang Cheng are with the Research Center for Brain-inspired Intelligence, Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), Beijing 100049, China (e-mail: tielin.zhang@ia.ac.cn).

Bo Xu was with the Research Center for Brain-inspired Intelligence, Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China, also with the University of Chinese Academy of Sciences (UCAS), Beijing 100049, China, and also with the Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai 200031, China (e-mail: xubo@ia.ac.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3085966.

Digital Object Identifier 10.1109/TNNLS.2021.3085966

***Index Terms*— Biologically plausible computing, neuronal dynamics, reward propagation, spiking neural network (SNN).**

I. Introduction

THE rapid development of deep learning [or deep neural network (DNN)] breaks many research barriers toward a unified solution by easy and efficient end-to-end network learning. DNNs have replaced many traditional machine learning methods in some specific tasks, and the number of these tasks is still increasing [1]. However, during the rapid expansion process of DNNs, many important and challenging problems are exposed accordingly, such as adversarial network attacking [2], [3], catastrophic forgetting [4], data-hungry [5], lacking causal inference [6], and low transparency [7]. Some researchers try to find answers by making efforts on the inner side research of the DNN itself, such as constructing specific network structures, designing more powerful cost functions, or building finer visualization tools to open the black box of DNNs. These efforts are efficient and have contributed to the further development of DNNs [8]. In this article, we think that there are some alternative and easier approaches to achieving these goals, especially on robust and efficient computation, by turning into the biological neural networks and getting inspirations from them [9]–[17].

The spiking neural network (SNN) is considered the third generation of artificial neural networks (ANNs) [18]. The basic information unit transferred between neurons in SNNs is discrete spikes, containing the membrane potential state's precise timing for reaching the firing threshold. This event-type signal contains inner neuronal dynamics and the historically accumulated (and decayed) membrane potential. Spike trains in SNNs, compared to their counterparts, firerates (here, we define the firerate as an analog value to describe the propagated information) in ANNs, have opened a new time coordinate for the better representation processing of sequential information. Besides neuronal dynamics, biologically featured learning principles are other key characteristics of SNNs, describing the modification of synaptic weights by local and global plasticity principles. For local principles, most of them are "unsupervised," including, but not limited, to the spike-timing-dependent plasticity (STDP) [19], [20], the short-term plasticity (STP) [21], the long-term potentiation (LTP) [22], [23], the long-term depression (LTD) [24], and the lateral inhibition [25], [26]. For global principles, they are more "supervised," with fewer number sizes than local ones, but more related to network functions, e.g., plasticity propagation [27],

reward propagation [28], feedback alignment [29], and target propagation [30].

SNNs are different from structures and functions, such as the echo state machine [31], the liquid state machine [32], the feedforward architecture with biological neurons [12], [33], and some task-related structures [34]–[36]. Some tuning methods of SNNs are BP-based (e.g., ReSuMe [37], SpikeProp [38], and ANNs trained with BP first and then converted into SNNs [13]) or BP-related (BP through time [39], [40], SuperSpike [41], and STDP-type BP [42]). There are still some efforts to train SNNs with biologically plausible plasticity principles (e.g., STDP or STP-based learning [19], [21], [43]–[47], equilibrium learning [48]–[50], multirule-integrative learning [51]–[53], and curiosity-based learning [54]).

This article focuses more on training SNNs with biologically plausible learning principles instead of directly tuning them with BPs to get closer to understanding the brain first and then achieving human-level artificial intelligence. Hence, the SNN with spiking-convolution layers (containing both 1-D and 2-D spiking-convolution kernels) and full-connection layers (containing neuronal dynamics) is constructed and tuned with the proposed biologically plausible reward propagation (BRP), named BRP-SNN. The BRP-SNN focuses on describing the neuron-level dynamic computation of membrane potential and the network-level rewiring of global synapses. Unlike other tuning methods that usually focus more on local plasticity principles (e.g., the STDP and the STP) or mixed local and global principles, we use the global BRP only for simplicity and clarity. After the global reward propagation, synaptic modifications are further induced with only local gradient differences that can also be replaced with the STDP and the differential Hebb's principle.

The main contributions of this article include the following.

1) We construct a multilayer SNN architecture containing LIF neurons and a network with efficient spiking-convolution and spiking-pooling layers.
2) We propose a new BRP for the efficient learning of SNNs. Unlike other BP-based tuning algorithms, the BRP tunes the hidden neurons globally with only labels from the output layer instead of layer-by-layer error backpropagation (BP). This new method has a much lower computational cost for reaching the same performance. Besides, the reward information is more biologically plausible than other error-based signals in the biological system. We think that this is an alternative effort to achieve biologically efficient learning by tuning biologically realistic SNNs with biologically plausible plasticity principles.
3) We use both spatial (including MNIST and Cifar-10) and temporal (including TIDigits and DvsGesture) datasets to test the proposed algorithm's performance on SNNs. The BRP-SNN achieves high performances (accuracy on test sets, 99.01% for the MNIST, 57.08% for the Cifar-10, 94.86% for the TIDigits, and 80.90% for the DvsGesture) compared with that by other state-of-the-art (SOTA) SNNs tuned with pure biologically plausible principles. Furthermore, the BRP-SNN also shows power on a lower computation cost (saving more than 50% neuron computation).

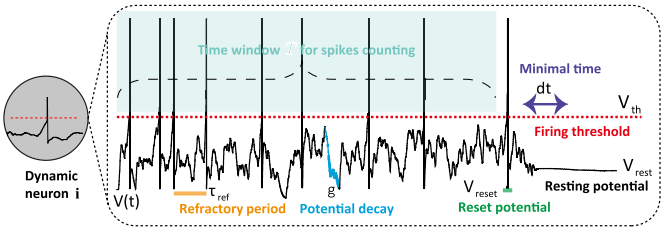

Fig. 1. Example of dynamic LIF neurons in SNNs.

The remainder of the article is organized as follows. Section II gives a brief introduction of the information processing in biological neurons and networks. Section III describes the multilayer SNN with spiking-convolution and spiking-pooling layers, which is then tuned by three types of target propagation methods (including BRP). Section IV introduces the comparable convergence, lower computational cost, and better (or compatible) performance of BRP-SNN compared to other SOTA algorithms tuned with biologically plausible (or BP-based) learning methods. Finally, some conclusions are given in Section V.

II. Background of Biologically Plausible Information Processing

A. Dynamic Spiking Neurons

Dynamic neurons in SNNs are very different from their counterpart activation functions in DNNs. The neurons communicate to each other in different layers of SNNs with discretely event-based spikes but not continuous firerates in DNNs. DNNs also contain various "neurons" (activation functions), such as the rectified linear unit (ReLU), the sigmoid function, and the Tanh function. However, they describe only the spatial nonlinear mapping between neurons' input and output signals instead of temporally wiring them with neuronal dynamics.

Basic computational units in SNNs are different types of dynamic neurons [33], for example, the Hodgkin–Huxley (H-H) neuron, the leaky integrated-and-fire (LIF) neuron, the Izhikevich neuron, and the spike response neuron (SRM). The LIF neuron describes the dynamics of membrane potentials (as shown in Fig. 1). The typical LIF neuron dynamically updates its membrane potential $V_i(t)$ in a long-range time domain according to

$$\begin{cases} C\dfrac{dV_i(t)}{dt} = g(V_i(t) - V_{\text{rest}}) + \displaystyle\sum_{j=1}^{N} W_{i,j}X_j(t) \\ V_i(t) = V_{\text{reset}}, \quad \text{if } (V_i(t) = V_{\text{th}}, t - t_{\text{spike}} > \tau_{\text{ref}}) \end{cases} \tag{1}$$

where τ_{ref} is the refractory period, g is the synaptic conductivity, and V_{reset} is the reset value after $V_i(t)$ reaches firing threshold V_{th}. The resting membrane potential V_{rest} is set as an attractor of $V_i(t)$, especially when no stimulus input is given. C is the membrane capacitance. dt is the minimal time step for $V_i(t)$ update (usually set as 0.1–1 ms). j is the neuron index of presynaptic neuron. i is the current neuron index.

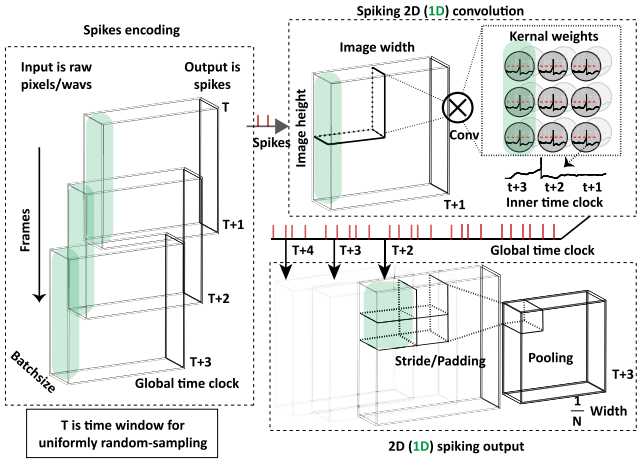

Fig. 2. Multiscale information processing encoded with multiclock spikes, during the 2-D (or 1-D) spiking-convolution and spiking-pooling layers.

N is the number of neurons in the current layer. $W_{i,j}$ is the synaptic weight between the presynaptic neuron j and the current neuron i. t_{spike} is the specific spike time of the neuron i. $X_j(t)$ is the neuronal input from the presynaptic neuron j. All these neuronal dynamics will proceed within a time window T.

B. Biological Plausibility of the BRP

The computation in the biological network is asynchronous, containing obviously multiclocks at different time scales, from synapses and neurons to networks [55], [56]. The biological brain handles this multiscale information processing challenge with event-based spikes containing spatial and temporal signals. These complex dynamics of spikes also make the conventional BP-based algorithms difficult on the network learning.

Fig. 2 shows an example of spikes information processing at convolution and pooling layers, where multiclock spikes (including local time clock inner neurons and global ones between layers) are integrated for the generally complex information encoding.

The biological system will release dopamines from neurons in the basal ganglia directly to other brain circuits (e.g., rerouting by the thalamus) to change the synaptic states. This procedure has inspired us to propose the BRP algorithm, which is efficient and different from the BP in ANNs [46].

III. Method

Here, we add a $Spike$ flag into standard LIF neurons in order to slow down the update speed of $V_i(t)$ instead of absolutely blocking it during the refractory period, shown as follows:

$$\begin{cases} C\dfrac{dV_i(t)}{dt} = g(V_i(t) - V_{\text{rest}})(1 - \text{Spike}) \\ \qquad\qquad + \displaystyle\sum_{j=1}^{N} W_{i,j}X_j(t) \\ V_i(t) = V_{\text{reset}}, \text{Spike} = 1, \quad \text{if } (V_i(t) = V_{\text{th}}) \\ \text{Spike} = 1, \quad \text{if } (t - t_{\text{spike}} < \tau_{\text{ref}}, t \in (1, T)) \end{cases} \tag{2}$$

where $V_i(t)$ is the integration of its historical membrane potential $V_i(t-k)$ $(k \in 1, 2, 3, \ldots)$ with decay g^k and its current input stimulus $X_j(t)$. g (i.e., conductivity) is the decay factor of $V_i(t)$, which is usually designed as a hyperparameter with a value smaller than 1 nS (nanosecond). $Spike$ will be generated if $V_i(t)$ reached the firing threshold V_{th}, and at the same time, $V_i(t)$ will also be reset as a predefined membrane potential V_{reset}. Furthermore, a parameter of refractory time period τ_{ref} is given, during which $V_i(t)$ will have a much lower update speed. V_{rest} can also be considered as one attractor of $V_i(t)$, especially when no input $X_i(t)$ is given and no $Spike$ is generated, where the membrane potential $V_i(t)$ will be dynamically decayed into V_{rest}. The inner iterative time window T for calculating neuronal dynamics is in a range of 10–100 ms.

A. Spiking-Convolution Layer With 1-D and 2-D Kernels

The 1-D and 2-D convolutions have been proved efficient on temporal and spatial information processing, respectively. The reusability of convolutional kernels (i.e., the sharing of synaptic weights) also contributes to network learning toward the antioverfitting characteristic of SNNs to some extent.

As shown in Fig. 2, each layer's signal is represented as spike trains. For simplicity, the time clock t inside the LIF neuron for the membrane potential update is designed as the same as the outside neuron clock T used for the signal propagation from previous to postlayers. This configuration means the time range of information propagation within and outside of neurons is all T for computation ease. The convolutional layer 2-D (or 1-D) spiking input with height and weight dimensions (or with only the height dimension for the 1-D input, i.e., green bars in Fig. 2) makes the convolution with 2-D (or 1-D) kernels. These neurons in kernels are designed as dynamic neurons containing spikes that encode event-based signals with a learning time t going by (until T). Then, the spikes after convolutions further proceed with pooling for the dimension reduction.

The convolutional processing in SNNs is different from that in traditional DNNs. For a clearer description of their differences, a schematic of two types of convolutions is described and shown in Fig. 3. In SNNs, the decay of membrane potentials will make LIF neurons more difficult to fire. On the contrary, membrane potentials' historical positive integrations will make LIF neurons generate relatively more output spikes. These two dynamical parts in the LIF neuron will contribute to the temporal information processing of SNNs at the microscale neuron level. On the contrary, ANNs generate the same firerates as outputs by ignoring the sequential differences of input signals. This sequence-related dynamics of LIF neurons will contribute to the temporal information representation and learning in SNNs.

B. BRP-SNN Architecture

The whole BRP-SNN architecture is shown in Fig. 4, where input spike trains are generated and encoded from the raw signal input, which can be the 1D auditory signal or the 2-D image signal. For 1-D and 2-D signals without spikes

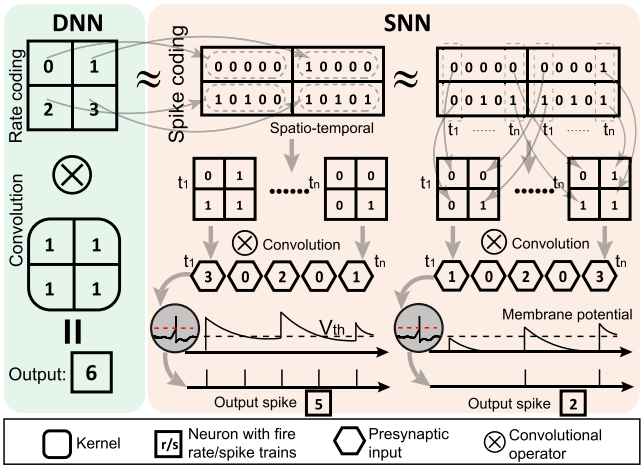

Fig. 3. Schematic depicting different procedures of the standard convolution in DNNs and the spiking convolution in SNNs, where the spiking neuron of SNNs can generate different outputs given input signals with the same firerates (but with different sequence orders).

(e.g., TIDigits and Cifar-10), the raw input signal $I_{\text{raw}}(t)$ is randomly sampled as sequential spike trains $I_{\text{spikes}}(t)$ first and then propagated into the following convolutional, pooling, and full-connection layers. For signals with natural spikes, such as 3-D dynamic-vision-sensor (DVS) video signals (e.g., DvsGesture), the raw signal has already been well encoded as 0 or 1 event-type symbols, which can be considered as spikes directly. Hence, the additional spike encoding is not necessary anymore. The spike generator is shown as follows:

$$I_{\text{spikes}}(t) = \sum_{i=1}^{T} \delta(t - t_i)(I_{\text{raw}}(t) < I_{\text{rd}}(\alpha)) \tag{3}$$

where I_{rd} is a $0-1$ random variable generator with a decay factor α and $I_{\text{rd}}(\alpha) = I_{\text{rd}} * \alpha$. T is the global network clock, which is the same as the inner clock of LIF neurons.

The spiking-convolution and pooling layers are constructed for spatial feature detection, and the feedforward full-connection layers are constructed for the next-step classification. Spike trains of different hidden layers in SNNs can be summed as firerates first after the whole time window T and then combine propagated label-reward signals to modify local synaptic weights.

C. Tuning SNNs With the Global BRP

Feedforward and feedback propagations are usually interleaved together for the convergence learning of SNNs. The feedback propagation is also described as a top-down refinement of the network structure (e.g., synaptic weights) with directly error-to-synapse modification (e.g., revised or pseudo-BPs) [10], [57]. However, these BPs are considered not biologically plausible for their layer-by-layer error BP.

In the biological system, the signal can backpropagate within inner neurons or only within neighborhood layers [10], [58]–[61]. Long-range propagation describes the reward propagation from neurons in the brain region for higher cognitive functions directly to target neurons in that for primary cognitive functions [10]. The idea of a direct random target propagation algorithm [30] fits these biological constraints. Hence, we select it as the underlying bone architecture of the BRP-SNN and further refine it with neuron dynamics, spiking convolutions, and reward propagations.

The biologically plausible reward propagation BRP is constructed according to

$$\text{TP}_{\text{BRP}} = \text{Label}_{\text{spikes}}(t) = \bar{y}(t) \tag{4}$$

describing a spike train to represent target labels (i.e., the one-shot $\text{Label}_{\text{spikes}}(t)$ during T) as the reward to be propagated to all of prehidden layers directly. The TP is a short name of "Target Propagation," which means giving the target goal from the output layer to all of the previous layers directly. Different from TP_{BRP}, another candidate TP_{err} is also constructed, shown as follows:

$$\text{TP}_{\text{err}} = y(t) - \bar{y}(t) \tag{5}$$

where TP_{err} describes the difference between output signals and teaching signals. $\bar{y}(t)$ is the mean firerate of teaching signals (e.g., labels). $y(t)$ is the mean summation of output spike trains from SNNs $[O_{\text{spikes}}(t)]$, shown as follows:

$$\begin{cases} y(t) = \dfrac{1}{T}\displaystyle\sum_{t=1}^{T} O_{\text{spikes}}(t) \\ \bar{y}(t) = \dfrac{1}{T}\displaystyle\sum_{t=1}^{T} \text{Label}_{\text{spikes}}(t) \end{cases} \tag{6}$$

where spikes in the time window T are summed together as the firerate $y(t)$ only at the end of T. TP_{sign} is the third type of TPs, which describes the sign of positive and negative errors, shown as follows:

$$\text{TP}_{\text{sign}} = \text{sign}(y(t) - \bar{y}(t)) \tag{7}$$

where the detailed error signal is not necessary, and only the sign of errors is used for the propagation. This effort propagates only positive ("+") and negative ("−") signals to hidden layers. However, this sign's calculation contains the premise that detailed errors have to be calculated first and then quantitatively normalized as signs of errors. Hence, it still needs both $y(t)$ and $\bar{y}(t)$, the same with TP_{err}, instead of the pure reward definition related only with the functional goal (where $y(t)$ is not necessary) of the network, such as TP_{BRP}. This idea is not new, and some previous work has described the sign instead of error propagation to the network. NormAD [62] is a good example in them, which can be seen as an efficient learning rule by the sign propagation, especially in shallow SNNs.

Different types of TPs will directly propagate to all hidden layers, including spiking-convolution layers and full-connection layers. The additional B_{rand} will be given as a randomly generated matrix during the network initialization, which will play roles in the different-dimensional matrix conversion from the output layer to all hidden layers. Notice that the global propagation can only propagate signals and affect neuron states (e.g., membrane potentials) instead of synaptic weights. Hence, the synaptic modification for the learned knowledge will be further processed in the next-step procedure of local synaptic weight consolidation.

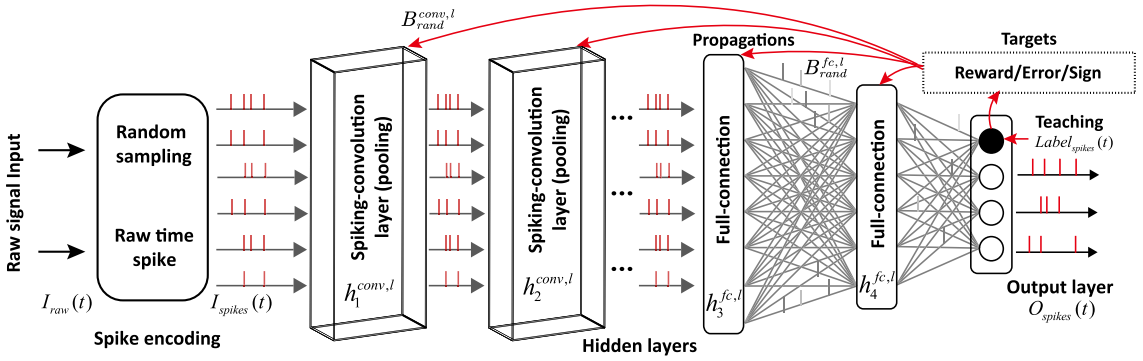

Fig. 4. Architecture of the BRP-SNN.

D. Local Synaptic Weight Consolidation

The BRP will propagate back directly into different hidden layers of SNNs at each time t. This effort will influence the current network states and, at the same time, give SNNs a desired target state for the next-step learning at the time $t+1$. Synaptic weights in SNNs can be updated based on the difference of locally propagated neural states $B_{\text{rand}}^{\text{conv},l} * TP_{\text{BRP}}(t)$ and current neural states $h^{\text{conv},l}(t)$. The modification of synaptic weights $\Delta W_{i,j}^{\text{conv},l}$ in spiking-convolution layers are updated as follows:

$$\Delta W_{i,j}^{\text{conv},l} = -\eta^{\text{conv}}\left(B_{\text{rand}}^{\text{conv},l} * \text{TP}_{\text{BRP}}(t) - h^{\text{conv},l}(t)\right) \tag{8}$$

where $B_{\text{rand}}^{\text{conv},l}$ is a randomly generated matrix for the hidden layer l before network learning and will not be further updated. $\Delta W_{i,j}^{\text{conv},l}$ is the modification of synaptic weights in the convolutional layer l from the presynaptic neuron j to the postsynaptic neuron i. η^{conv} is the learning rate. $h^{\text{conv},l}(t)$ is the neuron state (the same as the membrane potential V) in the current layer l. Similar to that, the modification of $\Delta W_{i,j}^{\text{fc},l}$ in the full-connection layer is updated as follows:

$$\Delta W_{i,j}^{\text{fc},l} = -\eta^{\text{fc}}\left(B_{\text{rand}}^{\text{fc},l} * \text{TP}_{\text{BRP}}(t) - h^{\text{fc},l}(t)\right) \tag{9}$$

where $B_{\text{rand}}^{\text{fc},l}$ is also a randomly generated matrix for the hidden layer l before network learning and will not be further updated during learning. $\Delta W_{i,j}^{\text{fc},l}$ is the synaptic weight in full-connection layer l from the presynaptic neuron j to the postsynaptic neuron i. η^{fc} is the learning rate, and $h^{\text{fc},l}(t)$ is the membrane potential state in the current layer l.

E. Pseudogradient Approximation

The dynamic LIF neuron in SNNs is nondifferential, which is difficult to propagate the gradient for further calculating the gradient from the global BRP to the local weight consolidation. Traditional local tuning of SNNs uses STDP or differential Hebb's principle for the locally synaptic modification [19], [45], [47]. Unlike that, here, we give an additional pseudogradient approximation of LIF neurons to tune SNNs with the Pytorch architecture [63]. The pseudogradient converts the nondifferential procedure of firing neurons (where the membrane potential $V_i(t)$ reaching the firing threshold V_{th} is then reset as V_{reset}) as a specific pseudogradient. The pseudogradient approximation is calculated as follows:

$$\Delta V_i(t) = V_i(t+1) - V_i(t) = 1, \quad \text{if } (V_i(t) = V_{\text{th}}) \tag{10}$$

where $\Delta V_i(t)$ is set as a finite number (here is 1 for simplicity) during the propagation, in order to bypass the nondifferential problem of $V_i(t)$ during the traditional procedure of gradient propagations, where the original $\Delta V_i(t)$ is actually infinite when $V_i(t) = V_{\text{th}}$.

TABLE I
ALGORITHM COMPLEXITY FOR TUNING K-LAYER SNNS

Strategy	Complexity
Pseudo BP	$O(nK + (m + mK)K)$
BRP	$O(nK + (m + 1)K)$
Pure STDP	$O(nK + mK)$

F. Algorithm Complexity Analysis

One advantage of using biologically plausible algorithms to tune K-layer SNNs is the lower algorithm complexity. As shown in Table I, the pure STDP updates synaptic weights directly during the feedforward information propagation. Hence, the algorithm complexity of STDPs is $O(nK + mK)$, where n and m are computational costs of one-step feedforward and one-step local-plasticity propagation, respectively.

The BRP finishes the feedforward procedure with $O(nK)$ and then parallelly propagates reward from the output layer to all hidden layers with an additional one-step matrix conversion ($B_{\text{rand}}^{\text{conv},l}$ and $B_{\text{rand}}^{\text{fc},l}$). Hence, the algorithm complexity of the BRP is $O(nK + (m + 1)K)$. For pseudo-BPs, besides the feedforward cost with $O(nK)$, it also contains multistep backpropagations layer by layer with differential chain rule, costing $O(nK + (m + mK)K)$.

G. BRP-SNN Learning Procedure

The detailed learning procedure of the BRP-SNN algorithm is shown in Algorithm 1.

H. Analysis of Why BRP Works

The proposed BRP is innovative and will be further verified to finish the following spatial and temporal classification tasks in Section IV. The BRP is very different from the traditional gradient-based learning algorithms, which have to calculate

Algorithm 1 BRP-SNN Learning Procedure

1. Start network initialization: Pack raw datasets as input with or without spikes generation by random sampling; Initialize all of hyper parameters related to dynamic neurons C, g, dt, V_{rest}, $W_{i,j}$, V_{reset}, τ_{ref}, and other related parameters of networks, e.g., kernel size, kernel numbers, number of hidden layers and neurons, learning rates η^{conv} and η^{fc}, propagation matrixes of $B_{rand}^{conv,l}$ and $B_{rand}^{fc,l}$ in each layer l, and time window T;
2. Start the training procedure:
(1). Load training samples;
(2). Feedforward information propagation of I_{spikes};
(3). Iteratively increase the network clock t and output O_{spikes} until t reaches the time window T;
(4). Calculate the firerate of network $y(t)$ until T;
(5). Apply BRP, TP_{err} or TP_{sign} on spiking-convolution and full-connection layers for different tasks, respectively;
(6). Local synaptic modifications with pseudo gradients;
(7). Iteratively train BRP-SNN from Step (2)–(7) until convergence; Save the tuned synaptic weights in all layers.
3. Start the testing procedure:
(1). Load test samples;
(2). Test the performance of BRP-SNNs with feedforward propagation based on saved synaptic weights in each layer.
(3). Output the neuron index with a maximum firerate $y(t)$ in time window T as the target class; Calculate the accuracy in the whole test dataset without cross-validation.
4. Finish SNN learning.

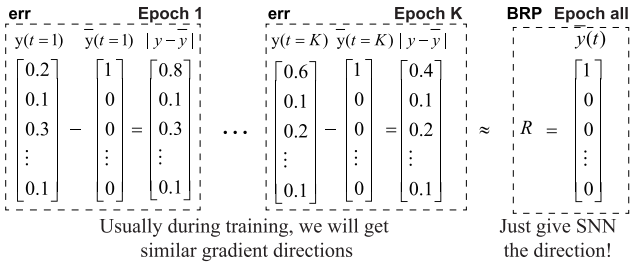

Fig. 5. Comparison of similar gradients between that from the BRP and that from the traditional error BP.

the error (the difference of target $\bar{y}(t)$ and output $y(t)$) first. It is a question of why BRP works. Here, we give an example to explain the plausible reasons intuitively.

As shown in Fig. 5, the errors are usually calculated first for the next-step gradient propagation. During the training procedure, for different epochs, the synaptic weights of SNNs will be tuned with these error-related gradients. However, in the process of learning, the absolute values of errors are more dominated by the $\bar{y}(t)$ instead of the differences of $\bar{y}(t)$ and $y(t)$. Hence, given the BRP, SNN can also get a similar gradient direction. We think that this is the main reason why the BRP works in the training procedure of SNNs.

IV. Experiments

A. Spatial Datasets

The MNIST [64] and Cifar-10 [65] datasets were selected as the two examples of spatial datasets. The MNIST dataset contains 70 000 28 × 28 one-channel gray hand-written digit-number images from 0 to 9, in which 60 000 samples are training samples and the remaining 10 000 samples are testing samples. The Cifar-10 dataset contains 60 000 32 × 32 three-channel color images covering ten classes: 50 000 for training and 10 000 for test. The samples in both of them are static 2-D images, which means that a further spike generation procedure is needed.

TABLE II
BRP-SNN Parameters for Different Tasks

Tasks	Topology	Parameters
MNIST	Cov5*5x28-28-FC1000-FC10	$I_{rd}(\alpha)$=1, η=1e-4
Cifar-10	Cov5*5x32-S2-FC1000-FC10	$I_{rd}(\alpha)$=1, η=1e-4
TIDigits	Cov1*3x100-S1-FC1000-FC10	$I_{rd}(\alpha)$=0.1, η=1e-4
DvsGesture	Cov5*5x32-S2-FC1000-FC10	$I_{rd}(\alpha)$=1, η=1e-4

B. Temporal Datasets

The TIDigits [66] and DvsGesture [56] datasets were selected out as the two examples of temporal datasets. TIDigits contain 4144 (20 kHz and around 1 s for each sample) spoken digits from 0 to 9. The DvsGesture dataset contains 1464 128 × 128 × 1800 gesture samples (low-sampled 32 × 32 size and 100 frames from the raw DVS video with event-based 1800 frames for the ease of computation), covering 11 gesture types. These two datasets are challenging in temporal information processing. For example, the TIDigits dataset contains sequential pronounces of spoken numbers, and the DvsGesture dataset contains sequential videos different from clockwise and counterclockwise arm content rotations. Signals in TIDigits are continuous, which means that a further procedure of spike generation is needed. Signals from the DvsGesture dataset are recorded from the DVS camera, representing natural spikes without an additional spike-conversion procedure.

C. Configuration of Parameters

Parameters of the BRP-SNN for different tasks were different from each other after a little refinement to reach better performances. The refined network topologies are shown in Table II, including the number of spiking-convolution layers and the number of full-connection layers.

These parameters are not carefully designed (e.g., constructing a very deep architecture) for reaching the best performance, just staying startlines of simple shallow architectures for easier comparisons. Both η^{conv} and η^{fc} were set as $1e-4$. For LIF neurons, $C = 1$ uF/cm^2, $g = 0.2$ nS, $\tau_{\text{ref}} = 1$ ms, $T = 10$–100 ms, $V_{\text{th}} = 0.5$ mV, and $V_{\text{reset}} = 0$ mV. The local pseudogradient was calculated with an MSE loss and an Adam optimizer. The batch size for all tasks was normalized as 50.

D. Convergence Learning of BRP-SNNs

The learning convergence is usually the premise of the next-step analysis of efficient learning. As shown in Fig. 6, three types of biologically plausible propagations were tested during the learning procedure of four datasets, including the proposed BRP (TP_{BRP}), the error propagation (TP_{err}), and the sparse error sign propagation (TP_{sign}). The time window T for

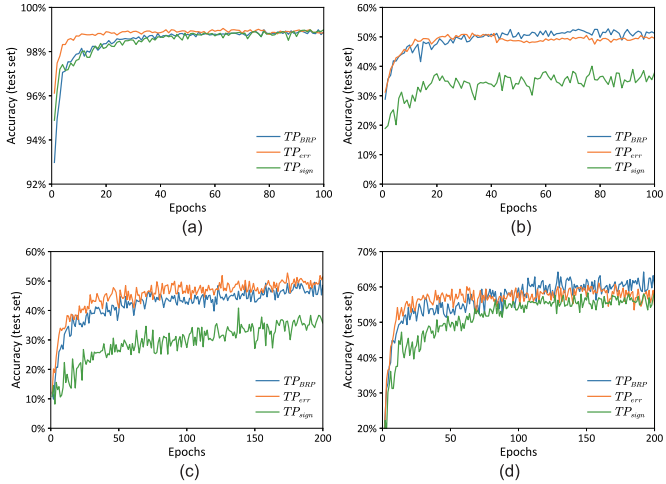

Fig. 6. Network convergence with different TPs on SNNs. (a) MNIST. (b) Cifar-10. (c) TIDigits. (d) DvsGesture.

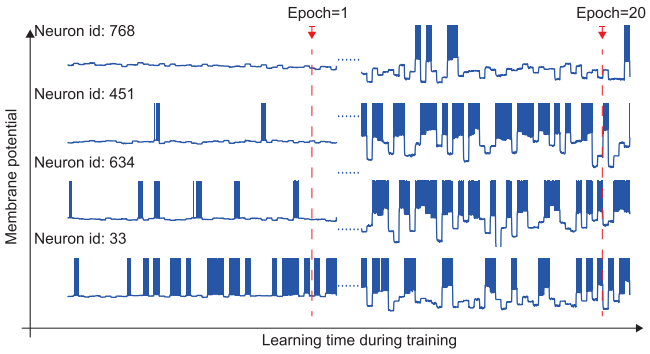

Fig. 7. Membrane potentials with spikes during training of the BRP-SNN.

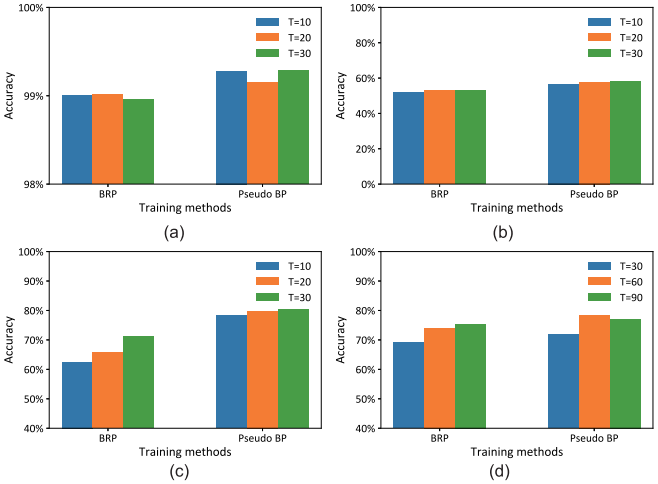

Fig. 8. Accuracy comparisons of spatial and temporal tasks between SNNs using BRPs and that using pseudo-BPs with different T's. Results on (a) MNIST dataset, (b) Cifar-10 dataset, (c) TIDigits dataset, and (d) DvsGesture dataset.

these datasets was predefined as $T = 20$ (for MNIST, Cifar-10, and TIDigits datasets) and $T = 30$ (for the DvsGesture dataset) to get better performances. The learning results on both training and test sets showed that all three biologically plausible propagations were convergent on the four datasets.

For the MNIST dataset, the test accuracy on the SNN using TP_{BRP} was convergent with the increment of training epochs from 0 to 100. SNNs using TP_{BRP} got a convergence rate higher than that using TP_{sign} but lower than that using TP_{err}. The final performances for TP_{BRP}, TP_{err}, and TP_{sign} reached 98.89%, 98.85%, and 98.99%, respectively.

Similar to MNIST, for the Cifar-10 and TIDigits datasets, SNNs using TP_{BRP} were also convergent by reaching convergence rates higher than that using TP_{sign} and comparable to that using TP_{err}. The final test accuracies for SNNs using TP_{BRP}, TP_{err} and TP_{sign} reached 51.21%, 49.48%, and 37.85% on the Cifar-10 dataset, respectively; and they reached 48.55%, 51.45%, and 35.69% on the TIDigits dataset, respectively.

For the DvsGesture dataset, the TP_{BRP} outperformed TP_{err} and TP_{sign} by reaching both higher accuracy and higher convergence rate. The final accuracies for TP_{BRP}, TP_{err}, and TP_{sign} were 60.42%, 54.86%, and 58.33%, respectively.

Furthermore, membrane potentials with spikes were also plotted for some neurons in full-connection layers of the BRP-SNN. The indices of neurons were randomly selected, as shown in Fig. 7. The neural states were dynamically changed during the learning procedure of the BRP-SNN from epoch 1 to epoch 20, toward a better information representation (with more frequent and more stable neuron firings). In order to achieve higher performances, additional experiments were also designed with larger epochs. Finally, SNNs using BRP reached 99.01% (for the MNIST) and 52.56% (for the Cifar-10) after 400 epochs and reached 65.68% (for the TIDigits) and 64.93% (for the DvsGesture) after 10 000 epochs. The high performance on four benchmarks showed the efficiency of the proposed BRP-SNN.

E. BRP Reached Comparable Performance With Pseudo-BP

The inner neuron time window T described the encoding ability of neuronal dynamics, especially for temporal information. As shown in Fig. 8(a) and (b), for spatial datasets, accuracies of SNNs trained with the BRP for different T's (from 10 to 30) were similar (around $98.99 \pm 0.01\%$ for the MNIST and around $52.65 \pm 0.61\%$ for the Cifar-10). One of the main reasons for this result was that the spatial data did not contain any temporal information because the temporal spike trains used for classification were usually generated from the random spike generator. The naturally temporal datasets further verified this hypothesis by SNNs using different T's in Fig. 8(c) and (d), where the accuracies of SNNs on temporal tasks (from 62.46% to 71.14% on the TIDigits and from 69.10% to 76.04% on the DvsGesture) showed a linearly increasing relationship with the size of time window T (from 10 to 30 on the TIDigits and 30 to 90 on the DvsGesture).

Furthermore, the performance of SNNs using BRPs is similar to that using standard pseudo-BPs. The SNNs using BRPs and pseudo-BPs, respectively, reached accuracies of (99.01% and 99.29%) on the MNIST, (53.11% and 57.88%) on the Cifar-10, (71.14% and 80.31% with 1-D kernels and 94.86% and 95.10% with 2-D kernels) on the TIDigits, and (76.04% and 78.21%) on the DvsGesture, respectively. In conclusion, the performance differences were no more than 0.28%, 4.77%,

TABLE III
PERFORMANCE COMPARISONS OF THE PROPOSED BRP-SNN WITH OTHER SOTA ALGORITHMS ON THE FOUR DATASETS

Task	Architecture	Training Type	Learning Rule	Performance
MNIST	Three-layer SNN [48]	Spike-based	Equilibrium learning + STDP	98.52%
	Three-layer SNN [50]	Spike-based	Balanced tuning +STDP	98.64%
	Convolutional-Recurrent SNN [68]	Spike-based	Pseudo BP	99.5%
	Convolutional SNN [69]	Rate-based	Backpropagation	99.1%
	Recurrent SNN [70]	Spike-Rate-based	Pseudo BP	99.6%
	2D-Convolutional SNN (Ours)	**Spike-based**	**BRP**	**99.01%**
Cifar-10	Three-layer SNN [55]	Spike-based	Curiosity+STDP	52.85%
	Two-convolutional spiking layers [41]	Spike-based	Spatio-temporal BP	50.70%
	Deep(9-11 layers) SNN [71]	Spike-based	Direct ANN-to-SNN conversion	91.55%
	2D-Sampling-Conv SNN (Ours)	**Spike-based**	**BRP**	**53.11%**
	2D-Convolutional SNN (Ours)	**Spike-based**	**BRP**	**57.08%**
TIDigits	SOM SNN [72]	Spike-Rate-based	SOM+BP	97.40%
	Liquid State Machine [73]	Spike-based	BP	92.30%
	1D-Sampling-Conv SNN (Ours)	**Spike-based**	**BRP**	**71.14%**
	1D-Convolutional SNN (Ours)	**Spike-based**	**BRP**	**85.85%**
	2D-Convolutional SNN (Ours)	**Spike-based**	**BRP**	**94.86%**
DvsGesture	Deep convolution-recurrent SNN [56]	Spike-based	BPTT	90.28%
	2D-Sampling-Conv SNN (Ours)	**Spike-based**	**BRP**	**76.04%**
	2D-Convolutional SNN (Ours)	**Spike-based**	**BRP**	**80.90%**

0.24%, and 2.17% on the MNIST, the Cifar-10, the TIDigits, and the DvsGesture datasets, respectively.

F. Comparison of BRP-SNNs With Other SOTA Algorithms

We further tested the performance of our BRP-SNN with other SOTA algorithms, including spike-based and rate-based SNNs. Some nonbiologically plausible algorithms were also listed for a better comparison. As shown in Table III, for the MNIST task, the best accuracy of SNNs was trained with BP, where signals were represented with both firerates and spikes, and contained convolution and recurrent loops (99.6%). Our BRP-SNN algorithm reached an accuracy of 99.01% higher than other SNNs tuned with STDPs.

For the Cifar-10 dataset, we only made a further comparison with [54] for its biologically plausible learning principles (with curiosity and the STDP). Our BRP-SNN reached 53.11% accuracy, which is higher than the curiosity-based SNN (52.85%).

For the TIDigits dataset, we got 71.14% and 94.86% accuracies on 1-D and 2-D convolutional SNNs, respectively. For the liquid state machine SNN, the accuracy only reached 92.30%. For the integration of ANNs (e.g., self-organizing map and SOM) and SNNs, the accuracy was higher than ours and reached 97.40%. This effort showed the power of the hybrid architecture by the integration of both SNNs and ANNs.

For the DvsGesture dataset, our 2-D convolutional SNN reached 76.04% accuracy with random sampling and 80.90% without that, which, as far as we know, was the first time to use a biologically plausible learning principle to train spike-based SNNs without any preprocessing and cooperation with other methods. These results showed the efficiency of the proposed BRP-SNN.

G. Low Computation Cost of the BRP-SNN With Silent Neurons

In ANNs, some signals with minimal values are surprisingly important. For example, the BP gradient values are smaller than $1e-6$ (still play roles on synaptic modifications) and the neuron with small output firerates but still relatively higher than that of other output neurons as the target output class. Hence, ANNs have to take more computational resources to deal with these special situations.

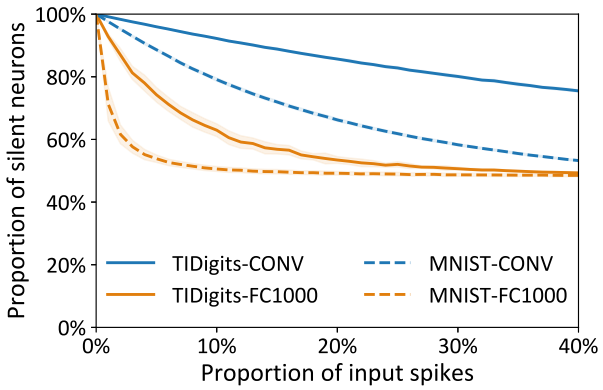

Fig. 9. Proportions of silent neurons in different hidden layers of SNNs (trained on the MNIST and the TIDigits datasets), with proportional increasements of input spikes in the first layer.

In SNNs, the basic unit of signals is a discontinuous spike. The neurons that have not received any spikes or received but still not reached the firing threshold could be considered silent neurons. These silent neurons will take nearly no computational cost on specially designed neuromorphic chips [73]. Hence, the number of these silent neurons in different layers of SNNs during network learning can be a good indicator for the computational cost, combined with the network's algorithm complexity for reaching the same accuracy.

As shown in Fig. 9, the MNIST and TIDigits datasets were selected out as two representatives of spatial and temporal tasks, respectively. The proportion values of silent neurons were all decreased with the proportional increment of input spikes (from 0% to 40%) in different hidden layers of SNNs trained with the MNIST and TIDigits datasets. More than half of the neurons were silent for spiking-convolution (CONV) and full-connection (FC1000 with 1000 hidden neurons) layers. These proportion values were reduced when a more significant proportion value of input spikes was given.

For example, when the proportion of input spikes reached 40%, the proportion values of silent neurons reached 53.26% (CONV) and 48.52% (FC1000) on the spatial MNIST dataset, which were much smaller (means the less computational cost) than that on the temporal TIDigits dataset, and reached 75.50% (CONV) and 49.32% (FC1000).

Hence, SNNs, to this extent, at least saved more than 50% computational cost, on the perspective of a relatively much smaller number of computing neurons (more silent neurons) than in ANNs. Besides, a lower computational cost might be more likely on temporal tasks than on spatial tasks.

V. Conclusion

During evolution, the biological brain is increasingly efficient in representing and processing complex signals with spikes, delicately designed network structures, and efficient biologically plausible plasticity principles. SNNs are from biological networks' inspiration, containing discontinuous spikes instead of continuous firerates for the information propagation. These seem-simple spikes are not trivial but have played essential roles in the computation of sparse (or event-based) signals. They also reflect the complex inner computation of neuronal dynamics and the balance between higher accuracy and lower computational cost. SNNs also contain various types of plasticity principles, including local plasticity principles (e.g., STDP, STP, Hebb, and lateral inhibition) and global plasticity principles (e.g., dopamine-like reward that propagates directly along with top-down connections from reward neurons to target neurons, without any multistep routing).

In this article, we focus on both spikes from neuronal dynamics and biologically plausible learning principles for the good tuning of SNNs, not for the highest accuracy, but an alternative effort to understand a possibly efficient tuning strategy better in the real biological brain. Inspired by the brain's direct reward signal propagation, we have designed the SNN using biologically plausible reward propagation (BRP-SNN) to directly propagate the right label index (with the form of label-spike trains) to target neurons. Spiking-convolution and full-connection layers with dynamic LIF neurons have been constructed for the BRP-SNN and verified on four benchmark datasets, including spatial datasets (the MNIST and the Cifar-10) and also temporal datasets (the TIDigits and the DvsGesture). The BRP-SNN has got comparable performances compared to that tuned with SOTA pseudo-BPs under the same SNN architectures, reaching testing accuracy with 99.01% for the MNIST, 57.08% for the Cifar-10, 94.86% for the TIDigits, and 80.90% for the DvsGesture, respectively. Besides, further computation-efficiency analysis has been given, showing that the BRP-SNN has saved at least 50% computational cost on both full-connection layers and spiking-convolution layers compared to that in standard ANNs.

The standard BP in ANNs backpropagates error signals from the final output neurons to previous neurons layer by layer, which is not biologically plausible and not energy-efficient. We think that the deeper research on biologically plausible plasticity principles (e.g., BRP or other local plasticity principles, such as STDP and STP) will someday finally replace the BP, toward human-level robust computation, energy-efficient learning, or even artificial general intelligence that the community of artificial intelligence is facing but still has not been resolved. We think that this research will also give us more hints on a better understanding of the biological system's intelligent nature and promote a better study of globally neural plasticity in biological networks.

Acknowledgment

The authors would like to thank Ruichen Zuo and Kaiyuan Liu for their assistance with the final version article checking. The source code of BRP-SNN is in https://github.com/thomasaimondy/BRP-SNN.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[2] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 427–436.

[3] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 2180–2188.

[4] K. James *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.

[5] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, Dec. 2015.

[6] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behav. Brain Sci.*, vol. 40, pp. 1–101, Jan. 2017.

[7] A. Mahendran and A. Vedaldi, "Visualizing deep convolutional neural networks using natural pre-images," *Int. J. Comput. Vis.*, vol. 120, no. 3, pp. 233–255, Dec. 2016.

[8] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3859–3869.

[9] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, "Neuroscience-inspired artificial intelligence," *Neuron*, vol. 95, no. 2, pp. 245–258, Jul. 2017.

[10] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton, "Backpropagation and the brain," *Nature Rev. Neurosci.*, vol. 21, no. 6, pp. 1–12, 2020.

[11] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Netw.*, vol. 111, pp. 47–63, Mar. 2019.

[12] S. Woźniak, A. Pantazi, T. Bohnstingl, and E. Eleftheriou, "Deep learning incorporating biologically inspired neural dynamics and in-memory computing," *Nature Mach. Intell.*, vol. 2, no. 6, pp. 325–336, Jun. 2020.

[13] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers Neurosci.*, vol. 10, p. 508, Nov. 2016.

[14] A. H. Marblestone, G. Wayne, and K. P. Kording, "Toward an integration of deep learning and neuroscience," *Frontiers Comput. Neurosci.*, vol. 10, p. 94, Sep. 2016.

[15] M. Matsugu, K. Mori, M. Ishii, and Y. Mitarai, "Convolutional spiking neural network model for robust face detection," in *Proc. 9th Int. Conf. Neural Inf. Process. (ICONIP)*, 2002, pp. 660–664.

[16] G. Zeng, Y. Chen, B. Cui, and S. Yu, "Continual learning of context-dependent processing in neural networks," *Nature Mach. Intell.*, vol. 1, no. 8, pp. 364–372, Aug. 2019.

[17] Z.-H. Zhou and J. Feng, "Deep forest," 2017, *arXiv:1702.08835*. [Online]. Available: http://arxiv.org/abs/1702.08835

[18] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Netw.*, vol. 10, no. 9, pp. 1659–1671, Dec. 1997.

[19] Y. Bengio, T. Mesnard, A. Fischer, S. Zhang, and Y. Wu, "STDP-compatible approximation of backpropagation in an energy-based model," *Neural Comput.*, vol. 29, no. 3, pp. 555–577, Mar. 2017.

[20] G.-Q. Bi and M.-M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," *J. Neurosci.*, vol. 18, no. 24, pp. 10464–10472, Dec. 1998.

[21] R. S. Zucker, "Short-term synaptic plasticity," *Annu. Rev. Neurosci.*, vol. 12, no. 1, pp. 13–31, 1989.

[22] T. J. Teyler and P. DiScenna, "Long-term potentiation," *Annu. Rev. Neurosci.*, vol. 10, no. 1, pp. 131–161, 1987.

[23] M. Zhang, H. Qu, X. Xie, and J. Kurths, "Supervised learning in spiking neural networks with noise-threshold," *Neurocomputing*, vol. 219, pp. 333–349, Jan. 2017.

[24] M. Ito, "Long-term depression," *Annu. Rev. Neurosci.*, vol. 12, no. 1, pp. 85–102, 1989.

[25] L. F. Abbott and S. B. Nelson, "Synaptic plasticity: Taming the beast," *Nature Neurosci.*, vol. 3, no. 11, p. 1178, 2000.

[26] C. Blakemore and E. Tobin, "Lateral inhibition between orientation detectors in the cat's visual cortex," *Exp. Brain Res.*, vol. 15, no. 4, pp. 439–440, Sep. 1972.

[27] G.-Q. Bi and M.-M. Poo, "Synaptic modification by correlated activity: Hebb's postulate revisited," *Annu. Rev. Neurosci.*, vol. 24, no. 1, pp. 139–166, Mar. 2001.

[28] E. J. Nestler and W. A. Carlezon, "The mesolimbic dopamine reward circuit in depression," *Biol. Psychiatry*, vol. 59, no. 12, pp. 1151–1159, Jun. 2006.

[29] D. Zhao, Y. Zeng, T. Zhang, M. Shi, and F. Zhao, "GLSNN: A multi-layer spiking neural network based on global feedback alignment and local STDP plasticity," *Frontiers Comput. Neurosci.*, vol. 14, p. 101, Nov. 2020.

[30] F. Charlotte, L. Martin, and B. David, "Learning without feedback: Fixed random learning signals allow for feedforward training of deep neural networks," *Front. Neurosci.*, vol. 15, pp. 1–20, Feb. 2021, Art. no. 629892.

[31] M. Lukoševičius, H. Jaeger, and B. Schrauwen, "Reservoir computing trends," *KI-Künstliche Intelligenz*, vol. 26, no. 4, pp. 365–371, Nov. 2012.

[32] H. Jaeger, W. Maass, and J. Principe, "Special issue on echo state networks and liquid state machines," *Neural Netw.*, vol. 20, no. 3, pp. 287–289, Apr. 2007.

[33] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569–1572, Nov. 2003.

[34] L. F. Abbott, B. DePasquale, and R.-M. Memmesheimer, "Building functional networks of spiking model neurons," *Nature Neurosci.*, vol. 19, no. 3, p. 350, 2016.

[35] Y. Cao, Y. Chen, and D. Khosla, "Spiking deep convolutional neural networks for energy-efficient object recognition," *Int. J. Comput. Vis.*, vol. 113, no. 1, pp. 54–66, May 2015.

[36] F. Zenke, E. J. Agnes, and W. Gerstner, "Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks," *Nature Commun.*, vol. 6, no. 1, pp. 1–13, Nov. 2015.

[37] F. Ponulak, "Analysis of the ReSuMe learning process for spiking neural networks," *Int. J. Appl. Math. Comput. Sci.*, vol. 18, no. 2, pp. 117–127, Jun. 2008.

[38] S. M. Bohte, J. N. Kok, and J. A. Poutre, "SpikeProp: Backpropagation for networks of spiking neurons," in *Proc. Eur. Symp. Artif. Neural Netw.*, 2000, pp. 419–424.

[39] G. Bellec, F. Scherr, E. Hajek, D. Salaj, R. Legenstein, and W. Maass, "Biologically inspired alternatives to backpropagation through time for learning in recurrent neural nets," 2019, *arXiv:1901.09049*. [Online]. Available: http://arxiv.org/abs/1901.09049

[40] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *Frontiers Neurosci.*, vol. 12, p. 331, May 2018.

[41] F. Zenke and S. Ganguli, "SuperSpike: Supervised learning in multilayer spiking neural networks," *Neural Comput.*, vol. 30, no. 6, pp. 1514–1541, Jun. 2018.

[42] A. Tavanaei and A. Maida, "BP-STDP: Approximating backpropagation using spike timing dependent plasticity," *Neurocomputing*, vol. 330, pp. 39–47, Feb. 2019.

[43] Y. Dan and M.-M. Poo, "Spike timing-dependent plasticity of neural circuits," *Neuron*, vol. 44, no. 1, pp. 23–30, Sep. 2004.

[44] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers Comput. Neurosci.*, vol. 9, p. 99, Aug. 2015.

[45] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "STDP-based spiking deep convolutional neural networks for object recognition," *Neural Netw.*, vol. 99, pp. 56–67, Mar. 2018.

[46] M. Mozafari, M. Ganjtabesh, A. Nowzari-Dalini, S. J. Thorpe, and T. Masquelier, "Bio-inspired digit recognition using reward-modulated spike-timing-dependent plasticity in deep convolutional networks," *Pattern Recognit.*, vol. 94, pp. 87–95, Oct. 2019.

[47] T. Zhang, Y. Zeng, M. Shi, and D. Zhao, "A plasticity-centric approach to train the non-differential spiking neural networks," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 620–628.

[48] B. Scellier and Y. Bengio, "Equilibrium propagation: Bridging the gap between energy-based models and backpropagation," *Frontiers Comput. Neurosci.*, vol. 11, p. 24, May 2017.

[49] T. Zhang, Y. Zeng, D. Zhao, and B. Xu, "Brain-inspired balanced tuning for spiking neural networks," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 1653–1659.

[50] B. Scellier and Y. Bengio, "Equilibrium propagation: Bridging the gap between energy-based models and backpropagation," 2016, *arXiv:1602.05179*. [Online]. Available: http://arxiv.org/abs/1602.05179

[51] Y. Zeng, T. Zhang, and B. Xu, "Improving multi-layer spiking neural networks by incorporating brain-inspired rules," *Sci. China Inf. Sci.*, vol. 60, no. 5, May 2017, Art. no. 052201.

[52] T. Zhang, Y. Zeng, D. Zhao, L. Wang, Y. Zhao, and B. Xu, "HMSNN: Hippocampus inspired memory spiking neural network," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2016, pp. 002301–002306.

[53] R. C. O'Reilly, "Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm," *Neural Comput.*, vol. 8, no. 5, pp. 895–938, Jul. 1996.

[54] M. Shi, T. Zhang, and Y. Zeng, "A curiosity-based learning method for spiking neural networks," *Frontiers Comput. Neurosci.*, vol. 14, p. 7, Feb. 2020.

[55] Y. Xing, G. Di Caterina, and J. Soraghan, "A new spiking convolutional recurrent neural network (SCRNN) with applications to event-based hand gesture recognition," *Frontiers Neurosci.*, vol. 14, p. 1143, Nov. 2020.

[56] A. Amir *et al.*, "A low power, fully event-based gesture recognition system," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7243–7252.

[57] S. Yin *et al.*, "Algorithm and hardware design of discrete-time spiking neural networks based on back propagation with binary activations," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2017, pp. 1–5.

[58] S. M. Bohte, J. N. Kok, and H. L. Poutré, "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, nos. 1–4, pp. 17–37, Oct. 2002.

[59] R. M. Fitzsimonds, H. J. Song, and M. M. Poo, "Propagation of activity-dependent synaptic depression in simple neural networks," *Nature*, vol. 388, no. 6641, pp. 439–448, 1997.

[60] M. Hausser, "Diversity and dynamics of dendritic signaling," *Science*, vol. 290, no. 5492, pp. 739–744, Oct. 2000.

[61] K. Svoboda, F. Helmchen, W. Denk, and D. W. Tank, "Spread of dendritic excitation in layer 2/3 pyramidal neurons in rat barrel cortex *in vivo*," *Nature Neurosci.*, vol. 2, no. 1, pp. 65–73, Jan. 1999.

[62] N. Anwani and B. Rajendran, "NormAD–normalized approximate descent based supervised learning rule for spiking neurons," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–8.

[63] A. Paszke *et al.*, "Automatic differentiation in PyTorch," in *Proc. Workshop 31st Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1–4.

[64] Y. LeCun. (1998). *The Mnist Database of Handwritten Digits*. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[65] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech Rep., 2009.

[66] R. G. Leonard and G. Doddington, "TIDIGITS LDC93S10," Linguistic Data Consortium, Philadelphia, PA, USA, Tech. Rep., 1993, doi: 10.35111/72xz-6x59.

[67] X. Cheng, Y. Hao, J. Xu, and B. Xu, "LISNN: Improving spiking neural networks with lateral interactions for robust object recognition," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, C. Bessiere, Ed., Jul. 2020, pp. 1519–1525.

[68] 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 *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–8.

[69] W. Zhang and P. Li, "Spike-train level backpropagation for training deep recurrent spiking neural networks," in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 7802–7813.

[70] C. Lee, S. S. Sarwar, P. Panda, G. Srinivasan, and K. Roy, "Enabling spike-based backpropagation for training deep neural network architectures," *Frontiers Neurosci.*, vol. 14, p. 119, Feb. 2020.

[71] J. Wu, Y. Chua, M. Zhang, H. Li, and K. C. Tan, "A spiking neural network framework for robust sound classification," *Frontiers Neurosci.*, vol. 12, pp. 1–22, Nov. 2018.
[72] Y. Zhang, P. Li, Y. Jin, and Y. Choe, "A digital liquid state machine with biologically inspired learning and its application to speech recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2635–2649, Nov. 2015.
[73] M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018.

Tielin Zhang received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2016.

He is currently an Associate Professor with the Research Center for Brain-Inspired Intelligence, Institute of Automation, Chinese Academy of Sciences, Beijing. His current research interests include theoretical research on neural dynamics and spiking neural networks.

Shuncheng Jia is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences, Beijing, China.

He is also working with the Institute of Automation, Chinese Academy of Sciences, Beijing. His current research interests include theoretical research on neural dynamics, auditory signal processing, and spiking neural networks.

Xiang Cheng is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences, Beijing, China.

He is also working with the Institute of Automation, Chinese Academy of Sciences, Beijing. His current research interests include biological informatics, neural dynamics, spatiotemporal signal processing, and spiking neural networks.

Bo Xu is currently a Professor, the Director of the Institute of Automation, Chinese Academy of Sciences, Beijing, China, and the Deputy Director of the Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Beijing. His research interests include brain-inspired intelligence, brain-inspired cognitive models, natural language processing and understanding, and brain-inspired robotics.