Self-Lateral Propagation Elevates Synaptic Modifications in Spiking Neural Networks for the Efficient Spatial and Temporal Classification

Tielin Zhang, Qingyu Wang, and Bo Xu

***Abstract*—The brain's mystery for efficient and intelligent computation hides in the neuronal encoding, functional circuits, and plasticity principles in natural neural networks. However, many plasticity principles have not been fully incorporated into artificial or spiking neural networks (SNNs). Here we report that incorporating a novel feature of synaptic plasticity found in natural networks, whereby synaptic modifications self-propagate to nearby synapses, named self-lateral propagation (SLP), could further improve the accuracy of SNNs in three benchmark spatial and temporal classification tasks. The SLP contains lateral pre (SLP-pre) and lateral post (SLP-post) synaptic propagation, describing the spread of synaptic modifications among output synapses made by axon collaterals or among converging synapses on the postsynaptic neuron, respectively. The SLP is biologically plausible and can lead to a coordinated synaptic modification within layers that endow higher efficiency without losing much accuracy. Furthermore, the experimental results showed the impressive role of SLP in sharpening the normal distribution of synaptic weights and broadening the more-uniform distribution of misclassified samples, which are both considered essential for understanding the learning convergence and network generalization of neural networks.**

***Index Terms*—Spiking Neural Network, Synaptic Plasticity, Self Lateral Propagation, Spatial Classification, Temporal Classification.**

I. Introduction

In artificial neural networks (ANNs), the idea that the synaptic modification should be tuned to achieve the correct output has led to the development of a computing algorithm for supervised learning known as backpropagation (BP) [1]. The BP, first invented in the 1980s [2], describes that errors in the neuronal outputs of the ANN output layer concerning the expected values are used to adjust the synaptic weights of upstream synapses layer-by-layer until the output meets the expectation.

However, BP usually assumes non-spiking units with continuous activation functions, limited time duration for simulation, and total supervised error for generating real-time global credit assignments. These constraints are all in conflict with the spirit of biological plausibility. It seems that the vanilla BP is not a proper rule for directly training biological neural networks, e.g., spiking neural networks (SNNs).

SNNs are considered the third-generation of ANNs [3], whereby more biologically plausible features are borrowed from biological networks, including but not limited to efficient encoding, network architectures, learning principles, and application paradigms. For the information encoding, SNNs use discontinuous spikes to convey information between neurons, containing both spatial (e.g., fire rates for the robust computation) and temporal (e.g., spike timing for efficient computation) features [4], [5]. For the network architecture, on the one hand, SNNs use the conventional structures from ANNs, such as feedforward, convolutional, and recurrent types; on the other hand, SNNs also make use of some specifically designed structures, such as synaptic connections with hierarchical time delay and various short-term dynamics. For the application paradigms, SNNs have shown much power in the unsupervised classification [6], continuous learning [7], and reinforcement learning [8].

For the learning principle, SNNs can incorporate both biological plasticity-based (which can be called plasticity-based for simplification in this work) rules and artificial gradient-based rules. Substantial plasticity-based principles have been proposed in guiding synaptic modifications of SNNs, including Hebb's rule [9], [10], short-term plasticity (STP) [11], [12], long-term potentiation (LTP) [13], [14], long-term depression (LTD) [15], and spike-timing-dependent plasticity (STDP) [16], [17]. Many gradient-based principles have been incorporated to train SNNs, whereby most of them are revised versions of conventional BP rules, such as spike-based gradient [18], [19] and reward propagation [20], and potential-based supervised learning [21]. There are also some spike-based local learning rules incorporating gradients with temporal dependencies [22]. Besides, it is a smart and efficient way to integrate both plasticity and gradient-based algorithms toward some new hybrid learning algorithms [23].

The plasticity-based rules contain more biologically plausible features of natural neural networks, which might give us more hints and inspirations on designing new learning algorithms with more local and unsupervised features there instead of in a BP-like global and supervised manner. However, most plasticity-based rules found in natural neural networks are local types, making them lack a global picture during learning. Are there any biologically plausible plasticity principles

This work was supported in part by the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB32070100 and XDA27010404), in part by the Shanghai Municipal Science and Technology Major Project (Grant No. 2021SHZDZX), and in part by the Youth Innovation Promotion Association of CAS. Tielin Zhang and Qingyu Wang are with the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China, and the School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), Beijing 100049, China. Bo Xu is with CASIA, UCAS, and the Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai 200031, China. The corresponding authors are Tielin Zhang (tielin.zhang@ia.ac.cn) and Bo Xu (xubo@ia.ac.cn). Tielin Zhang and Qingyu Wang are co-first authors.

founded in natural neural networks and closely relevant to the spirit of global BP? Looking back and reviewing biological discoveries, we might give a clear and positive answer.

It has been identified that the backpropagation of current signals inner a neuron exists in biological experiments [24], [25], describing that when dendrites of pyramidal cells are active, the action potential could backpropagate from axons to dendrites [26]. Spontaneous backpropagating action potentials have also been found in dendrites of pyramidal neurons in layer $2/3$ of barrel cortex [27]. Furthermore, backpropagation of action potentials to the dendritic arbor has been identified as being actively supported by $Na+$ channels, and this dendritic super linearity enables neurons in layer 2/3 to integrate ascending sensory input from layer four and associative input to layer one [28].

In this paper, we focus more on the propagation of synaptic plasticity in a self-organized manner. We here name it self-propagation for simplicity. The self-propagation phenomena have been widely discovered in cultures of hippocampal neurons [29]–[31] and retinotectal circuits in vivo [32]. The self-propagation consists of three forms of LTP/LTD propagation: The cross-layer self-backpropagation (SBP) of LTP and LTD from output synapses to input synapses of a neuron (Fig. 1A), which has already been discussed in our previous paper [33]; The presynaptic self-lateral propagation (SLP_{pre}) of LTP and LTD among synapses made by axon collaterals of the same presynaptic neuron (Fig. 1B); The postsynaptic lateral propagation (SLP_{post}) of LTD among converging synapses on the postsynaptic neuron (Fig. 1C). These phenomena represent non-local effects of activity-dependent synaptic plasticity that endow coordinated changes in synaptic connections within the neural network. The contributions of this paper can be concluded as the following parts:

- While the implication of SLP in the development and function of natural neural circuits remains to be fully understood, we extended our previous research of SBP [33] by providing two additional new pathways of STDP propagation, which are biologically plausible and complete the research theory of mesoscale plasticity propagation. The experimental results show the effectiveness of better accuracy for SLP-improved SNNs, e.g., improving 3.8% and 1.1% accuracy on the MNIST and DvsGesture datasets, respectively, than those using pure STDP-based algorithms. Deep SNNs using SLP can finally achieve 98.91% accuracy on MNIST, 83.5% on DvsGesture, and 81.48% on N-MNIST. These results underscore the usefulness of introducing new plasticity rules found in the natural neural network into SNNs.
- The experimental results showed the impressive role of SLP in sharpening the normal distribution of synaptic weights and broadening the more-uniform distribution of misclassified samples, which are key for better understanding the learning convergence and network generalization.
- The computational efficiency of SNNs using SLP has been verified and compared with other algorithms. The results indicate that the SNN using SLP can reduce the computational cost in the training period.

II. Related works

Many learning algorithms have been proposed for training SNNs, including ANN-to-SNN conversion, plasticity-based learning, and gradient-based learning.

The ANN-to-SNN conversion has been widely used for reaching the comparable performance of ANNs. During conversion, the necessary processing steps are stochastic gradient descent [34], [35], rectified linear unit (ReLU) [36], weight constraint, or normalization in an approximate linear space [37]. In some cases, the non-differential spikes and discrete synapses are treated as continuous variables, which makes it possible to train SNNs using standard BP [38], [39] or non-differential versions of BP [40], [41]. Even though this approach is more accessible and efficient than tuning SNNs directly, they still cannot help us understand the intelligent nature of biological plasticity, let alone inspire new learning rules.

The plasticity-based learning algorithms use biologically plausible plasticity principles, including STDP [24], [37], [42], [43], lateral inhibition [10], STP [12], reward propagation [20] for efficient and unsupervised synaptic modifications. The SBP [33] has been deeply examined in both SNNs and ANNs for improving computational efficiency, which showed the usefulness of self-propagation in the direction of backpropagation.

The gradient-based learning contains spike-timing-based and spike-rate-based algorithms. SpikeProp [44] and first-spike encoding [45] incorporate spike timing instead of fire rate as the leading information indicators and tune SNNs with pure BP through time. However, each neuron in this paradigm could only fire a single spike by limiting gradient computation at the temporal scale since there will be no gradients without spikes. E-prop [8] is a standard spike-rate hybrid algorithm that incorporates well global target propagation (a revised version of BP, some other candidate revisions could be feedback alignment [30] or reward propagation [46]) and local eligibility trace for the efficient SNN learning.

III. Methods

A. The propagation of synaptic plasticity

As shown in Fig. 1, when an LTP ("+") or LTD ("-") is inducted on a specific synapse ("dashed circuits"), three directions of plasticity propagation will occur, including the backpropagation (SBP, Fig. 1A, which has been discussed in our previous research [33] and will not be extended here), pre-synaptic lateral propagation (SLP_{pre}, Fig. 1B), and post-synaptic lateral propagation (SLP_{post}, Fig. 1C). It is an interesting phenomenon that both LTP and LTD could be propagated in SBP and SLP_{pre}, but only LTD could be propagated in SLP_{post}. These three types of propagations are all self-organized, which means the propagation of plasticity will not acquire the additional supervised signal. The initial induction of LTP or LTD can be caused by different synaptic modification algorithms, including but not limited to the STDP, Hebb, or other subversions of gradient-based BP or target propagation (TP) [47]. However, the propagated plasticity will be stopped after calculation and will not cause further propagated plasticity.

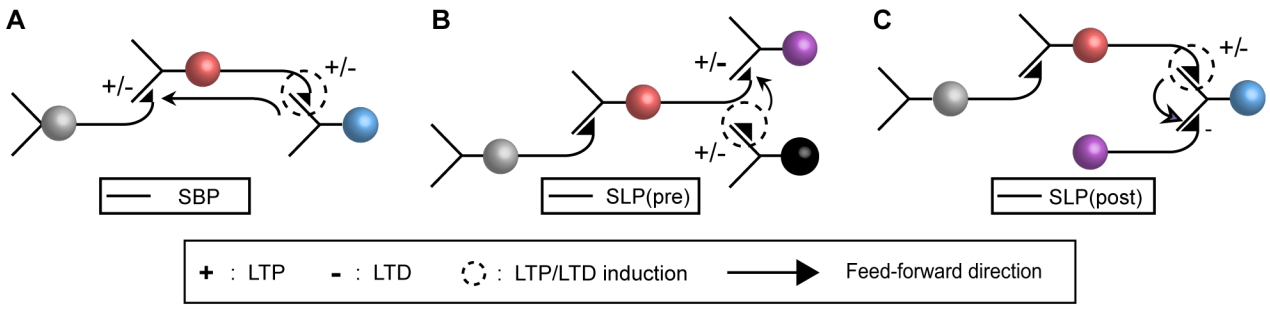

Fig. 1: Schematic diagrams depicting the three types of self-organized plasticity propagation. **A**, The self-backpropagation (SBP) [33] is caused by an initial plasticity induction (dashed circuits), including long-term potentiation (LTP, "+") or long-term depression (LTD, "-") between the presynaptic neuron (red) to the postsynaptic neuron (blue). The spread of plasticity contains both "+" and "-" from the post to pre-synapses. **B**, Same as that in A but for the lateral presynaptic propagation (SLP_{pre}), where the plasticity is laterally propagated among synapses made by axon collaterals of the presynaptic neuron (red), with both "+" and "-" plasticity. **C**, Same as that in A but for the lateral postsynaptic propagation (SLP_{post}), where the plasticity is propagated to converging synapses on the postsynaptic neuron (blue), with only "-" plasticity.

B. The basic SNN architecture

Here we construct a three-layer SNN (Fig. 2A) to examine the role of SLP on network learning. The classification accuracy will be calculated after testing the trained network on the benchmark datasets, further discussed in the experimental results section. In the first layer, neurons receive spike trains converted from the raw data set for the network architecture. Random sampling is used for the signal conversion from raw pixel information to spike trains (see Section of experimental results for more details). The second (hidden) layer consists of excitatory and inhibitory leaky integrate-and-fire (LIF) neurons exhibiting refractory period, non-linear integration, non-differentiable membrane potential, and probabilistic firing. The third (output) layer is supported by all excitatory LIF neurons receiving spiking signals from hidden-layer neurons and additional teaching signals.

The learning process employed both local (STDP) and non-local (SLP) forms of synaptic modifications and consisted of several sequential steps (Fig. 2C,D). First, feedforward processing of spiking signals was performed without introducing synaptic plasticity (Fig. 2C). Second, a homeostatic adjustment of membrane potential (ΔV_m) in the hidden layer neurons was performed to help maintain a stable membrane potential that allowed stable spiking capability of the network. Third, the symmetric STDP approach was induced, with LTP (labeled as "+") or LTD (labeled as "-") of synaptic weights ($W_{j,k}$) occurring at synapses made by the hidden neurons onto the output neurons (Fig. 2D). Finally, these synaptic modifications are allowed to propagate by SLP rules. For enhancing the specificity in synaptic modification via SLP, we set a range for SLP_{pre} and SLP_{post}, allowing LTP/LTD propagation to a fraction of neurons adjacent to those involved in the induction of LTP/LTD (with both "+" and "-" at SLP_{pre} in Fig. 2D). Following biological findings, we allowed SLP_{post} for LTD, but not for LTP (only with "-" at SLP_{post} in Fig. 2D). We also introduce a parameter λ_{pre}, λ_{post} to set the amount of SLP propagated to other synapses.

The integration of STDP, two subtypes of SLP, and homeostatic membrane potential will be integrated for reaching the efficient credit assignment towards the best $W_{j,k}$ (from presynaptic neuron j to postsynaptic neuron k, Fig. 2B).

C. The LIF neuron in SNN

Different types of neuron models have been proposed and used in SNNs, including but not limited to the LIF or spike-response-model neurons (with the maximal single equilibrium state point), Izhikevich neuron (with the maximal two equilibrium state points). LIF neuron is considered a good compromise between biophysical plausibility and computational efficiency, making it more appropriate as the basic neuron model in the shallow SNN.

The LIF neuron model works for non-linear information integration and non-differential spikes generation. When the neuron receives the spike signals from presynaptic neurons, the membrane potential V_t will be dynamically integrated, and spikes will be generated when reaching the firing threshold V^{Tr}, and a range of refractory time is kept after spikes. The V_t will be kept with the resting potential during the refractory time.

A diagram depicting the dynamic changes of membrane potential in LIF is shown in Fig. 2A. The detailed description of the integrated and fire procedure of the LIF neuron model is shown as the following Equation (1).

$$\begin{cases} \tau_m \frac{V_t}{dt} = -(V_t - V_L) - \frac{g_{E/I}}{g_L}(V_t - V_{E/I}) + \frac{I_t}{g_L} \\ \tau_s \frac{dI_t}{dt} = -I_t + \Sigma_{j \in N_E} W_{i,j} \delta(t - t_s) \\ V_t^F = V_t \end{cases}, \quad (1)$$

where the dynamic membrane potential V_t in LIF will be integrated with dt, the $\tau_m = C_m/g_L$, the C_m is the membrane capacitance, the g_L is the leaky conductance, V_L is the leaky potential, I_s is the current transferred from presynaptic spikes by synapses, g_E is the excitatory conductance, g_I is the inhibitory conductance, V_E and V_I are the reversal potentials for the excitatory and inhibitory neurons respectively, $\delta(t-t_s)$ marks the position of spikes (i.e., $\delta(t - t_s) = 1$ if $t = t_s$).

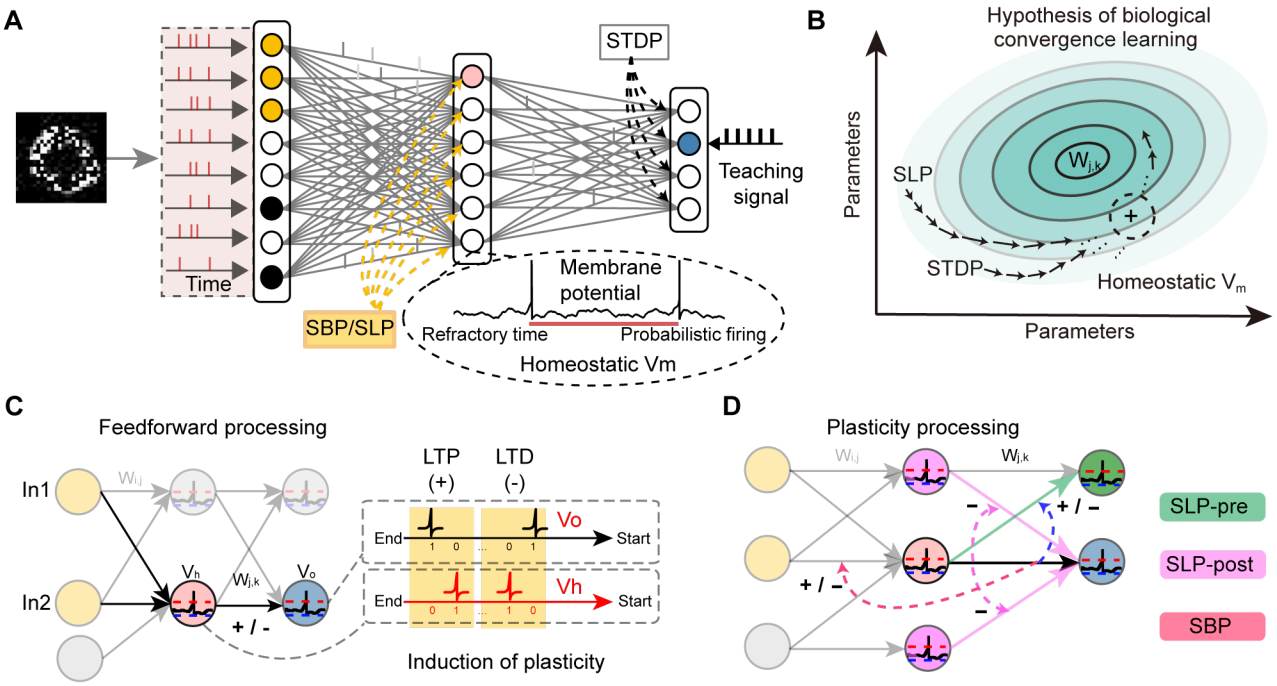

Fig. 2: Introducing biological lateral pre-and-post synaptic propagations (SLP) into SNNs. **A**, The three-layer architecture of SNN, in which SLP and local plasticity (STDP, homeostatic ΔV_m adjustment) was introduced at synapses onto hidden and output layers. (**B-C**), Schematic diagrams depicting the hypothesis of convergent learning in natural neural networks. **C**, Feed-forward propagation of spiking signals through the three layers without synaptic plasticity. LIF neurons in the hidden and output neurons are shown with spikes together with resting membrane potential (blue) and threshold (red). $W_{i,j}$ and $W_{j,k}$, synaptic weights. Out, spike sequences at an output layer neuron are shown in the box. And induction of LTP/LTD at the synapses on output neurons by STDP, based on the relative timing of presynaptic spikes and postsynaptic spikes resulting from the addition of out spike train and teaching spike train. V_h and V_o are the membrane potentials of hidden and output neurons, respectively. **D**, Back- and lateral presynaptic self-propagation of LTP and LTD. SBP, LTP, and LTD created at synapses of hidden layer neuron (pink) onto an output neuron (blue) were allowed to spread with a proportional factor ($\sim$0.1) to synapses from a fraction ($\sim$0.2) of input layer neurons (orange). SLP_{pre}, LTP and LTD were also allowed to spread with a proportional factor ($\sim$0.1) to output synapses of a hidden layer neuron (pink) to a fraction ($\sim$0.2) of its output synapses on the output layer neurons (blue). Lateral postsynaptic self-propagation LTD. SLP_{post}, LTD created at synapses made by a hidden layer neuron (pink) onto an output neuron (blue) was allowed to spread with a proportional factor ($\sim$0.1) to all synapses made onto the output neuron (blue).

The detailed configurations of these variables in different experiments are further shown in the experimental parts.

D. The unsupervised learning of SNNs by integrated local plasticity principles

The tuning of synaptic weights in SNNs is a challenging task. The spikes leave the gradient-like propagation stopped from membrane potential to its synaptic input weights. However, many tricks can help us resolve this problem to some extent. It is not difficult to conclude that in ANNs, the synaptic weights $W_{i,j}$ are directly linked to a global energy function E or cost function C.

In SNNs, there is no direct relationship between synaptic weights and global functions through the conventional differential chain rule, which is also commonly described as biological implausibility. Then we can use the tuning of membrane potential to get around this non-biological plausible problem. A particular energy function E can be designed and used to represent the global network state of SNN, with a similar spirit of energy function in the Hopfield network. The tuning of membrane potential can then be described as the following Equation (2).

$$E = \Sigma_{j=0}^{N}[\alpha_a V_{j,t}^2 + \alpha_b \Sigma_i^N V_{i,t} W_{i,j} V_{j,t} + \alpha_c V_{j,t}], \quad (2)$$

where the $V_{j,t}^2$ shows the second-order exponential power membrane potential of $V_{j,t}$, $V_{i,t}W_{i,j}V_{j,t}$ is the product of synaptic weight $W_{i,j}$ and its presynaptic ($V_{i,t}$) and postsynaptic ($V_{j,t}$) neuron states, $V_{j,t}$ shows the first-order exponential power of $V_{j,t}$.

We could also set $\alpha_a = 1/2$, $\alpha_b = -1$, $\alpha_c = -V^{Tr}$, and get the unsupervised representation of membrane potential as follows in the following Equation (3).

$$E = \Sigma_{j=0}^{N}[\frac{1}{2} V_{j,t}^2 - (\Sigma_i^N V_{i,t} W_{i,j} V_{j,t} + V^{Tr} V_{j,t})]. \quad (3)$$

Hence, in conclusion, after making a partial differential of E and V, we could get the following Equation (4).

$$\begin{cases} \Delta E_j \propto \frac{\partial E_j}{\partial V_{j,t}} = V_{j,t} - (\Sigma_i^N V_{i,t} W_{i,j} + V^{Tr}) \\ \Delta V_{j,t}^E = -\eta^e [V_{j,t} - (\Sigma_i^N V_{i,t} W_{i,j} + V^{Tr})] \end{cases}, \quad (4)$$

where the ΔE_j could be considered as the difference between current neuron state $V_{j,t}$ and its future state $V_{j,t+1} = (\Sigma_i^N (V_{i,t} W_{i,j} + V^{Tr})$, which represents temporal differential membrane potential state of neuron j.

Then the tuning direction of $\Delta V_{j,t}$ will go for the equilibrium state of all the neurons, contributing to the network balance. The membrane potential from different pathways, including that from Equation (1) and Equation (4), can be integrated as the following Equation (5):

$$\Delta V_{j,t} = \frac{t}{T} \Delta V_{j,t}^F + (1 - \frac{t}{T}) \Delta V_{j,t}^E, \quad (5)$$

where the $V_{j,t}^F$ shows the neuron state updated in the feed-forward procedure in Equation (1), T is the total simulated time. The two types of neuron state $V_{j,t}^E$ in Equation (4) and $V_{j,t}^F$ will be weighted summation as the total membrane potential $V_{j,t}$, with different weighted parameters of $\frac{t}{T}$ and $1 - \frac{t}{T}$, respectively. During learning, the weight strength for $V_{j,t}^F$ will be dynamically increased until 100% ($\frac{t}{T} = 1$ given $t = T$), while at the same time, the influence of $V_{j,t}^E$ will decrease until zero as the training time t goes by.

E. Supervised learning of the readout layer in SNNs

It is necessary to give a supervised teaching signal to guide SNNs to the correct target in classification tasks. We here also generate a teacher signal by repeating the one-hot label vector during input spike trains. This effort can guide the distribution of output spikes to a predefined target spike distribution based on the synaptic modification using STDP. The spike trains with a stable frequency are designed as a teacher signal and given to the readout neurons of SNNs. To calculate the distance between output spikes and desired target spikes, the distance similar to the Van-Rossum distance [48] is used to measure it during learning, shown as the following Equation (6).

$$\begin{cases} C = \frac{1}{2} \Sigma_j^N (\alpha \times V_{j,t} - \alpha \times \delta(t - t_{sp}))^2 \\ \Delta V_{j,t}^C \propto -\eta^c (\alpha \times V_{j,t} - \alpha \times \delta(t - t_{sp})) \end{cases}, \quad (6)$$

where N is the number of neurons in the readout layer, α is a constant coefficient. We set the differences of $V_{j,t}$ and $\delta(t - t_{sp})$ as the total cost of the network. The distance of the readout layer is defined as cost C, which is different from the previously described energy E.

These loss-like functions of C in Equation (6) and E in Equation (3) are different but share some similarities. The E represents an unsupervised energy measurement, while the C is a supervised loss measurement. However, both of them are designed for changing the states of membrane potentials, i.e., using $\frac{\delta E}{\delta V_j}$ and $\frac{\delta C}{\delta V_j}$, instead of calculating synaptic weights directly, i.e., using $\frac{\delta E}{\delta W_j}$ and $\frac{\delta C}{\delta W_j}$.

F. Induction of synaptic plasticity by different types of STDPs

The application of self-lateral propagation has an important precondition of plasticity induction. Many algorithms can cause inductions. Here we select STDP for its simplicity. The standard bi-phasic STDP principle [31] is shown as the following Equation (7).

$$\Delta W_{i,j} = \begin{cases} A_+ e^{(\frac{\Delta t_j}{\tau_+})} & if(\Delta t_j < 0) \\ -A_- e^{(\frac{-\Delta t_j}{\tau_-})} & if(\Delta t_j > 0) \end{cases}, \quad (7)$$

where A_+ and A_- are the scale factors, and the Δt_j is the time difference between the spiking time of presynaptic and postsynaptic neurons. τ_+ and τ_- are the delay time parameter of the bi-phasic STDP.

However, the computation cost of this two-phasic STDP is very high. Here we further use another differential version of it, according to the equivalent definition, by using differential temporal presynaptic activity multiplying the postsynaptic activity [16]. The presynaptic membrane potential-based STDP, postsynaptic membrane potential-based STDP, and symmetry type STDP are shown as following Equation (8), Equation (9), and Equation (10), respectively.

$$\Delta W_{i,j} \propto V_{i,t+1} V_{j,t} - V_{i,t} V_{j,t}. \quad (8)$$

$$\Delta W_{i,j} \propto V_{i,t} V_{j,t+1} - V_{i,t} V_{j,t}. \quad (9)$$

$$\Delta W_{i,j} \propto V_{i,t+1} V_{j,t+1} - V_{i,t} V_{j,t}. \quad (10)$$

These STDPs have been successfully applied on pattern recognition tasks [16]. In this paper, these three STDP principles will be tested in the experiments in the next section.

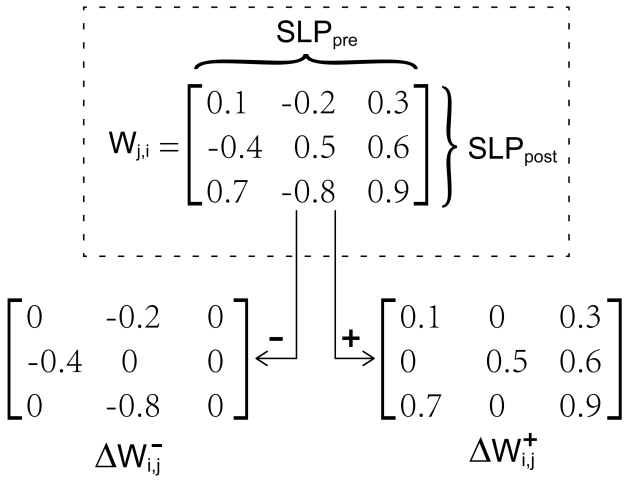

Fig. 3: The SLP_{pre} and SLP_{post} from a perspective of matrix factorization on inducted synaptic modification of $\Delta W_{i,j}$.

G. SLP_{pre} and SLP_{post} in SNNs

From the perspective of matrix factorization, the function of lateral presynaptic propagations could also be considered

as the column alignment of the weight matrix, which describes the column influence from a specific $\Delta W_{i,j}$ (induction synapse). A similar conclusion could also be given that the lateral postsynaptic propagation (SLP_{post}) is doing the row alignment. It is necessary to mention that only LTD is propagated during SLP_{post}.

As shown in Fig. 3, the $\Delta W_{i,j}$ matrix is calculated by the STDP rules, which has previously calculated by the dynamic membrane potential in the differential procedure of unsupervised E and supervised C to membrane potential. In the $\Delta W_{i,j}$ matrix, the positive values indicate the LTP induction, while the negative values indicate the LTD induction. Matrix factorization can help us separate LTP and LTD, which makes the different propagation of SLP_{pre} and SLP_{post} possible.

Hence, the three subtypes of synaptic weight update procedures in SLP_{pre} and SLP_{post}, named as $\Delta W_{i,j}^{LTP_{pre}}$, $\Delta W_{i,j}^{LTD_{pre}}$, and $\Delta W_{i,j}^{LTD_{post}}$, respectively, as shown in the following Equation (11).

$$\begin{cases} \Delta W_{i,\sim}^{LTP_{pre}} = \eta_1 E_{diag}(E_I + \sigma_i(\Sigma_j \Delta W_{i,j}^{+})) \times \Delta W_{i,\sim}^{+} \\ \Delta W_{i,\sim}^{LTD_{pre}} = \eta_2 E_{diag}(E_I + \sigma_i(\Sigma_j \Delta W_{i,j}^{-})) \times \Delta W_{i,\sim}^{-} \\ \Delta W_{\sim,j}^{LTD_{post}} = \eta_3 E_{diag}(E_I + \sigma_j(\Sigma_i \Delta W_{i,j}^{-})) \times \Delta W_{\sim,j}^{-} \end{cases}, \tag{11}$$

where σ_i denotes a non-linear mapping function which zooms out $W_{i,j}^{+}$ into a range of [0, 1], $W_{i,\sim}^{+}$ and $W_{\sim,j}^{+}$ are synapses influenced by the pre-lateral and post-lateral propagations. E_I denotes the unit matrix, E_{diag} denotes the function of creating a proper diagonal matrix, η_1, η_2, η_3 are the proportionality coefficients.

Both SLP_{pre} and SLP_{post} will be dynamically changed. While in the LTP procedure, only SLP_{pre} will be updated accordingly. These three types of propagation show that the LTP has a more substantial influence on lateral propagation. This phenomenon fits very well for the fundamental constructive motivation of the postsynaptic differential STDP principle, as shown in Equation (9), where the postsynaptic differentiation is more important and efficient than the presynaptic one [16].

H. The gradient-based SLP

The vanilla SLP rule (i.e., plasticity-based SLP) is biologically plausible but also shares some strong limitations: 1) only one neighborhood layer is accepted for the plasticity propagation, which is mainly caused by the limitation of the biological neurons where the internal currents or calcium signals can only propagated limited distances; 2) LTP and LTD have different flexibility in SLP, whereby LTD can propagate in both SLP_{pre} and SLP_{post}, while as a comparison, the LTP can only propagate in SLP_{pre} without SLP_{post}. Here we further release these biological limitations by allowing the further artificial propagation of SLP with more layers and more types, as shown in Equation (12). Here we name it "gradient-based SLP" for its similarity to the conventional gradient-based propagation computation.

Algorithm 1 The learning pipeline of SNNs using SLP

1. Start network initialization:

Conversion from raw data to spike trains. Initialization of parameters, including weights $W_{i,j}$, $W_{j,k}$ (under distribution of $U(0, 0.2)$), membrane potential V_i, V_j, V_k, leaky potential V_L, iteration time I_{ite}, simulation time T, differential time Δt, learning rate η^e, η^c, and conditional factor λ in SLP;

2. Start training procedure:

if not convergence **then**

2.1 Load training samples;
2.2 Update feedforward V_j^F by Equation (1);
2.3 Update homeostatic V_j^E by Equation (4);
2.4 Update integrative V_j^F and V_j^E by Equation (5);
2.5 Update readout V_j by Equation (6);
2.6 Update $W_{i,j}$ and $W_{j,k}$ by Equation (8)-(10);
2.7 Update $W_{i,j}$ and $W_{j,k}$ by SLP by Equation (11);

end if

Save the tuned $W_{i,j}$ and $W_{j,k}$ for the test.

3. Start test procedure:

3.1 Load test samples;
3.2 Test SNN using SLP with feedforward propagation;
3.3 Output test performance without cross-validation.

$$\begin{cases} \Delta W_{i,\sim}^{gLTP_{pre}} = \eta_1 E_{diag}(E_I + \sigma_i(\Sigma_j \Delta W_{i,j}^{+})) \times \Delta W_{i,\sim}^{+} \\ \Delta W_{i,\sim}^{gLTD_{pre}} = \eta_2 E_{diag}(E_I + \sigma_i(\Sigma_j \Delta W_{i,j}^{-})) \times \Delta W_{i,\sim}^{-} \\ \Delta W_{\sim,j}^{gLTP_{post}} = \eta_3 E_{diag}(E_I + \sigma_j(\Sigma_i \Delta W_{i,j}^{+})) \times \Delta W_{\sim,j}^{+} \\ \Delta W_{\sim,j}^{gLTD_{post}} = \eta_4 E_{diag}(E_I + \sigma_j(\Sigma_i \Delta W_{i,j}^{-})) \times \Delta W_{\sim,j}^{-} \\ \Delta W_{i,j}^{gBP} = \eta_5 E_{diag}(E_I + \sigma_j(\Sigma_k \Delta W_{j,k})) \times \Delta W_{i,j} \end{cases}, \tag{12}$$

where $\Delta W_{i,j}$ and $\Delta W_{j,k}$ are synaptic modifications in different layers.

I. The learning procedure of SNNs using SLP

The detailed learning procedure of SLP-improved SNNs is shown in the Algorithm 1. The SLP can be well integrated with the plasticity induction caused by STDPs. The membrane potential-centric learning architecture is used by calculating the credit assignment of membrane potential first (by both energy function and cost function) and then consolidating the learned credit assignment (by temporally saved membrane potential changes) to the local synaptic weights by both STDP and SLP.

IV. Experiments

A. Three benchmark datasets

In this study, we examined the effects of introducing SLP on the accuracy of SNNs for three benchmark learning tasks involving different extents of temporal information. The first one is the recognition of handwritten digits, using a modified national institute of standards and technology (MNIST) dataset (Fig. 4A) [49]. The second is the recognition of neuromorphic MNIST (N-MNIST) [50], which is a spiking version of the

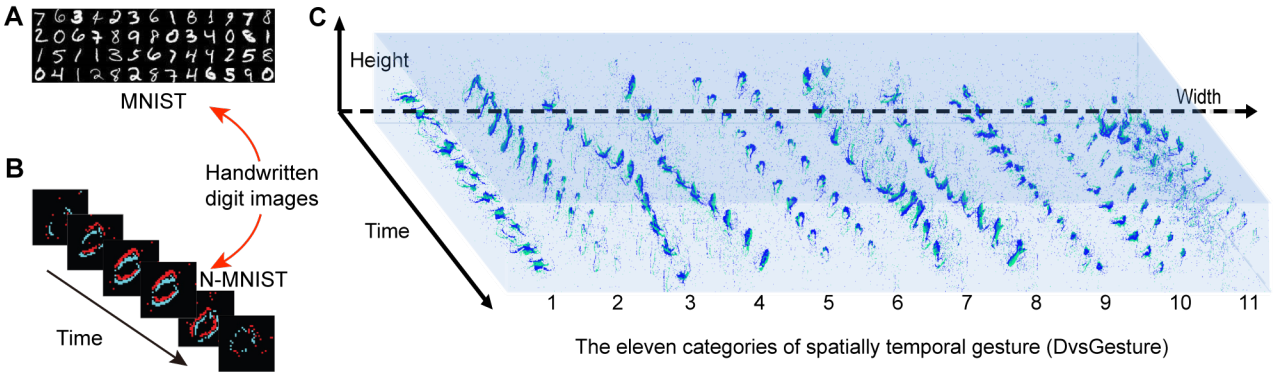

Fig. 4: The three benchmark datasets. **A**, The MNIST dataset contains 10 classes and 70,000 (60,000 training and 10,000 test samples) spatial handwritten images. **B**, The neuromorphic-MNIST (N-MNIST) dataset contains 10 classes and 70,000 (60,000 training and 10,000 test samples) spiking versions of the handwritten images.**C**, The DvsGesture dataset contains 11 classes and 1,464 (1,176 training and 288 test samples) spatial-temporal gesture sequences. Each sample comprises 1,800 video frames of event-based gesture records. The 11 classes include hand waving (both arms), large straight arm rotations (both arms, clockwise and counterclockwise), forearm rolling (forward and backward), air guitar, air drums, and others.

MNIST dataset (Fig. 4B). The third is gesture recognition, using an event-based dynamic vision sensor gesture (DvsGesture) dataset (Fig. 4C) [51].

B. Experimental configurations

Most of the key parameters in SNNs using SLP would keep the same while learning two tasks. The number of epochs is 100. The proportions of excitatory and inhibitory neurons are 50% and 50%, respectively. The reversal potential for E-neuron (V_E) and I-neuron (V_I) is 0.2 mV and 0.1 mV, respectively. Leaky potential (V_L) is 0 mV. The firing threshold (V_{th}) is 0.1 mV. Resetting membrane potential (V_{reset}) is 0 mV. The time parameter τ_m in LIF is 2 ms. τ_E and τ_I are 2 ms and 4 ms, respectively. Leaky variables g_E and g_I are set as 1. The basic time slot dt is 0.01 ms during simulation, and the total simulation time T is 10 ms. The architectures of SNNs for MNIST and DvsGesture are [784, 500, 10] (input dimension, hidden dimension, and output dimension) and [1024, 500, 11], respectively. The batch size is 20 (MNIST, N-MNIST) and 10 (DvsGesture), respectively. The positive and negative iterations of them are [5, 20] (MNIST) and [10, 10] (DvsGesture), respectively. The learning rate of them to output and hidden layers are [0.05, 0.1] (MNIST) and [0.05, 0.05] (DvsGesture), respectively.

C. Convergence learning by SLP-based plasticity principle

Network convergence is usually an essential premise for the next-step performance analysis. We compared the performance of two types of SLPs on two different tasks.

For learning hand digit recognition on the MNIST dataset, we used an SNN comprising 784 input neurons, 500 hidden neurons (half excitatory and half inhibitory), and ten output neurons. We trained the SNN with a subset (60,000) of the MNIST dataset and tested its accuracy and computational cost using the remaining MNIST data (10,000).

After learning MNIST, as shown in Fig. 5A, the test error rates of these two algorithms were dynamically reduced until a plateau of 6.08% and 8.27% for SNNs using SLP_{pre} and SLP_{post}, respectively. The SNNs using SLP_{pre} exhibited a better performance (lower test error rate) than that using SLP_{post}. One of the possible reasons is that SLP_{pre} propagates both LTP and LTD, while SLP_{post} propagates only LTD. Furthermore, SLP_{pre} is more like a pre-synaptic membrane potential-based STDP, as shown in Equation (8), while SLP_{post} is more like a post-synaptic membrane potential-based STDP as shown that in Equation (9). The differences between these two types of STDPs have already been discussed [16], [24].

For learning gesture recognition on the DvsGesture dataset, we used an SNN architecture comprising 1,024 input neurons, 500 hidden neurons (half excitatory and half inhibitory), and 11 output neurons (corresponding to 11 gesture types). After learning DvsGesture, the test error rates of SLP_{pre} and SLP_{post} were also convergence, reaching a plateau of 17.13% and 32.59% for SNNs using SLP_{pre} and SLP_{post}, respectively. As shown in Fig. 5B, the experimental results showed that a relatively better convergence learning was achieved by introducing SLP_{pre} than SLP_{post}, similar to those in the classification of the MNIST dataset.

We also tried to understand the SLPs by further analyzing the internal neuronal states of input and hidden layers during network learning. We used a dimension-reduction method (i.e., t-SNE [52]) to convert information from the high-dimension to a much lower two-dimension one without changing the relative distances of samples. As shown in Fig. 5C, the 10-class MNIST data were clustered after dimensional reduction from 784 dimensions of spikes trains in the first layer of SLP improved SNN, where different colors represented different classes. A better clustering performance could be achieved for neurons at hidden layers after the t-SNE clustering (Fig. 5D), which to some extent showed the proper convergence learning of membrane potential guided by the SLP improved SNN.

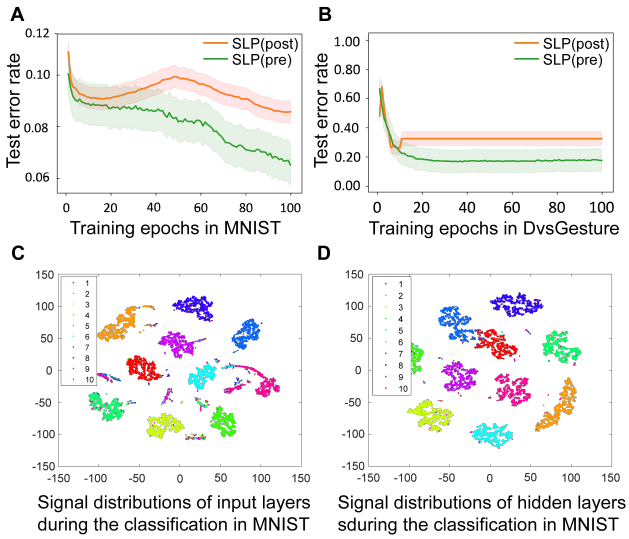

Fig. 5: Performance of SNNs using SLP on two tasks. **A**, SNNs using SLP_{pre} and SLP_{post} were convergent while learning the MNIST dataset. The SNNs using SLP_{pre} exhibited a lower test error rate than those using SLP_{post}. **B**, Similar to **A**, but while learning the DvsGesture dataset, containing a special network anomaly point where the network stops learning. All figures are averaged over five repeating experiments with different random seeds. (**C, D**), Signal distributions of input (**C**) and hidden (**D**) layers of SNNs during the classification of the MNIST dataset. The t-SNE tool [52] is used for dimensional reduction. Each color represents a specific class.

D. Sharpening the distribution of synaptic weights

The distributions of synaptic weights in neural networks are highly related to their flexibility, generalization, and interpretability. We tried to understand the function of SLP by analyzing their distributions of synaptic weights after learning MNIST and DvsGesture datasets.

We used W_1 to represent the synaptic weights between input and hidden layers ($W_{i,j}$), represented as a matrix with the dimension of 500×784 (784 input neurons and 500 hidden neurons). For easier comparison, we normalized W_1 by summing all of the absolute weights "onto" the same hidden neuron, represented as a vector with the dimension of 500×1 (Fig. 6A1). The weights in SNNs using or without using SLP were normally distributed but with different sharpen characteristics, where those using SLP were much sharper than those without SLP (Fig, 6A2). This much sharper feature of weight distributions has been considered necessary for better information discrimination [53].

Similar to that in Fig. 6A1, we use W_2 to represent the synaptic weights between hidden and output layers ($W_{j,k}$), with a dimension of 10×500 (500 hidden neurons and ten output neurons). The weights after normalization (also "onto" the hidden neuron, with a dimension of 1×500) were different from each other for SNNs using or without using SLP (Fig. 6B1), and the normal distributions for SNNs using SLP were sharper than those without using SLP (Fig. 6B2).

As shown in Fig. 6C, more histogram distribution and scatter diagrams were used to analyze the relationship between W_1/W_2 and with/without SLP.

- As shown in Fig. 6C1, W_1 and W_2 given SLP exhibited a linear relationship on the scatter diagram.
- As shown in Fig. 6C2, a similar linear relationship was achieved by W_1 and W_2 without given SLP.
- As shown in Fig. 6C3, W_1 with or without SLP exhibited a non-linear relationship.
- As shown in Fig. 6C4, W_2 with or without SLP exhibited a non-linear relationship.

These results indicated that the SLP would keep the linear relationship of W_1 and W_2 weight vectors (weights onto the hidden layers, Fig. 6C1,C2). Inner a specific W_1 or W_2 weight vector, no linear relationship was detected by giving or not giving SLP (Fig. 6C3,C4).

For the DvsGesture task, a similar conclusion (Fig. 6D,E,F) could also be given, like those in the MNIST dataset (Fig. 6A,B,C). These results showed that the change of learning policy (with or without SLP) might significantly change the synaptic weights that came from the same initialization.

E. More uniform distribution of misclassified samples

For opening the black box of neural networks, many visualization methods have been used to monitor the dynamic learning process of some learned parameters, e.g., the synaptic weights or some inner states, e.g., the learning trajectory of neuronal activity. Here we proposed a special effort to better learn networks' generalization by monitoring the performance of misclassified samples.

As the MNIST task, for example, after learning and in the inference phase, all 10,000 samples were tested and then calculated within a confusion matrix, which described the difference between predicted and expected labels (Fig. 7A1). Some misclassified images were selected and shown as examples in Fig. 7A2, where some digits were a blur and might alter from one type to another (e.g., digits four and nine). We further selected the non-diagonal elements in the confusion matrix and plotted them with the pure misclassified samples (Fig. 7A3, A4 for the MNIST task). This result showed that the distributions of error samples after learning with SLP were more uniform than that without SLP (e.g., the over misclassified of digit seven and night, red dot in Fig. 7A4), congruent with the t-SNE visualization of network output given misclassified samples with SLP (Fig. 7C1) or without SLP (Fig. 7C2).

Similar conclusions could also be given after analyzing results from the DvsGesture task, where the more uniform distribution of misclassified samples was achieved by using SLP than without using SLP (Fig. 7B1, B2, 288 samples were tested), congruent with the t-SNE visualization of network output given misclassified samples with SLP (Fig. 7D1) or without SLP (Fig. 7D2).

F. Higher accuracy on MNIST dataset

Some benchmark SNN algorithms using plasticity-based learning rules were used to compare with those using SLP

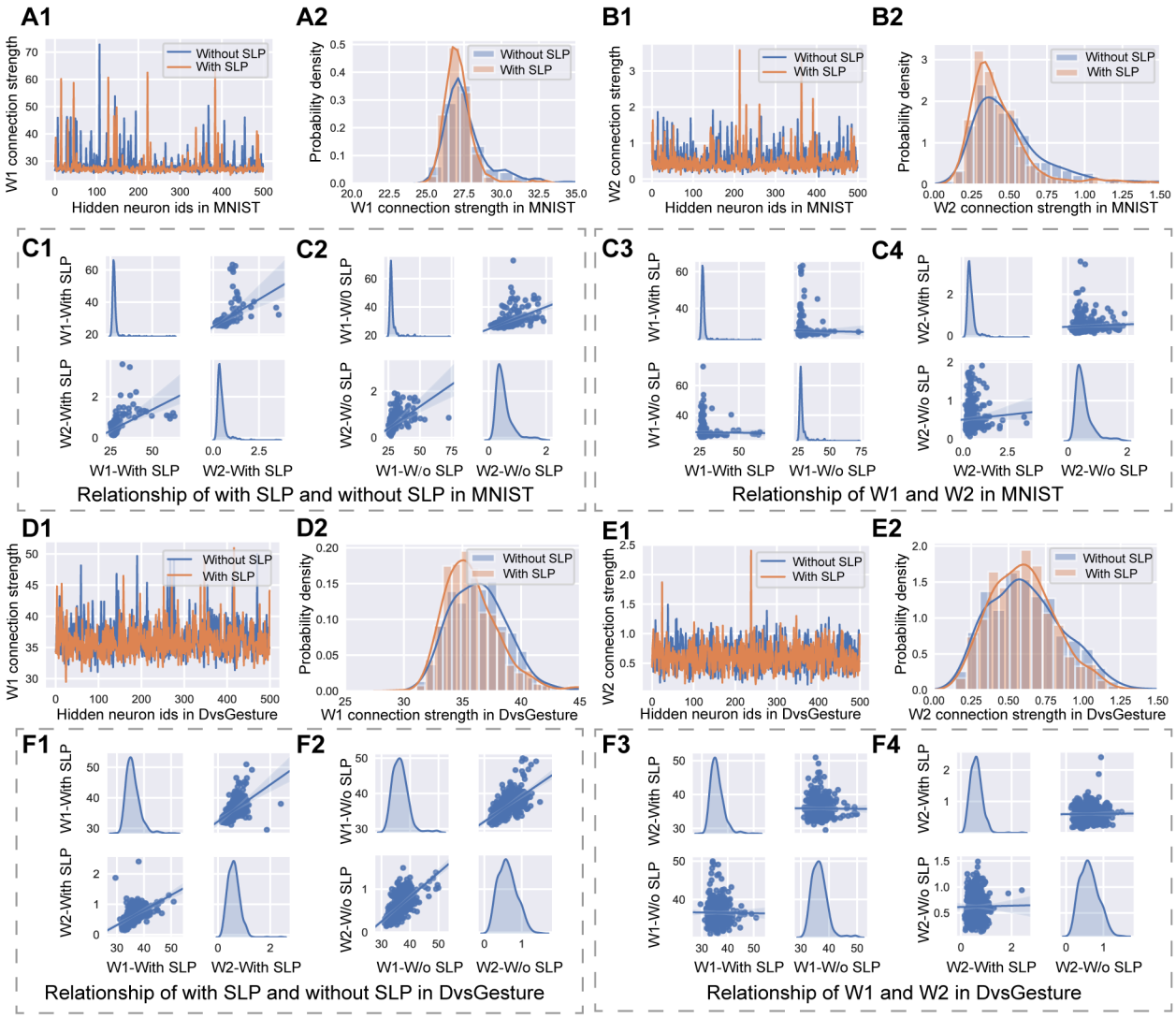

Fig. 6: SLP improved sharper distributions of synaptic weights in the MNIST task. (**A-C**), The distributional comparisons and analyses of W_1 (the set of $W_{i,j}$ onto the hidden layer) and W_2 (the set of $W_{j,k}$ onto the hidden layer) tuned with or without SLP on the MNIST dataset. **A1**, The distributions of W_1 (reduced into a one-dimensional vector with the same size of hidden neurons) with or without SLP. **A2**, The distributional comparison of W_1 between tuned with or without SLP. **B1**, The normalized W_2 distributions with or without SLP. **B2**, The distributional comparisons of W_2 between tuned with or without SLP. **C1**, The relationship analyses between different W_1 tuned with and without SLP. **C2**, The relationship analyses between different W_2 tuned with or without SLP. **C3**, The relationship analyses between W_1 and W_2 tuned with SLP. **C4**, The relationship analysis between W_1 and W_2 tuned without SLP. (**D-F**), Similar to those in (**A-C**) but on the DvsGesture dataset.

on the MNIST dataset. These algorithms were all milestone algorithms in the research area of SNNs, focusing more on efficient credit assignment by using pure biologically plausible plasticity principles instead of using gradient-based algorithms like an approximate gradient or pseudo-BP.

As shown in Fig. 8, some SOTA algorithms were listed for the performance comparison with our proposed SNNs using SLP. The first algorithm was a two-layer SNN using unsupervised STDP for synaptic modification, lateral inhibition for generating competition of neurons, and label reassignment after learning, as shown in the algorithm [I] in Fig. 8. These local plasticity principles were well-integrated together for the successful credit assignment and made SNN reach 95% accuracy on the MNIST dataset [37]. The second one was a three-layer SNN using the homostatic membrane potential to keep input-output equilibrium at the neuronal level and STDP for synaptic modification, reaching 98.5% accuracy [24], as shown in the algorithm [II] in Fig. 8. The third one was a multi-layer SNN using the reward-modulated STDP for strengthening the supervised learning [46], where convolutional layers and full connection layers were used for classification (the algorithm [III] in Fig. 8). The fourth one used latency coding, which focused more on the first-spike signal, and combined it with STDP for efficient multi-layer SNN learning, reaching 98.4% accuracy (the algorithm [IV] in Fig. 8) [43].

We built a simple three-layer SNN without using con-

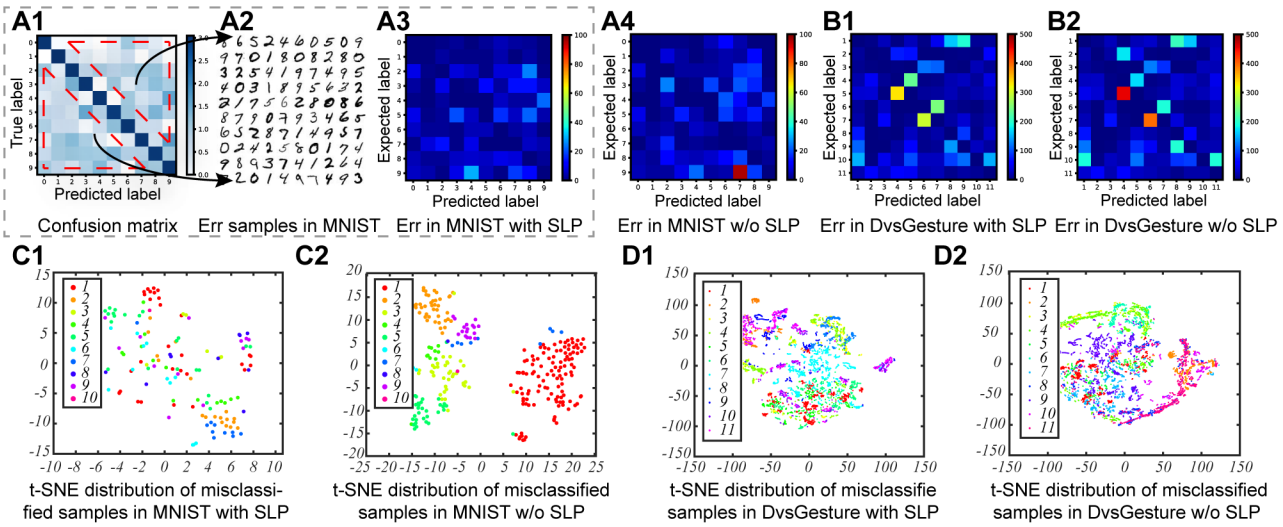

Fig. 7: SLP improved more-uniform distributions of network outputs for misclassified samples in confusion matrices. **A1**, The confusion matrix of classification accuracy with SNNs using SLP on the MNIST dataset. **A2**, The examples of misclassified samples with SNNs using SLP. **B1**, The matrix of misclassified samples with SNNs using SLP. **B2**, The matrix of misclassified samples for SNNs without using SLP. (**C1, C2**), The matrix of misclassified samples with (**C1**) and without (**C2**) SLP, respectively, on the MNIST dataset. (**D1, D2**), Same to those in (**C1, C2**) but on the DvsGesture dataset.

volutional layers to make a fair comparison. The SLP and STDP were local plasticity principles without using label signals, which were the most different features than other plasticity principles ([II, III, IV] in Fig. 8). The performance of synaptic modifications of the three-layer SNN using SLP and STDP reached 98.8% accuracy, which was higher than that using unsupervised STDP on two-layer SNN (95.0%, [I]) and reward-STDP on multi-layer SNN (98.4%, [III]).

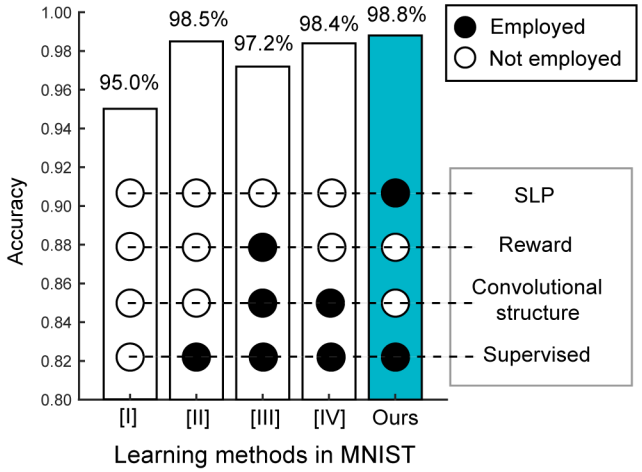

Fig. 8: The performance of SLP improved SNNs on the MNIST task, compared to other state-of-the-art (SOTA) methods, including a two-layer SNN using STDP [37] [I], three-layer SNN using the homostatic membrane potential and STDP [II] [24], multi-layer SNN using the reward-modulated STDP [III] [46], multi-layer SNN using latency coding and STDP [IV] [43] on the MNIST dataset.

G. Higher accuracy on DVS-based datasets

The proposed algorithm was also verified on more complex DVS-based datasets containing sequential spike trains, including N-MNIST and DvsGesture datasets. Some SOTA algorithms were refined or adapted to test the performance of these algorithms.

For the N-MNIST dataset, as shown in Table I, our three-layer SNN algorithm using plasticity-based SLP rule achieved 81.48% accuracy. As a comparison, the classification accuracy for the SKIM algorithm using kernel methods reached 83.44%. The HFIRST algorithm using hard or soft classifier achieved maximal 71.15% accuracy.

For the DvsGesture dataset, as shown in Table I, the algorithms included the two-layer SNN using STDP [37], three-layer SNN using differential STDP, three-layer SNN using symmetric STDP, and three-layer SNN using the homostatic membrane potential and STDP [24]. The SNNs using SLP reached 83.5% accuracy, higher than other plasticity-based algorithms.

H. The deeper SNN using gradient-based SLP

Usually, a deeper architecture is commonly designed for more challenging tasks to reach higher performance. Here we replaced the vanilla SLP with the gradient-based SLP. We found a slightly higher accuracy was achieved by the gradient-based SLP (98.89%), compared to that by the plasticity-based SLP (i.e., 98.80%) on the MNIST dataset under the same network configurations (three-layer SNN with 500 hidden neurons, Table. II). Furthermore, after the layer number of SNNs increased from three to five, given 500 hidden neurons for each layer, we found the accuracy slightly and gradually increased from 98.89% to 98.90% and 98.91%, respectively. These experimental results indicated that the SLP might be

TABLE I: The performance of SLP improved SNNs on the DVS-based datasets.

N-MNIST	Learning Rule	Accuracy (%)
HFIRST [54]	Hard-classifier	71.15
HFIRST [54]	Soft-classifier	58.40
SKIM [55]	Kernel methods	83.44
SLP	**Plasticity-based**	**81.48**
DVS-Gesture	**Learning Rule**	**Accuracy (%)**
2-layer SNN [37]	STDP	82.7
3-layer SNN [56]	Differential STDP	82.4
3-layer SNN [12]	Symmetric STDP	69.7
3-layer SNN [24]	Homeo-V	26.9
SLP	**Plasticity-based**	**83.5**

more effective for relatively shallow SNNs under similar accuracy performance by requiring less computational cost (see the next section for more details).

TABLE II: Accuracy of shallow SNNs using plasticity-based SLP and deep SNNs using gradient-based SLP on the MNIST dataset.

SLP Type	Layers	Neurons	Accuracy (%)
Plasticity	Three	500	98.80
Gradient	Three	500	98.89
Gradient	Four	500	98.90
Gradient	Five	500	98.91

It is necessary to mention that the classification accuracy of SNNs using plasticity-based algorithms was much lower than those using gradient-based algorithms. The conversion-based algorithm, whereby the trained synaptic weights in ANNs were then converted to SNNs by changing ReLU active function to LIF neuronal model and other related constraints, could even reach a much higher accuracy (around 99.6% accuracy on MNIST) that is even closer to the best of deep ANNs. However, this paper's key research point is training SNN with biologically plausible plasticity principles. We insisted on this and believed that plasticity-based algorithms would inspire a new-generation algorithm incorporating much better interpretability, flexibility, and energy efficiency without affecting accuracy.

I. Reduced computational cost by SLP

The computational cost of SNNs includes two types. The first type is the inference cost used in neuromorphic systems, whereby the key reason for saving energy consumption is using addition instead of multiplication calculation during spike propagation. The second type is the training cost, which is more important for animal survival for reaching lifelong energy-efficient learning. Here we focus more on the second type for its importance. The computational complexity is represented as $O(\cdot)$, including feed-forward propagation (FF_1 and FF_2) and feed-back propagation (FB_1 and FB_2) in each training epoch, where FF_1 ($O(mn)$) and FF_2 ($O(nk)$) represent the forward propagation procedure from input-to-hidden and hidden-to-output layers, respectively, where m, n, k represent the numbers of neurons in different layers. FB_1 ($O(kn)$) and FB_2 ($O(nm)$ for BP and $O(n)$ for SLP) are similar to them but used for the backward propagation procedure. The calculation of computational cost is shown as the following Equation (13).

$$Cost = \left(\sum_{i=1}^{k-1} N_{pre} \times Rate \times N_{post} \times E_{op} \right) \times T \times Steps, \tag{13}$$

where k is the number of layers; N_{pre} and N_{post} are the number of pre and post-synaptic neurons, respectively; $Rate$ is the firing rate of a pre-synaptic neuron (1 for ANN and 0$\sim$1 for SNN); E_{op} is the energy consumption per operation (4.6 pJ for ANN and 0.9 pJ for SNN [57]), T is the time window (1 for ANN and 20 for SNN), $Steps$ is the learning steps for reaching a predefined accuracy (i.e., 90% for MNIST task). After calculation, we find that the plasticity-based algorithms (e.g., SLP) can achieve lower computational cost ($4.6\times10^7 pJ$) than that using the gradient-based algorithms (e.g., BP, $7.3 \times 10^7 pJ$ cost).

V. Discussion and conclusion

One of the critical motivations of this paper is to open the black box of the multi-scale tuning mechanisms in natural neural networks and then apply them to SNNs. We focused more on the learning methods. Hence the network architectures are designed as simply as possible. The networks with deeper or more complicated layers, including convolutional kernels, sparse connections, or long-term feedback loops, are out of this paper's scope and will be further discussed in the next-step research. We have successfully tuned SNNs with SLP and the standard biologically plausible STDP principle. Many essential and important questions can be further discussed here.

Why is the integration of SLP and STDP not conflicted? The STDP describes the synaptic modification based on pre and postsynaptic neuron firing differences. SLP describes the propagation of STDP plasticity from the synapse of induction to neighborhood synapses. These two principles could be two-step plasticity that covers sequential induction and propagation, making them non-conflict. This phenomenon also hints that the natural neural network formed different functions by integrating more unsupervised plasticity principles instead of a single globally supervised one like BP in ANNs.

Why is the unsupervised learning algorithm finally convergent? The SNN using SLP and STDP did not contain global plasticity like BP in ANNs. They are both unsupervised, which might lead the direction of synaptic modification to a non-global minimal. To resolve this problem, we used our previous SNN learning strategy [24] that contained two unsupervised learning phases for updating membrane potential and a supervised learning phase for consolidating output synapses towards the expected network outputs. The interleaves of two phases could guide the network towards the accepted minimum.

Why can the SLP improve the efficiency of SNNs? Some experimental results have already shown that a higher accuracy would be achieved on MNIST and DvsGesture datasets by SNNs using SLP. These results might be more caused by plasticity's propagating feature, which broadens plasticity's influence. Furthermore, these additional propagations start from the presynaptic or postsynaptic neurons. Hence it could be considered a special type of neuronal tuning. Since neuronal and synapse tuning is two basic viewpoints of network learning, we might integrate the neuron-centric and synapse-centric learning algorithms.

Why sharpening the distribution of synaptic weights is important? The sharpening feature is highly related to the information discrimination ability of the network. To some extent, the SLP propagates STDP to an anti-overfitting problem in SNNs, which has also been verified as important to an anti-over-strengthen pathway in network learning. This viewpoint could also explain the uniform distribution of misclassified samples.

Could these plasticity-based principles finally compete with BP? The answer is definitely. Here, we highlight the two pathways available for biologically plausible computation. One is that from an artificial viewpoint by refining and modifying BP-like gradient with the inspiration of neuroscience [1], [25], [24], [58], [30]. The other is from a biological viewpoint by improving the current plenty of plasticity principles [10], [12], [37], [42], [46], [43]. Here we stand more on the second viewpoint and try to give an alternative hypothesis that the biologically plausible SLP is one of these key plasticity principles that are important for understanding the nature of biological intelligence. The SLP found in the natural neural network [29], [30], [31], [32] is an integrative learning principle that contains three sub-principles: presynaptic lateral spread propagation, postsynaptic lateral spread propagation, and neighborhood-layer backpropagation [33]. This paper focuses on the first two sub-principles of synaptic plasticity. They are mesoscale and closer to the global plasticity in the natural neural network that is still not identified.

What is the future of SNNs? To the best of our knowledge, the performances of SNNs still lag behind that of ANNs in classification tasks, especially on the accuracy of many benchmarks. However, brain-inspired SNNs have added many new features, making them more significant in intelligent information processing. The research on spike-based sparse encoding, population coding, and machine learning applications (e.g., XOR problem, Iris data classification, and MNIST and harder ones) have become hot topics [41], [59]. Some other efforts also speed up the SNN simulation environment by empowering computations on CPU parallelism (e.g., Auryn [60], Brian2 [61]) or GPU acceleration (e.g., GeNN [62]). These improvements are making research of SNNs easier and faster. For example, our recent work has shown that applying multiscale coding improved SNNs to reinforcement learning paradigms for continuous controlling can achieve higher reward scores by SNNs than ANNs [63].

In addition, hundreds of new plasticity principles or new architectures have been found with the fast advancement of neuroscience. In the future, biologically plausible learning principles will contribute to a more comprehensive framework for learning and give more inspiration to the next generation of artificial intelligence toward higher accuracy, lower computational cost, faster learning convergence, stronger robustness, and more flexibility without losing interpretability.

REFERENCES

[1] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton, "Backpropagation and the brain," *Nature Reviews Neuroscience*, pp. 1–12, 2020.

[2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

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

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

[5] I. M. Comsa, K. Potempa, L. Versari, T. Fischbacher, A. Gesmundo, and J. Alakuijala, "Temporal coding in spiking neural networks with alpha synaptic function," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Conference Proceedings, pp. 8529–8533.

[6] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal, "The "wake-sleep" algorithm for unsupervised neural networks," *Science*, vol. 268, no. 5214, pp. 1158–1161, 1995.

[7] S. Z. Muller, A. N. Zadina, L. F. Abbott, and N. B. Sawtell, "Continual learning in a multi-layer network of an electric fish," *Cell*, vol. 179, no. 6, p. 1382, 2019.

[8] G. Bellec, F. Scherr, A. Subramoney, and et al., "A solution to the learning dilemma for recurrent networks of spiking neurons," *Nature Communications*, vol. 11, no. 1, p. 3625, 2020.

[9] S. Song, K. D. Miller, and L. F. Abbott, "Competitive hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neuroscience*, vol. 3, no. 9, pp. 919–926, 2000.

[10] F. Zenke, E. J. Agnes, and W. Gerstner, "Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks," *Nature Communications*, vol. 6, p. 6922, 2015.

[11] R. S. Zucker, "Short-term synaptic plasticity," *Annual Review of Neuroscience*, vol. 12, no. 1, pp. 13–31, 1989.

[12] T. Zhang, Y. Zeng, D. Zhao, and B. Xu, "Brain-inspired balanced tuning for spiking neural networks," in *The Twenty-Seventh International Joint Conference on Artificial Intelligence*. IJCAI, 2018, Conference Proceedings, pp. 1653–1659.

[13] T. J. Teyler and P. DiScenna, "Long-term potentiation," *Annual Review of Neuroscience*, vol. 10, no. 1, pp. 131–161, 1987.

[14] T. V. Bliss and G. L. Collingridge, "A synaptic model of memory: long-term potentiation in the hippocampus," *Nature*, vol. 361, no. 6407, pp. 31–39, 1993.

[15] M. Ito, "Long-term depression," *Annual Review of Neuroscience*, vol. 12, no. 1, pp. 85–102, 1989.

[16] Y. Bengio, T. Mesnard, A. Fischer, S. Zhang, and Y. Wu, "Stdp as presynaptic activity times rate of change of postsynaptic activity approximates back-propagation," *Neural Computation*, vol. 10, 2017.

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

[18] Y. Bengio, N. Leonard, and A. C. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *CoRR*, vol. abs/1308.3432, p. 1308.3432, 2013.

[19] I. Fiete and H. S. Seung, "Gradient learning in spiking neural networks by dynamic perturbation of conductances," *Physical review letters*, vol. 97 4, p. 048104, 2006.

[20] T. Zhang, S. Jia, X. Cheng, and B. Xu, "Tuning convolutional spiking neural network with biologically plausible reward propagation," *IEEE Transactions on Neural Network and Learning System*, vol. PP, 2021.

[21] M. Zhang, H. Qu, A. Belatreche, Y. Chen, and Z. Yi, "A highly effective and robust membrane potential-driven supervised learning method for spiking neurons," *IEEE Transactions on Neural Network and Learning System*, vol. 30, no. 1, pp. 123–137, 2019.

[22] C. Ma, R. Yan, Z. Yu, and Q. Yu, "Deep spike learning with local classifiers," *IEEE Transactions on Cybernetics*, pp. 1–13, 2022.

[23] M. Zhang, J. Wang, J. Wu, and et al., "Rectified linear postsynaptic potential function for backpropagation in deep spiking neural networks," *IEEE Transactions Neural Network and Learning Systems*, vol. 33, no. 5, pp. 1947–1958, 2022.

[24] T. Zhang, Y. Zeng, M. Shi, and D. Zhao, "A plasticity-centric approach to train the non-differential spiking neural networks," in *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, Conference Proceedings, pp. 620–628.

[25] M. Helmstaedter, "The mutual inspirations of machine learning and neuroscience," *Neuron*, vol. 86, no. 1, pp. 25–28, 2015.

[26] M. Hä, usser, N. Spruston, and G. J. Stuart, "Diversity and dynamics of dendritic signaling," *Science*, vol. 290, no. 5492, pp. 739–744, 2000.

[27] 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 Neuroscience*, vol. 2, no. 1, pp. 65–73, 1999.

[28] J. Waters, M. Larkum, B. Sakmann, and F. Helmchen, "Supralinear ca2+ influx into dendritic tufts of layer 2/3 neocortical pyramidal neurons in vitro and in vivo," *Journal of Neuroscience*, vol. 23, no. 24, pp. 8558–8567, 2003.

[29] 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. 439–48, 1997.

[30] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, "Random synaptic feedback weights support error backpropagation for deep learning," *Nature Communications*, vol. 7, p. 13276, 2016.

[31] G.-q. Bi and M.-m. Poo, "Synaptic modification by correlated activity: Hebb's postulate revisited," *Annual review of neuroscience*, vol. 24, no. 1, pp. 139–166, 2001.

[32] J. L. Du, H. P. Wei, Z. R. Wang, S. T. Wong, and M. M. Poo, "Long-range retrograde spread of ltp and ltd from optic tectum to retina," *Proceedings of the National Academy of Sciences (PNAS)*, vol. 106, no. 45, pp. 18 890–6, 2009.

[33] T. Zhang, X. Cheng, S. Jia, M. M. Poo, Y. Zeng, and B. Xu, "Self-backpropagation of synaptic modifications elevates the efficiency of spiking and artificial neural networks," *Science Advances*, vol. 7, no. 43, p. eabh0146, 2021.

[34] H. S. Seung, "Learning in spiking neural networks by reinforcement of stochastic synaptic transmission," *Neuron*, vol. 40, no. 6, pp. 1063–1073, 2003.

[35] S. B. Shrestha and G. Orchard, *SLAYER: Spike layer error reassignment in time*. Curran Associates, Inc., 2018, pp. 1412–1421.

[36] D. Zambrano, R. Nusselder, H. S. Scholte, and S. Bohte, "Efficient computation in adaptive spiking neural networks," *Frontiers in Neuroscience*, vol. 12, p. 987, 2018.

[37] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in Computational Neuroscience*, vol. 9, 2015.

[38] C. Eliasmith, T. C. Stewart, X. Choo, T. Bekolay, T. DeWolf, Y. Tang, and D. Rasmussen, "A large-scale model of the functioning brain," *Science*, vol. 338, no. 6111, pp. 1202–1205, 2012.

[39] J. Guerguiev, T. P. Lillicrap, and B. A. Richards, "Towards deep learning with segregated dendrites," *ELife*, vol. 6, p. e22901, 2017.

[40] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, and et al., "Deep learning in spiking neural networks," *Neural Networks*, 2018.

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

[42] J. M. Brader, W. Senn, and S. Fusi, "Learning real-world stimuli in a neural network with spike-driven synaptic dynamics," *Neural Computation*, vol. 19, no. 11, pp. 2881–2912, 2007.

[43] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and e. Masquelier, Timothé, "Stdp-based spiking deep convolutional neural networks for object recognition," *Neural Networks*, vol. 99, pp. 56–67, 2018.

[44] S. M. Bohte, J. N. Kok, and H. La Poutre, "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, no. 1, pp. 17–37, 2002.

[45] J. Göltz, L. Kriener, A. Baumbach, and et al., "Fast and energy-efficient neuromorphic deep learning with first-spike times," *Nature Machine Intelligence*, vol. 3, no. 9, pp. 823–835, 2021.

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

[47] A. Meulemans, F. S. Carzaniga, J. Suykens, and et al., "A theoretical framework for target propagation," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 024–20 036.

[48] M. C. W. v. Rossum, "A novel spike distance," *Neural Computation*, vol. 13, no. 4, pp. 751–763, 2001.

[49] Y. LeCun, "The mnist database of handwritten digits," *http://yann.lecun.com/exdb/mnist/*, 1998.

[50] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Frontiers in Neuroscience*, vol. 9, p. 437, 2015.

[51] A. Amir, B. Taba, D. Berg, and et al., "A low power, fully event-based gesture recognition system," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7243–7252.

[52] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[53] W. Severa, C. M. Vineyard, R. Dellana, and et al., "Training deep neural networks for binary communication with the whetstone method," *Nature Machine Intelligence*, vol. 1, no. 2, pp. 86–94, 2019.

[54] G. Orchard, C. Meyer, R. Etienne-Cummings, and et al., "Hfirst: A temporal approach to object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 10, pp. 2028–2040, 2015.

[55] J. C. Tapson, G. K. Cohen, S. Afshar, and et al., "Synthesis of neural networks for spatio-temporal spike pattern recognition and processing," *Frontiers in Neuroscience*, p. 153, 2013.

[56] M. Shi, T. Zhang, and Y. Zeng, "A curiosity-based learning method for spiking neural networks," *Frontiers in Computational Neuroscience*, vol. 14, p. 7, 2020.

[57] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. IEEE, 2014, pp. 10–14.

[58] A. H. Marblestone, G. Wayne, and K. P. Kording, "Toward an integration of deep learning and neuroscience," *Frontiers in Computational Neuroscience*, vol. 10, p. 94, 2016.

[59] A. Banerjee, "Learning precise spike train-to-spike train transformations in multilayer feedforward neuronal networks," *Neural Computation*, vol. 28, no. 5, pp. 826–848, 2016.

[60] F. Zenke and W. Gerstner, "Limits to high-speed simulations of spiking neural networks using general-purpose computers," *Frontiers in Neuroinformatics*, vol. 8, p. 76, 2014.

[61] M. Stimberg, D. F. M. Goodman, V. Benichoux, and R. Brette, "Equation-oriented specification of neural models for simulations," *Frontiers in Neuroinformatics*, vol. 8, p. 6, 2014.

[62] E. Yavuz, J. Turner, and T. Nowotny, "Genn: a code generation framework for accelerated brain simulations," *Scientific Reports*, vol. 6, p. 18854, 2016.

[63] D. Zhang, T. Zhang, S. Jia, and B. Xu, "Multiscale dynamic coding improved spiking actor network for reinforcement learning," in *36th AAAI Conference on Artificial Intelligence*, Conference Proceedings.

Tielin Zhang received his Ph.D. degree from the Institute of Automation Chinese Academy of Sciences, Beijing, China, in 2016. He is an Associate Professor and PI in the laboratory of Cognition and Decision Intelligence for Complex Systems, at the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current interests include theoretical research on neural dynamics, spiking neural networks, and AI for Sciences.

Qingyu Wang is an M.D. candidate at the Institute of Automation, the Chinese Academy of Sciences, and the University of Chinese Academy of Sciences. His current interests include theoretical research on neural dynamics and spiking neural networks.

Bo Xu is a professor, the director of the Institute of Automation, Chinese Academy of Sciences, and deputy director of the Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences. His main research interests include brain-inspired intelligence, brain-inspired cognitive models, natural language processing and understanding, and brain-inspired robotics.