

Patterns

Backpropagation with biologically plausible spatiotemporal adjustment for training deep spiking neural networks

Highlights

- BPSA avoids the unnecessary influence from non-spiking neurons to the weights update
- BPTA helps transmit errors across spikes to enhance the temporal dependency
- BPSA and BPTA have improved the performance of BP-based SNNs
- BPSA and BPTA can reduce the energy consumption and latency of BP-based SNNs

Authors

Guobin Shen, Dongcheng Zhao,
Yi Zeng

Correspondence

yi.zeng@ia.ac.cn

In brief

The proposal of the surrogate gradient provides convenience for the training of SNNs. Most researchers often regard SNNs as a substitute for RNNs and ignore the characteristics of spiking neurons. This paper analyzes problems in the backpropagation-based SNNs and proposes a more biologically plausible spatiotemporal adjustment to help mitigate problems from the spatial and temporal dimensions. The adjustments significantly improve the performance of SNNs and reduce energy consumption and latency.



Article

Backpropagation with biologically plausible spatiotemporal adjustment for training deep spiking neural networks

Guobin Shen,^{1,4,6} Dongcheng Zhao,^{1,6} and Yi Zeng^{1,2,3,4,5,7,*}

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

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

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

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

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

⁶These authors contributed equally

⁷Lead contact

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

<https://doi.org/10.1016/j.patter.2022.100522>

THE BIGGER PICTURE The spiking neural network (SNN) captures more important aspects of brain information processing and has been applied to various domains. The biggest problem restricting the development of SNN is the training algorithm. Backpropagation (BP)-based training has extended SNNs to more complex network structures and datasets. However, the traditional design of BP ignores the dynamic characteristics of SNNs and is not biologically plausible. This paper rethinks the problems in BP-based SNNs and proposes a biologically plausible spatiotemporal adjustment to replace the traditional artificial design. The adjustment greatly improves the performance of the SNNs and reduces energy consumption and latency. The long-term ambition of this research is to take more inspiration on learning mechanisms and structures from the cognitive brain at different levels of details to build even more biologically plausible SNNs as a foundation for future artificial intelligence models.



Proof-of-Concept: Data science output has been formulated, implemented, and tested for one domain/problem

SUMMARY

The spiking neural network (SNN) mimics the information-processing operation in the human brain. Directly applying backpropagation to the training of the SNN still has a performance gap compared with traditional deep neural networks. To address the problem, we propose a biologically plausible spatial adjustment that rethinks the relationship between membrane potential and spikes and realizes a reasonable adjustment of gradients to different time steps. It precisely controls the backpropagation of the error along the spatial dimension. Secondly, we propose a biologically plausible temporal adjustment to make the error propagate across the spikes in the temporal dimension, which overcomes the problem of the temporal dependency within a single spike period of traditional spiking neurons. We have verified our algorithm on several datasets, and the experimental results have shown that our algorithm greatly reduces network latency and energy consumption while also improving network performance.

INTRODUCTION

Deep neural networks (DNNs) have achieved success in various research areas, such as object detection,¹ visual tracking,² face recognition,³ etc. However, they are still far away from the information-processing mechanisms of the human brain. Spiking

neural networks (SNNs) are known as the third-generation artificial neural network.⁴ They have been widely used in many fields, such as semantic segmentation,⁵ visual explanations,⁶ privacy protection,^{7,8} and object detection.⁹ The discrete spikes used to transmit information are more energy efficient and are more in line with the information-processing mechanism in the brain.



Combined with neuromorphic computing,¹⁰ it promises to realize real intelligence.

However, due to the complex neural dynamics and non-differential characteristics of SNNs, it is still a challenge to train SNNs efficiently. Existing SNN training methods can be roughly divided into three categories: the biologically plausible method, the conversion method, and the backpropagation-based method.

The biologically plausible method, such as Hebbian learning rules¹¹ and spike-timing-dependent plasticity (STDP),¹² is mainly inspired by the synaptic learning rules in the human brain. The Hebbian theory believes that the connection between pre- and post-synaptic neurons will increase due to continuous and repetitive stimulation of pre-synaptic neurons. STDP is an extended Hebbian learning rule based on the temporal difference between pre- and post-synaptic neurons. Diehl et al.¹³ used the STDP learning rule and lateral inhibition in a two-layer SNN and achieved 95% accuracy on the MNIST dataset. Saeed et al.¹⁴ introduced a weight-sharing strategy and designed a spiking convolutional neural network. The weight was learned by the STDP layer-wisely. Kherapisheh et al.¹⁵ used the hand-crafted difference of Gaussian (DoG) features as the input of the SNNs and trained the subsequently convolutional layer through STDP. These methods rely on the local activities of neighboring neurons to update network weights and lack the supervision of global signals. Although Zhao et al. designed a multi-layer SNN based on global feedback connections and local optimization learning rules (GLSNN),¹⁶ it still performs poorly when transplanted to some deep networks for some complex tasks.

The conversion method is an alternative way to get high-performance SNNs. It first trains the well-performed DNNs, then converts the DNNs into SNNs with some additional adjustments.^{17–21} The analog values of DNNs are converted into the firing rates of SNNs. Although the conversion method makes the SNNs achieve performance close to the traditional DNNs, the simulation time is too long, which causes the network to have poor real-time performance and high energy consumption. Also, the conversion methods rely highly on the well-trained DNNs and do not take advantage of the temporal information of SNNs.

The success of deep learning depends heavily on the proposal of the backpropagation algorithm. Several studies provide evidence for backpropagation in the brain. The feedback connections may make predictions of activities of low-level brain areas,^{22–25} and the biological neurons will backpropagate the action potentials to provide crucial signals for synaptic plasticity.^{26–29} Lillicrap et al.³⁰ argued that the differences with the feedforward and feedback neural activities may locally approximate the error signals in backpropagation. Researchers in SNN domains also introduced the backpropagation algorithm into the optimization of SNNs with the surrogate-gradient method.^{31–34} Surrogate gradient helps SNNs perform backpropagation through time (BPTT) so that SNNs can be adopted to larger-scale network structures, such as VGG, ResNet, etc., and perform better on more complex datasets. However, directly applying the surrogate gradient into the training of SNNs may lead to some problems. First, the surrogate gradient obtains the gradient by smoothing the spike firing function. Neurons with membrane potential around the threshold will participate in the backpropaga-

tion. As a result, the neurons that do not emit spikes may participate in weight updating, significantly increasing the network's energy consumption. Second, the spiking neuron will reset to the resting potential after the spike is emitted. The reset operation will cut off the error along the temporal dimension during the backpropagation so that errors cannot propagate across spikes, which significantly weakens the temporal dependence of the SNNs. To address the problems mentioned above, we introduced a biologically plausible spatiotemporal adjustment to improve the backpropagation training of SNNs, which can be summarized as follows:

- We study the influence of the surrogate gradient on the spatial dimension of the SNNs, rethink the relationship between the neuron membrane potential and the spikes, and propose a more biologically plausible spatial adjustment (BPSA) to help regulate spike activities.
- We study the limitations of the surrogate gradient in the temporal dimension and introduce a more biologically plausible temporal adjustment (BPTA), which enables the SNNs to propagate errors across the spikes, enhancing the temporal dependence of the SNNs.
- We conduct experiments on several commonly used datasets. For the static datasets MNIST, CIFAR10, and CIFAR100, we get remarkable performance compared with other state-of-the-art SNNs. To the best of our knowledge, we have reached state-of-the-art performance for the neuromorphic datasets N-MNIST, DVS-CIFAR10, and DVS-Gesture. For the Google Speech Commands dataset, we have reached comparable performance with other artificial neural networks designed for speech recognition. Moreover, our method dramatically reduces energy consumption and latency through analysis compared with other state-of-the-art SNNs.

RESULTS

In this section, we conduct experiments using the PyTorch framework³⁵ with NVIDIA A100 graphic processing unit (GPU). The network weights are initialized with the default method of PyTorch. We use the AdamW³⁶ algorithm as the optimizer, the learning rate lr is set with 1×10^{-3} , and the same learning rate control strategy as in SGDR³⁷ is used. The same method in temporal spike sequence-learning backpropagation (TSSL-BP) is used to warm up the model. The membrane potential threshold u_{th} of the neuron is set to 0.5, the membrane potential decay constant $\lambda = 0.9$, and the default simulation duration T is set to 16. The training epochs are set to 300. The α in Equation 10 is set to 0.2. First, we conduct experiments on the static MNIST, CIFAR10, and CIFAR100 datasets. To further illustrate the superiority of our algorithm, we also conduct experiments on the neuromorphic datasets N-MNIST,³⁸ DVS-Gesture,³⁹ and DVS-CIFAR10.⁴⁰ And to demonstrate the adaptability of our algorithm in other domains, we conduct experiments on the speech-recognition dataset Google Speech Commands.⁴¹ For the static datasets, we use the direct input encoding used in Wu et al.³² as well as the voting strategy. For the neuromorphic dataset, we use the same data preprocessing strategy used in SpikingJelly.⁴² For different datasets, we designed three

Table 1. Classification accuracy on MNIST, CIFAR10, and CIFAR100 datasets

Models	Training method	MNIST	CIFAR10	CIFAR100
Spiking CNN ⁴⁴	conversion	–	82.95	–
BackRes ⁴⁵	BP	–	84.98	–
ContinueSNN ⁴⁶	conversion	99.44	90.85	–
Spike-Norm ¹⁹	conversion	–	91.55	–
STBP ³¹	BP	99.42	50.7	–
HM2BP ³³	BP	99.49	–	–
LISNN ⁴⁷	BP	99.5	–	–
BNTT ⁴⁸	BP	–	90.5	66.6
STBP NeuNorm ³²	BP	–	90.53	–
BackEISNN ⁴⁹	BP	99.67	90.93	–
SBPSNN ⁴³	BP	99.59	90.95	–
TSSL-BP ³⁴	BP	99.53	91.41	–
ST-RSBP ⁵⁰	BP	99.62	–	–
RNL ⁵¹	conversion	99.51	93.45	75.1
SNASNet-Fw ⁵²	NAS + BP	–	93.64	70.06
SNASNet-Bw ⁵²	NAS + BP	–	94.12	73.04
Our method	BP	99.67	92.15	68.28
Our method	BP	–	94.51	69.32
ResNet34				

different network structures to adapt to different sizes and complexities. The small network is 128C3-MP2-128C3-256C3-MP2-2048FC-DP-10Voting, the middle is 128C3-MP2-128C3-MP2-256C3-MP2-512C3-AP4-512FC-10Voting, and the large is 128C3-128C3-MP2-128C3-MP2-256C3-MP2-512C3-MP2-1024C3-AP4-DP-1024FC-10Voting. AP denotes the average-pooling operation, MP denotes max-pooling operation, DP denotes neuron dropout,⁴³ and C denotes the Conv-BN-ReLU-LIF operation.

Static datasets

MNIST is one of the most common classification datasets in the deep-learning domain, with 60,000 training datasets and 10,000 test datasets. The samples in the datasets are 28 × 28 gray-scale images representing handwritten numbers from 0 to 9, respectively. We use the small structure for the evaluation. The CIFAR10 dataset is more challenging for most existing SNNs. The training set has 50,000 samples, and the test set has 10,000 samples. The dataset is a 32 × 32 color dataset. A deeper network will achieve better performance. Hence, we adopt the middle structure to conduct the experiment. CIFAR100 is a more challenging version than CIFAR10; it has 100 categories, and each category has only 600 samples: 500 for training and 100 for testing. The network structure is the same with CIFAR10. Experimental results are compared with several deep SNN models, including conversion and BP based, as shown in Table 1.

The spatiotemporal BP (STBP) NeuNorm³² is the STBP method with the neuron norm. For the normal network structures we set, our network achieves comparable performance with other SNN algorithms. Also, in order to illustrate the adaptability of our algorithm to deeper networks, we tested it based

Table 2. Classification accuracy on N-MNIST, DVS-Gesture, and DVS-CIFAR10 datasets

Models	Method	N-MNIST	DVS-Gesture	DVS-CIFAR10
HM2-BP ³³	BP	98.88	–	–
SLAYER ⁵³	BP	99.2	93.64	–
TSSL-BP 30 ³⁴	BP	99.28	–	–
IIRSNN ⁵⁴	BP	99.28	–	–
TSSL-BP 100 ³⁴	BP	99.4	–	–
STBP ³¹	BP	99.44	–	–
LISNN ⁴⁷	BP	99.45	–	–
STBP NeuNorm ³²	BP	99.53	–	60.5
BNTT ⁴⁸	BP	–	–	63.2
SALT ⁵⁵	BP	–	–	67.1
STBP-tdBN ⁵⁶	BP	–	96.87	67.8
LMCSNN ⁵⁷	BP	99.61	97.57	74.8
BackEISNN ⁴⁹	BP	99.57	–	–
Our method	BP	99.71	98.96	78.95

on the network structure ResNet34. As can be seen in the Table 1, for the CIFAR10 dataset, our network has reached state-of-the-art performance compared with other famous SNNs, whether based on BP or conversion. For the CIFAR100 dataset, although our network still has a little gap compared with RNL⁵¹ and SASNet,⁵² the RNL algorithm directly converts the well-trained DNNs to SNNs, while SNASNet searches a better network structure based on neural architecture search (NAS).

Neuromorphic datasets

To better illustrate our spatiotemporal adjustment, we conduct experiments on the neuromorphic datasets N-MNIST, DVS-Gesture, and DVS-CIFAR10. N-MNIST is the neuromorphic version of MNIST. The dynamic version sensor (DVS) is put in front of the static images on a computer screen. The images shift due to the DVS moving in the direction in three sides of the isosceles triangle in turn, and the two-channel spike event (on and off) is collected. DVS-Gesture is a real-time gesture-recognition dataset reported by DVS. The dataset has 11 hand gestures such as hand clips, arm rolls, etc., collected from 29 individuals under three illumination conditions. DVS-CIFAR10 is a neuromorphic version converted from the CIFAR10 dataset. 10,000 frame-based images are converted into 10,000 event streams with DVS. For N-MNIST, we use the middle structure, and for the DVS-Gesture and DVS-CIFAR10, which are more complex, we use the large structure.

As can be seen in Table 2, for the N-MNSIT dataset, our method has surpassed STBP by 0.3%; even with the introduction of NeuNorm, our work still performs better than them. For the more complex gesture dataset, our model surpasses the latest STBP-tdBN⁵⁶ by 2% and LMCSNN⁵⁷ by 1.4%. Our model has reached state-of-the-art performance compared with other current famous SNNs. For the DVS-CIFAR10 dataset, compared with the latest STBP-tdBN, we surpassed them by nearly 11%. For LMCSNN, which make many parameters in the leaky integrate-and-fire (LIF) spiking neurons learnable, we

Table 3. Classification accuracy on Google Speech Commands dataset

Models	Method	Accuracy
Sample-level ⁵⁸	DNN	92.53
Attention RNN ⁵⁹	DNN	93.9
Sample-level + SE ⁶⁰	DNN	93.95
Harmonic filters ⁶¹	DNN	96.39
Our method	SNN	94.2

also surpass them by 4%. Our method has achieved state-of-the-art performance for the DVS-CIFAR10 dataset.

Speech-recognition dataset

To verify the performance of our algorithm in other domains, we validate the proposed method on the Google Speech Commands dataset. There are two versions of this dataset, and the second version is used for testing. There are 105,000 utterances in 35 categories, and each utterance is 1 s long. The two training datasets are rebalanced by repeating random samples to make the number of samples the same in each class.

As can be seen in Table 3, even compared with the artificial neural networks designed for speech recognition, our algorithm still shows comparable performance.

Conclusion

In this paper, first, we analyze the existing problems in the SNNs trained with BP. We find that the current setting will cause the earlier spiking neurons repeat participating in the gradient calculation of the network, making a more significant influence on the network weight. The BPTT algorithm on the SNNs only propagates errors backward in a single-spike period. The temporal dependence between spikes will be truncated. By introducing the biologically plausible spatial adjustment, it will consider the spikes generated by the membrane potential of different strengths, which will have different effects on the parameter update during the backpropagation process. In addition, the biologically plausible temporal adjustment is introduced, and it considers the backpropagation across the spikes. We have achieved remarkable performance on MNIST, CIFAR10, CIFAR100, and Google Speech Commands datasets and achieved the current best performance on N-MNIST, DVS-Gesture, and DVS-CIFAR10 datasets. By analyzing the energy consumption and latency of the SNNs, we find that the BPSAs and BPTAs significantly reduces energy consumption and latency while improving performance.

DISCUSSION

In this section, firstly, we conduct the ablation study to the BPSA and BPTA mentioned above and analyze the contribution of each module. Secondly, we explore the energy consumption of the SNNs for these adjustments. Thirdly, we discuss the latency of the SNNs affected by these adjustments. Finally, we give the limitations of our algorithm and future work. Through the analysis, it is fully illustrated that the above two adjustments can make the behavior of the spiking neurons more stable and establish a better performance while reducing network latency and energy consumption.

Table 4. The ablation study of the two adjustments on DVS-Gesture and DVS-CIFAR10 datasets

	Baseline	BPSA	BPSA + BPTA
DVS-Gesture	93.92	97.56	98.96
DVS-CIFAR10	71.40	75.30	78.95

Ablation study

We conduct the ablation study on the neuromorphic datasets DVS-Gesture and DVS-CIFAR10 due to the more complex spatial structure and stronger temporal information, which will fully illustrate our adjustments' importance. We use Lillicrap et al.³¹ as our baseline and then continue to add the BPSA and BPTA.

As can be seen in Table 4, with the introduction of the two adjustments, the performance of the network is gradually improved, among which the spatial adjustment brings more significant improvement.

We also give the test curves of the DVS-Gesture dataset. As shown in Figure 1, with the number of epochs increasing, the accuracy of the model with biologically plausible spatiotemporal adjustment fluctuates less. Because with the introduction of the two adjustments, the firing pattern of neurons is more stable, making the model more robust to more minor parameter changes. Meanwhile, a reasonable gradient allocation strategy in the BP improves the model's generalization performance and avoids overfitting to a certain extent.

Energy-efficiency study

To illustrate the energy efficiency of our algorithm, we visualize the firing frequency of different layers in the MNIST experiment. As can be seen from the Figure 2, due to the biologically plausible spatiotemporal adjustment, our method exhibits an extremely low firing rate, especially in the initial convolutional layers.

We compare the accuracy and energy efficiency of the SNNs trained by the method used in Wu et al.,³¹ the model we propose,

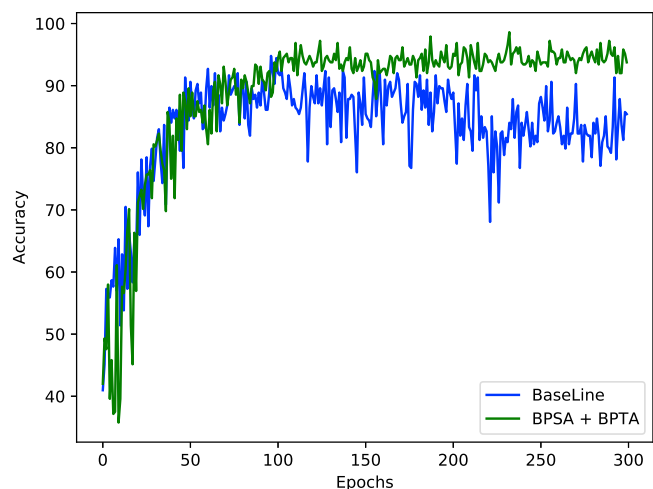


Figure 1. The test accuracy curve on DVS-Gesture of our method and the baseline

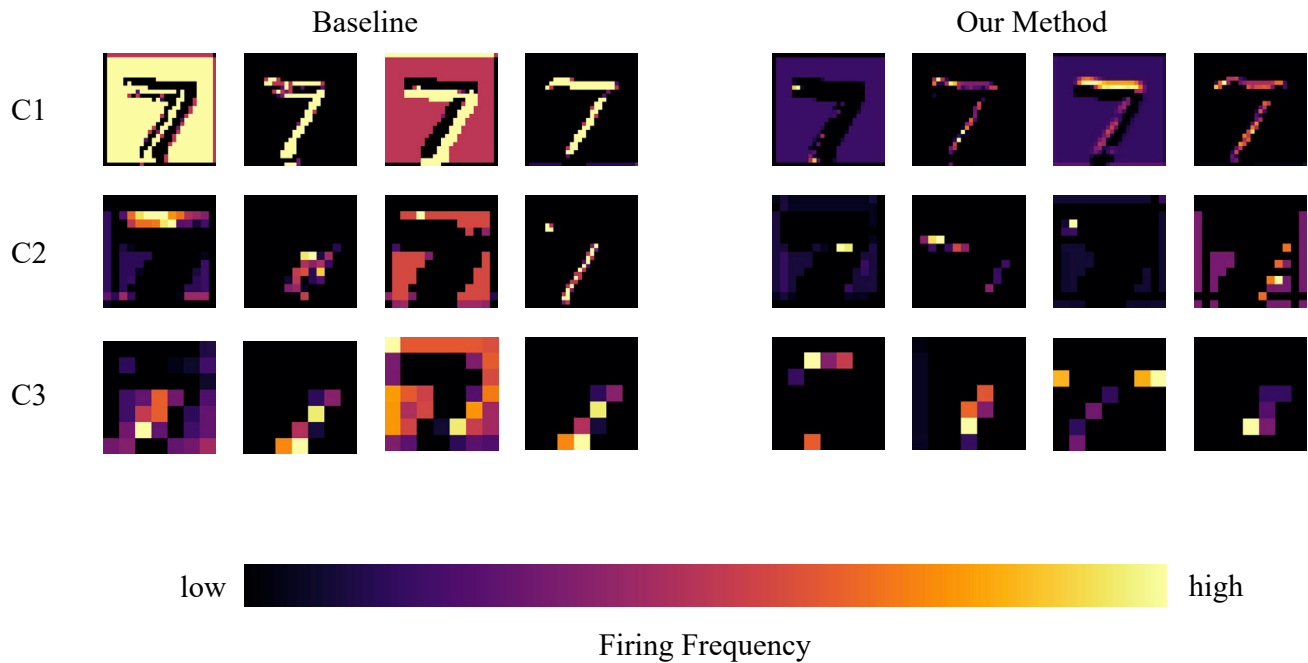


Figure 2. The firing frequency of different convolutional layers on MNIST of our method and the baseline

and the artificial neural networks (ANNs) using the same network structure and network parameters. Most operations in ANNs are multiply accumulate (MAC), while in SNNs, the spikes transmitted in the network are sparse, and the spikes are integrated into the membrane potential. As a result, most operations in SNNs are accumulate (AC) operations. We calculate the energy consumption of the SNN by multiplying floating-point operations (FLOPS) and the energy consumption of MAC and AC operations. We use the same energy-efficiency calculations as in Chakraborty et al.,⁶² and the computation details can be seen in Equation 1.

$$E_{ANN} = FLOPS_{ANN} \times E_{FL_MAC}$$

$$E_{SNN} = FLOPS_{SNN} \times E_{INT_AC} \times T \quad (\text{Equation 1})$$

As can be seen in Table 5, our method has a lower firing rate and higher energy efficiency. The training method of the SNNs proposed in this paper distributes the gradient more reasonably along the spatial and temporal dimensions, avoiding the problem that the earlier spiking neurons would have a more significant influence on the network parameters. The cross-spikes propagation will also enhance the temporal dependence of the SNNs. Therefore, the method proposed in this paper achieves lower network power consumption while maintaining a higher accuracy.

Latency study

The latency of the SNNs is one of the main problems that restricts the development of SNNs. The spiking neurons need to accumulate membrane potential, and once they reach the threshold, they fire spikes and transmit information. Therefore, SNNs often require a long simulation time to achieve higher per-

formance. Here, we study the influence of different simulation lengths on the network performance.

As shown in the Figure 3, when our adjustments are not introduced, when the simulation time is reduced, the test curve of the network is not very smooth, that is, the network needs a long simulation time to converge. As can be seen in Table 6, with the introduction of the two adjustments, our training method still achieves high accuracy while reducing the simulation time. The low latency of our approach further lays the foundation for the practical application of SNNs.

Limitations of the study

In this paper, through the analysis of the training of the BP-based SNN, we find that neurons that do not generate spikes will still participate in the update of network weights. Also, the error signals along the temporal dimension cannot propagate across the spikes due to the reset operation. By introducing the BPSA and BPTA mechanisms, our network is more consistent with the brain in terms of weight update, and the energy consumption and latency

Table 5. The energy-efficiency study of our model with baseline on different datasets

Dataset	Accuracy (%)	Firing rate	EE = $\frac{E_{ANN}}{E_{SNN}}$ (x)
MNIST	99.58/99.42	0.082/0.183	35.1/15.7
N-MNIST	99.61/99.32	0.097/0.176	29.6/16.3
CIFAR10	92.33/89.49	0.108/0.214	26.6/13.4
DVS-Gesture	98.26/93.92	0.083/0.165	34.6/17.4
DVS-CIFAR10	77.76/71.40	0.097/0.177	29.5/16.2

Represented as baseline/our method.

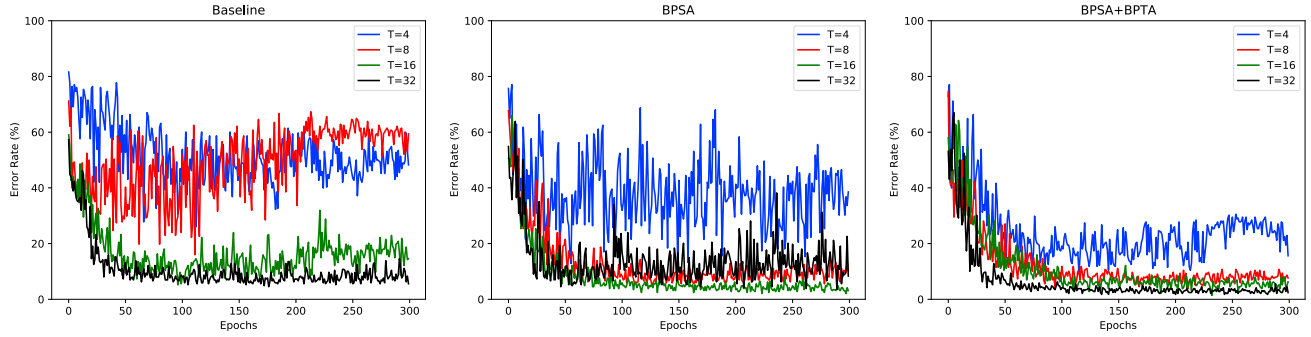


Figure 3. The test accuracy of different simulation lengths on DVS-Gesture dataset with our method and the baseline

of the SNN are greatly reduced. However, there is no independent module in the brain specially designed for the BP pathway. In future work, we will explore more biologically plausible learning methods to train SNNs with high performance and robustness.

EXPERIMENTAL PROCEDURES

Resource availability

Lead contact

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

Materials availability

This study did not generate new unique materials.

Data and code availability

All original code has been deposited at <https://github.com/Brain-Inspired-Cognitive-Engine/BP-STA> under <https://doi.org/10.5281/zenodo.6489856> and is publicly available as of the date of publication.

Spiking-neuron model

Many spiking-neuron models with biological neural characteristics have been proposed in recent years, and the LIF model is adopted in most common neuron models in deep SNNs. The LIF neurons continuously accumulate the membrane potential and emit spikes once they reach the threshold. We give a detailed description of the LIF neuron models. As shown in Equation 2, the membrane potential of the neuron changes dynamically with the input current.

$$\tau \frac{du_i(t)}{dt} = -u_i(t) + R I_i(t) \quad (\text{Equation 2})$$

$I_i(t)$ denotes the input current, which is composed of input spikes. R is the membrane resistance, and $\tau = RC$ is the synaptic time constant. When the membrane potential is greater than the threshold u_{th} , the neuron will spike and be reset to u_{reset} . Without loss of generality, we set the reset potential $u_{reset} = 0, C = 1$. To facilitate the calculation and simulation, we convert Equation 2 into a discrete form with Euler method with $dt = 1$ so that we can get

$$u_i[t + 1] - u_i[t] = -\frac{u_i(t)}{\tau} + I_i(t) \quad (\text{Equation 3})$$

The input $I_i(t)$ can be obtained from the pre-synaptic spikes $\sum_{j=1}^{M_{l-1}} w_{ji} o_j^{-1}[t]$, M_{l-1} is the number of neurons in the $l - 1$ layer, then we can get

$$u_i[t + 1] = \lambda u_i[t] + \sum_{j=1}^{M_{l-1}} w_{ji} o_j^{-1}[t]$$

$$o_i[t + 1] = g(u_i[t + 1]) = 1, \quad u_i[t + 1] = 0, \quad \text{if } u > u_{th} \quad (\text{Equation 4})$$

$\lambda = 1 - \frac{1}{\tau}$, and the function g is the threshold function. w_{ji} is the synaptic weight from the l^{th} layer from neuron j to neuron i . $o_j^{-1}[t]$ denotes the neuron j spikes in l^{th} layer at time t .

Spatiotemporal characteristics of SNNs

The discontinuity of the spike firing function makes it challenging to apply the BP directly to the training of SNNs. In recent years, surrogate gradient has been proposed to replace the discontinuous gradient with a smooth gradient function to enable the SNNs to conduct BP in the spatial and temporal domains. Here, we use the mean average firing rates of the last layer to approximate the classification label and train the network through the mean squared error (MSE):

$$L = \frac{1}{S} \sum_{s=1}^S \left\| y_s - \frac{1}{T} \sum_{t=1}^T o_t \right\|^2. \quad (\text{Equation 5})$$

T denotes the simulation length, y_s denotes the real labels, and the o_t denotes the output at time t . By applying chain rule, we can obtain the gradient with respect to weight:

$$\begin{aligned} \frac{\partial L}{\partial w'} &= \sum_{t=1}^T \frac{\partial L}{\partial o_t[t]} \frac{\partial o_t[t]}{\partial u_i[t]} \frac{\partial u_i[t]}{\partial w'} \\ &= \sum_{t=1}^T \delta_i^l[t] g'(u_i[t]) o_i^{-1}[t] \end{aligned} \quad (\text{Equation 6})$$

$$\delta_i^l[t] = \sum_{j=1}^M \frac{\partial L}{\partial o_j^{l+1}[t]} \frac{\partial o_j^{l+1}[t]}{\partial o_i[t]} + \frac{\partial L}{\partial o_i[t+1]} \frac{\partial o_i[t+1]}{\partial u_i[t]} \quad (\text{Equation 7})$$

$\delta_i^l[t]$ denotes the derivative with respect to o in the l^{th} layer at time step t and can be derived from the $(l + 1)^{th}$ layer (spatial) and $t + 1$ time step (temporal).

As can be seen in Equation 6 and Figure 4, the traditional surrogate-gradient method will calculate the gradient around the threshold, even if the spiking neurons do not emit spikes in the forward process. This will cause a large number of neurons that do not emit spikes to participate in the parameter update, increasing network's energy consumption. Also, as can be seen in Figure 4, for the neuron o_i^{-1} , it will participate in the weight update repeatedly according to the chain rule, and the earlier spiking moment $o_i^{-1}[T - 1]$ will have a larger influence on the weight update compared with $o_i^{-1}[T]$. While in neurophysiology, the farther away the spiking activity is from the current moment, the smaller the effect.

For an SSN trained with BP, the temporal dependence mainly comes from accumulating membrane potential over time. As a result, the backward process for the temporal dimension can be written as

$$\frac{\partial o_i[t+1]}{\partial u_i[t]} = \frac{\partial o_i[t+1]}{\partial u_i[t+1]} \frac{\partial u_i[t+1]}{\partial u_i[t]}. \quad (\text{Equation 8})$$

Table 6. The test accuracy on DVS-Gesture dataset of different simulation lengths of our method and the baseline

	T = 32	T = 16	T = 8	T = 4
BPSA + BPTA	98.27	98.26	96.18	92.01
BPSA	96.53	97.56	94.44	89.58
Baseline	95.49	93.92	84.03	73.96

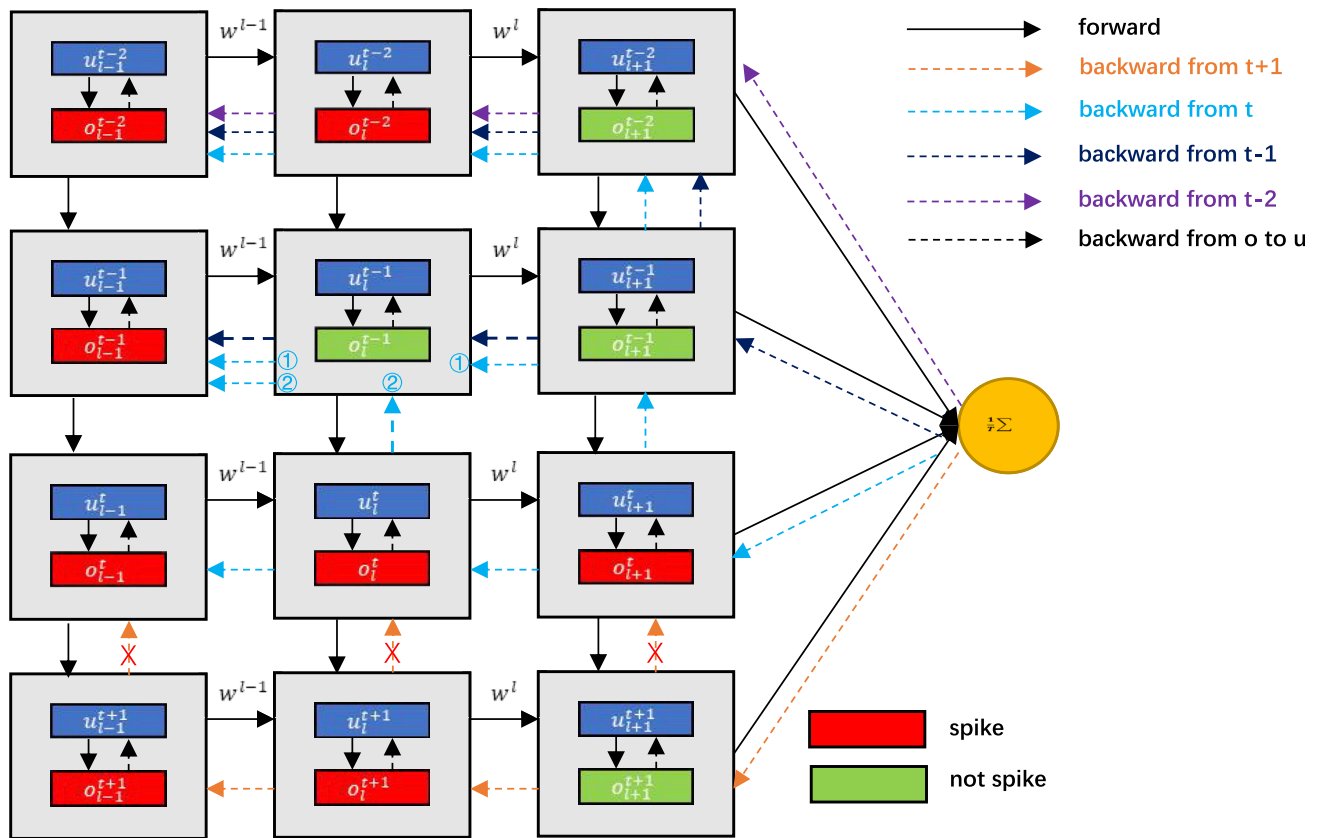


Figure 4. The forward and backward process of spiking neural networks

The dotted lines of different colors indicate the impact on the network at different time steps. The earlier spiking node will have more influence on the parameter update.

Since the spiking neurons will reset to the resting potential after reaching the threshold, that is to say that $u_i^l[t + 1]$ will have no relationship with $u_i^l[t]$, and the temporal dependence will no longer exist, as shown in Figure 5.

To tackle the problems mentioned above, we propose the BPSA in which the neurons along with the hierarchical layers that emit spikes will participate in the weight update. Also, we propose the BPTA to help the errors transmit to the initial time step without being clipped.

BPSA

The membrane potential of spiking neurons changes as a process of information accumulation. After the neurons have accumulated enough information, they will send the information to the post-synaptic neurons in the form of spikes. As a result, the binary spikes can be regarded as a normalization of

the information contained in the membrane potential. For the BP process, it is more reasonable to only calculate the gradient of the neuron at the moment of spiking to the membrane potential. We propose a BPSA to improve the BP-based training SNNs. When the membrane potential does not reach the threshold, we will clip the gradient of the spikes to the membrane potential to avoid the problems of repeated updates at an earlier time, as in Figure 4. When the membrane potential reaches the threshold, we normalize the membrane potential and spread the information in spikes. Then, the derivative of the spikes concerning the membrane potential can be expressed as

$$\frac{\partial o_i^l[t]}{\partial u_i^l[t]} = \begin{cases} \frac{1}{u_i^l[t]}, & o_i^l[t] = 1 \\ 0, & \text{otherwise} \end{cases} \quad (\text{Equation 9})$$

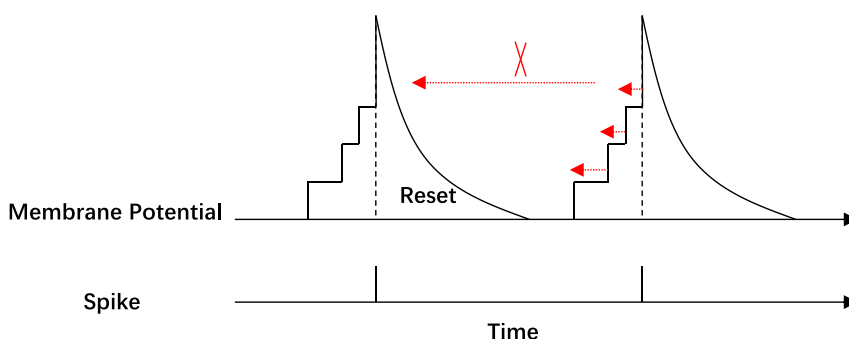


Figure 5. The temporal backpropagation of LIF neurons

The information can only propagate within a single-spike period and cannot propagate cross spikes.

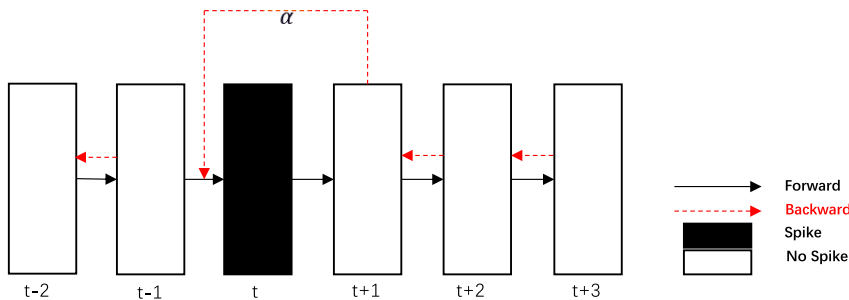


Figure 6. The temporal residual pathway helps the error transfer from time step $t + 1$ to time step t

This method considers the influence of the spikes generated by the membrane potential of different strengths on the parameter update. For a spike excited by larger membrane potential, there will be a minor optimization step for the model parameters in the BP process to ensure the stability of the spikes. The spikes excited by the membrane potential near the spike threshold u_{th} will have a more significant impact on the model parameters, allowing the model to quickly push the membrane potential close to the threshold away to obtain more stable spikes.

BPTA

In biological neurons, the spike that the neuron fires will affect the subsequent spikes of the neuron. When directly using the BP algorithm to optimize the parameters of the SNNs, the gradient of the loss function to the neuron output will only be propagated from the time the neuron was last excited to the present and will not cross the spikes as shown in Equation 8 and Figure 5. So, the influence between spikes will not be considered in the temporal dimension. Then, we propose a BPTA cross the spikes. Considering that the temporal dependence disappears during the BP process, we add the residual connection between spikes during the backward pathway, as shown in Figure 6. The influence to control the error transfer from time step $t + 1$ to t is controlled by the residual factor α . The temporal feedback process can be written as

$$\frac{\partial o_i[t+1]}{\partial u_i[t]} = g'(u_i[t+1])\lambda[(1 - o_i[t]) + \alpha g'(u_i[t])] \quad (\text{Equation 10})$$

As can be seen in Equation 10, when the neurons do not emit a spike at time t , $g'(u_i[t]) = 0$, which is the same with the traditional BP algorithm. However, when the neuron fires a spike at time t , then $1 - o_i[t] = 0$, the temporal dependence can be written as $g'(u_i[t+1])\lambda\alpha g'(u_i[t])$. With the introduction of the BPSAs and BPTAs, the influence of different spikes becomes more reasonable, and the temporal residual backward pathway enables it to propagate errors over spikes.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program (grant no. 2020AAA0104305) and the Strategic Priority Research Program of the Chinese Academy of Sciences (grant no. XDB32070100).

AUTHOR CONTRIBUTIONS

Conceptualization, G.S., D.Z., and Y.Z.; methodology, G.S. and D.Z.; investigation, G.S.; writing – original draft, G.S., D.Z., and Y.Z.; writing – review & editing, G.S., D.Z., and Y.Z.; supervision, Y.Z.; funding acquisition, Y.Z.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: January 24, 2022

Revised: March 29, 2022

Accepted: May 6, 2022

Published: June 2, 2022

REFERENCES

- Zou, Z., Shi, Z., Guo, Y., and Ye, J. (2019). Object detection in 20 years: a survey. Preprint at arXiv. <https://doi.org/10.48550/arXiv.1905.05055>.
- Li, P., Wang, D., Wang, L., and Lu, H. (2018). Deep visual tracking: review and experimental comparison. *Pattern Recogn.* 76, 323–338. <https://doi.org/10.3389/fnins.2020.00119>.
- Masi, I., Wu, Y., Hassner, T., and Natarajan, P. (2018). Deep Face Recognition: A Survey. In 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI) (IEEE), pp. 471–478. <https://doi.org/10.1109/SIBGRAPI.2018.00067>.
- Maass, W. (1997). Networks of spiking neurons: the third generation of neural network models. *Neural Network.* 10, 1659–1671. [https://doi.org/10.1016/S0893-6080\(97\)00011-7](https://doi.org/10.1016/S0893-6080(97)00011-7).
- Kim, Y., Chough, J., and Panda, P. (2021). Beyond classification: directly training spiking neural networks for semantic segmentation. Preprint at arXiv. <https://doi.org/10.48550/arXiv>.
- Kim, Y., and Panda, P. (2021). Visual explanations from spiking neural networks using inter-spike intervals. *Sci. Rep.* 11, 1–14. <https://doi.org/10.1038/s41598-021-98448-0>.
- Kim, Y., Venkatesha, Y., and Panda, P. (2021). Privatesnn: fully privacy-preserving spiking neural networks. Preprint at arXiv. <https://doi.org/10.48550/arXiv.2104.03414>.
- Venkatesha, Y., Kim, Y., Tassiulas, L., and Panda, P. (2021). Federated learning with spiking neural networks. *IEEE Trans. Signal. Process.* 69, 6183–6194. <https://doi.org/10.1109/TSP.2021.3121632>.
- Kim, S., Park, S., Na, B., and Yoon, S. (2020). Spiking-yolo: spiking neural network for energy-efficient object detection. *Proc. AAAI Conf. Artif. Intell.* 34, 11270–11277.
- Roy, K., Jaiswal, A., and Panda, P. (2019). Towards spike-based machine intelligence with neuromorphic computing. *Nature* 575, 607–617. <https://doi.org/10.1038/s41586-019-1677-2>.
- Hebb, D.O. (1949). *The Organization of Behavior* (Wiley).
- Bi, G.q., and Poo, M.m. (1998). Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci.* 18, 10464–10472. <https://doi.org/10.1523/JNEUROSCI.18-24-10464.1998>.
- Diehl, P.U., and Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* 9, 99. <https://doi.org/10.3389/fncom.2015.00099>.
- Kheradpisheh, S.R., Ganjtabesh, M., and Masquelier, T. (2016). Bio-inspired unsupervised learning of visual features leads to robust invariant object recognition. *Neurocomputing* 205, 382–392. <https://doi.org/10.1016/j.neucom.2016.04.029>.
- Kheradpisheh, S.R., Ganjtabesh, M., Thorpe, S.J., and Masquelier, T. (2018). Stp-based spiking deep convolutional neural networks for object recognition. *Neural Network.* 99, 56–67. <https://doi.org/10.1016/j.neunet.2017.12.005>.

16. Zhao, D., Zeng, Y., Zhang, T., Shi, M., and Zhao, F. (2020). Glsnn: a multi-layer spiking neural network based on global feedback alignment and local stdp plasticity. *Front. Comput. Neurosci.* *14*, 576841. <https://doi.org/10.3389/fncom.2020.576841>.
17. Diehl, P.U., Neil, D., Binas, J., Cook, M., Liu, S.C., and Pfeiffer, M. (2015). Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International Joint Conference on Neural Networks (IJCNN) (IEEE)*, pp. 1–8.
18. Xu, Q., Qi, Y., Yu, H., Shen, J., Tang, H., and Pan, G. (2018). Csnn: An Augmented Spiking Based Framework with Perceptron-Inception (IJCAI), pp. 1646–1652.
19. Sengupta, A., Ye, Y., Wang, R., Liu, C., and Roy, K. (2019). Going deeper in spiking neural networks: vgg and residual architectures. *Front. Neurosci.* *13*, 95. <https://doi.org/10.3389/fnins.2019.00095>.
20. Hu, Y., Tang, H., Wang, Y., and Pan, G. (2018). Spiking deep residual network. Preprint at arXiv. <https://doi.org/10.48550/arXiv.1805.01352>.
21. Li, Y., Zeng, Y., and Zhao, D. (2021). Bsnn: towards faster and better conversion of artificial neural networks to spiking neural networks with bistable neurons. Preprint at arXiv. <https://doi.org/10.48550/arXiv.2105.12917>.
22. Rao, R.P., and Ballard, D.H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nat. Neurosci.* *2*, 79–87. <https://doi.org/10.1038/4580>.
23. Bastos, A.M., Usrey, W.M., Adams, R.A., Mangun, G.R., Fries, P., and Friston, K.J. (2012). Canonical microcircuits for predictive coding. *Neuron* *76*, 695–711. <https://doi.org/10.1016/j.neuron.2012.10.038>.
24. Kok, P., and Lange, F.P.d. (2015). Predictive coding in sensory cortex. In *An introduction to model-based cognitive neuroscience* (Springer), pp. 221–244. https://doi.org/10.1007/978-1-4939-2236-9_11.
25. Roelfsema, P.R., and Holtmaat, A. (2018). Control of synaptic plasticity in deep cortical networks. *Nat. Rev. Neurosci.* *19*, 166–180. <https://doi.org/10.1038/nrn.2018.6>.
26. Bereshpolova, Y., Amitai, Y., Gusev, A.G., Stoelzel, C.R., and Swadlow, H.A. (2007). Dendritic backpropagation and the state of the awake neocortex. *J. Neurosci.* *27*, 9392–9399. <https://doi.org/10.1523/JNEUROSCI.2218-07.2007>.
27. Schiess, M., Urbanczik, R., and Senn, W. (2016). Somato-dendritic synaptic plasticity and error-backpropagation in active dendrites. *PLoS Comput. Biol.* *12*, e1004638. <https://doi.org/10.1371/journal.pcbi.1004638>.
28. Richards, B.A., and Lillicrap, T.P. (2019). Dendritic solutions to the credit assignment problem. *Curr. Opin. Neurobiol.* *54*, 28–36. <https://doi.org/10.1016/j.conb.2018.08.003>.
29. Fitzsimonds, R.M., Song, H.J., and Poo, M.m. (1997). Propagation of activity-dependent synaptic depression in simple neural networks. *Nature* *388*, 439–448. <https://doi.org/10.1038/41267>.
30. Lillicrap, T.P., Santoro, A., Marris, L., Akerman, C.J., and Hinton, G. (2020). Backpropagation and the brain. *Nat. Rev. Neurosci.* *21*, 335–346. <https://doi.org/10.1038/s41583-020-0277-3>.
31. Wu, Y., Deng, L., Li, G., Zhu, J., and Shi, L. (2018). Spatio-temporal backpropagation for training high-performance spiking neural networks. *Front. Neurosci.* *12*, 331. <https://doi.org/10.3389/fnins.2018.00331>.
32. Wu, Y., Deng, L., Li, G., Zhu, J., Xie, Y., and Shi, L. (2019). Direct training for spiking neural networks: faster, larger, better. *Proc. AAAI Conf. Artif. Intell.* *33*, 1311–1318.
33. Jin, Y., Zhang, W., and Li, P. (2018). Hybrid macro/micro level backpropagation for training deep spiking neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7005–7015.
34. Zhang, W., and Li, P. (2020). Temporal spike sequence learning via backpropagation for deep spiking neural networks. *Adv. Neural Inf. Process. Syst.* *33*, 12022–12033.
35. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: an imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* *32*. <https://doi.org/10.48550/arXiv.1912.01703>.
36. Loshchilov, I., and Hutter, F. (2018). Decoupled weight decay regularization. In *International Conference on Learning Representations*. <https://doi.org/10.48550/arXiv.1711.05101>.
37. Loshchilov, I., and Hutter, F. (2016). Sgdr: stochastic gradient descent with warm restarts. Preprint at arXiv. <https://doi.org/10.48550/arXiv.1608.03983>.
38. Orchard, G., Jayawant, A., Cohen, G.K., and Thakor, N. (2015). Converting static image datasets to spiking neuromorphic datasets using saccades. *Front. Neurosci.* *9*, 437. <https://doi.org/10.3389/fnins.2015.00437>.
39. Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., Nayak, T., Andreopoulos, A., Garreau, G., Mendoza, M., et al. (2017). A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7243–7252.
40. Li, H., Liu, H., Ji, X., Li, G., and Shi, L. (2017). Cifar10-dvs: an event-stream dataset for object classification. *Front. Neurosci.* *11*, 309. <https://doi.org/10.3389/fnins.2017.00309>.
41. Warden, P. (2018). Speech commands: a dataset for limited-vocabulary speech recognition. Preprint at arXiv. <https://doi.org/10.48550/1804.03209>.
42. Fang, W., Chen, Y., Ding, J., Chen, D., Yu, Z., Zhou, H., and Tian, Y. (2020). Other Contributors. Spikingjelly.
43. Lee, C., Sarwar, S.S., Panda, P., Srinivasan, G., and Roy, K. (2020). Enabling spike-based backpropagation for training deep neural network architectures. *Front. Neurosci.* *14*, 119.
44. Hunsberger, E., and Elias Smith, C. (2015). Spiking deep networks with lif neurons. Preprint at arXiv. <https://doi.org/10.48550/1510.08829>.
45. Panda, P., Aketi, S.A., and Roy, K. (2020). Toward scalable, efficient, and accurate deep spiking neural networks with backward residual connections, stochastic softmax, and hybridization. *Front. Neurosci.* *14*, 653. <https://doi.org/10.3389/fnins.2020.00653>.
46. Rueckauer, B., Lungu, I.A., Hu, Y., Pfeiffer, M., and Liu, S.C. (2017). Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Front. Neurosci.* *11*, 682. <https://doi.org/10.3389/fnins.2017.00682>.
47. Cheng, X., Hao, Y., Xu, J., and Xu, B. (2020). Lissn: Improving Spiking Neural Networks with Lateral Interactions for Robust Object Recognition (IJCAI), pp. 1519–1525.
48. Kim, Y., and Panda, P. (2020). Revisiting batch normalization for training low-latency deep spiking neural networks from scratch. *Front. Neurosci.* *15*, 773954. <https://doi.org/10.3389/fnins.2021.773954>.
49. Zhao, D., Zeng, Y., and Li, Y. (2021). Backeissn: a deep spiking neural network with adaptive self-feedback and balanced excitatory-inhibitory neurons. Preprint at arXiv. <https://doi.org/10.48550/arXiv.2105.13004>.
50. Zhang, W., and Li, P. (2019). Spike-train level backpropagation for training deep recurrent spiking neural networks. *Adv. Neural Inf. Process. Syst.* *32*. <https://doi.org/10.48550/arXiv.1908.06378>.
51. Ding, J., Yu, Z., Tian, Y., and Huang, T. (2021). Optimal ann-snn conversion for fast and accurate inference in deep spiking neural networks. Preprint at arXiv. <https://doi.org/10.48550/2105.11654>.
52. Kim, Y., Li, Y., Park, H., Venkatesha, Y., and Panda, P. (2022). Neural architecture search for spiking neural networks. Preprint at arXiv. <https://doi.org/10.48550/2201.10355>.
53. Shrestha, S.B., and Orchard, G. (2018). Slayer: spike layer error reassignment in time. *Adv. Neural Inf. Process. Syst.* *31*.
54. Fang, H., Shrestha, A., Zhao, Z., and Qiu, Q. (2020). Exploiting neuron and synapse filter dynamics in spatial temporal learning of deep spiking neural network. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. <https://doi.org/10.48550/arXiv.2003.02944>.

55. Kim, Y.S., and Panda, P. (2021). Optimizing Deeper Spiking Neural Networks for Dynamic Vision Sensing. *Neural Network*. *144*, 686–698. <https://doi.org/10.1016/j.neunet.2021.09.022>.
56. Zheng, H., Wu, Y., Deng, L., Hu, Y., and Li, G. (2021). Going deeper with directly-trained larger spiking neural networks. *Proc. AAAI Conf. Artif. Intell.* *35*, 11062–11070.
57. Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., and Tian, Y. (2021). Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2661–2671.
58. Lee, J., Park, J., Kim, K.L., and Nam, J. (2017). Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. Preprint at arXiv. <https://doi.org/10.48550/arXiv.1703.01789>.
59. de Andrade, D.C., Leo, S., Viana, M.L.D.S., and Bernkopf, C. (2018). A neural attention model for speech command recognition. Preprint at arXiv. <https://doi.org/10.48550/arXiv.1808.08929>.
60. Kim, T., Lee, J., and Nam, J. (2019). Comparison and analysis of samplecnn architectures for audio classification. *IEEE J. Sel. Top.Signal. Process.* *13*, 285–297. <https://doi.org/10.1109/JSTSP.2019.2909479>.
61. Won, M., Chun, S., Nieto, O., and Serrc, X. (2020). Data-driven harmonic filters for audio representation learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (IEEE)*, pp. 536–540.
62. Chakraborty, B., She, X., and Mukhopadhyay, S. (2021). A fully spiking hybrid neural network for energy-efficient object detection. Preprint at arXiv. <https://doi.org/10.48550/2104.10719>.