Adaptive Gradient Learning for Spiking Neural Networks by Exploiting Membrane Potential Dynamics

Jiaqiang Jiang[1] , **Lei Wang**[1] , **Runhao Jiang**[2] , **Jing Fan**[1] and **Rui Yan**[1*]
[1]College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China
[2]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
{jqjiang, LeiWang23}@zjut.edu.cn, RhJiang@zju.edu.cn, {fanjing, ryan}@zjut.edu.cn

Abstract

Recent advancements have focused on directly training high-performance spiking neural networks (SNNs) by estimating the approximate gradients of spiking activity through a continuous function with constant sharpness, known as surrogate gradient (SG) learning. However, as spikes propagate within neurons and among layers, the distribution of membrane potential dynamics (MPD) will deviate from the gradient-available interval of fixed SG, hindering SNNs from searching the optimal solution space. To maintain the stability of gradient flows, SG needs to align with evolving MPD. Here, we propose a novel adaptive gradient learning for SNNs by exploiting MPD, namely MPD-AGL. It fully accounts for the underlying factors contributing to membrane potential shifts and establishes a dynamic association between SG and MPD at different timesteps to relax gradient estimation, which provides a new degree of freedom for SG learning. Experimental results demonstrate that our method achieves excellent performance at low latency. Moreover, it increases the proportion of neurons that fall into the gradient-available interval compared to fixed SG, effectively mitigating the gradient vanishing problem. Code is available at https://github.com/jqjiang1999/MPD-AGL.

1 Introduction

As a new paradigm with biological plausibility and computational efficiency, spiking neural networks (SNNs) achieve unique sparse coding and asynchronous information processing by modeling the spike firing and temporal dynamics of biological neurons. Instead of artificial neural networks (ANNs) that work with continuous activation and multiply-and-accumulate (MAC) operations, SNNs operate with threshold firing and accumulate (AC) operations, which allow low-latency inference and low-power computation on neuromorphic hardware [Pei *et al.*, 2019; Ma *et al.*, 2024]. Nowadays, with the development of SNNs, it has exhibited high potential in many applications, such as event-based image classification [Yang *et al.*, 2024; Liang *et al.*, 2024], object detection [Wang *et al.*, 2025c; Wang *et al.*, 2025b], reinforcement learning [Qin *et al.*, 2023; Qin *et al.*, 2025], etc. Backpropagation-based learning is a favorable methodology for training high-performance SNNs [Huh and Sejnowski, 2018; Dampfhoffer *et al.*, 2024]. Nevertheless, the discontinuous nature of spiking neurons hinders the direct application of gradient descent in SNNs. To tackle the non-differentiability of spike activity, surrogate gradient (SG) methods employ a smooth curve to distribute the gradient of output signals into a group of analog items in temporal neighbors [Zhang and Li, 2020]. Unfortunately, as spikes propagate in the spatio-temporal domain (STD), the distribution of membrane potential will shift and may not align with the gradient-available interval of fixed SG, leading to gradient vanishing or mismatch problems [Guo *et al.*, 2022c].

In Fig. 1, the main reason for gradient vanishing or mismatch problems is that the overlap area between the evolving membrane potential dynamics (MPD) and the gradient-available interval of fixed SG becomes too narrow or too wide. On the one hand, the limited overlap area causes many membrane potentials to fall into the area with zero approximate derivatives, leading to gradient propagation blockage. On the other hand, when the overlap area saturates, neurons contribute many inaccurate approximate gradients, enlarging the error with true gradients. To match SG and MPD, two groups of methods have been developed: (1) membrane potential regulation and (2) SG optimization. Membrane potential regulation methods redistribute the membrane potential before firing [Guo *et al.*, 2022b] or define a distribution loss to rectify it [Guo *et al.*, 2022c; Wang *et al.*, 2025a], aiming to balance the distribution to minimize the undesired shifts, but this increases the inference burden or requires more parameters and computations. By contrast, SG optimization methods update the SG by capturing the direction of accurate gradients that can automatically calibrate the SG sharpness in response to MPD for better gradient estimation [Guo *et al.*, 2022a; Wang *et al.*, 2023; Wang *et al.*, 2025a]. However, most of these methods either focus only on regulating membrane potentials or only on optimizing SG, ignoring their correlation, which cannot effectively control their alignment across the entire timestep. Moreover, the lack of comprehensive analysis regarding the causes of membrane potential shifts leaves room for improvement in these methods.

[*] Corresponding author

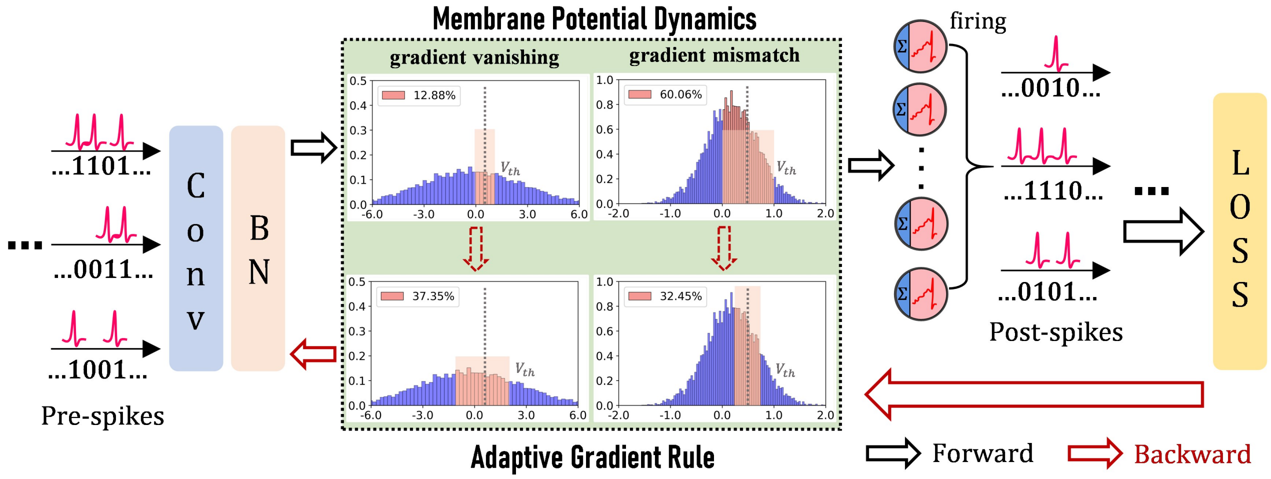

Figure 1: The overall framework of MPD-AGL. Pre-spikes are passed through the convolutional and normalization layers and then injected into spiking neurons to compute membrane potentials and fire spikes. The distribution of evolving MPD in forward propagation may not align with the fixed SG, leading to gradient vanishing or mismatch problems in backward propagation. Instead, the proposed adaptive gradient rule can synchronously adjust the width of SG to respond to evolving MPD during the entire timestep.

The reason for the membrane potential shifts and how to optimize SG to align with the evolving MPD in SNN learning are our main concerns. In this work, we propose an adaptive gradient learning algorithm for SNNs by exploiting MPD. Specifically, we realized that the affine transformation in normalization layers would force the pre-synaptic input to deviate from the desired distribution, affecting the distribution of MPD, which is the main cause of membrane potential shifts. Considering the influence of affine transformation, we derive the specific distribution of MPD at different timesteps during forward propagation and accordingly design a correlation function between SG and MPD to dynamically optimize SG, capturing the evolving MPD. The overall framework of our method is illustrated in Fig. 1. In summary, the main contributions of this work can be summarized as follows:

- We provide a new perspective for understanding the membrane potential shifts in SNN forward propagation by analyzing the effect of learnable affine transformation in the normalization layers on the distribution of MPD.
- We propose an adaptive gradient rule that synchronously adjusts the gradient-available interval of SG in response to the distribution of membrane potentials at different timesteps, aligning with the evolving MPD.
- Extensive experiments on four datasets CIFAR10, CIFAR100, CIFAR10-DVS, and Tiny-ImageNet show that our method overwhelmingly outperforms existing advanced SG optimization methods. Moreover, MPD-AGL consumes only 5.2% energy of ANN for a single inference at ultra-low latency $T = 2$.

2 Related Work

2.1 Direct Training of SNNs

With the introduction of spatio-temporal backpropagation and approximate derivatives of spike activity [Wu *et al.*, 2018; Wu *et al.*, 2019], direct training of SNNs has ushered in a new opportunity. [Fang *et al.*, 2021a; Hu *et al.*, 2025] modified the basic spiking ResNet to achieve identity mapping, effectively mitigating the degradation problem of directly training SNNs and achieving very deep SNNs. [Yao *et al.*, 2023; Lee *et al.*, 2025] incorporated the attention mechanism to estimate the saliency of different domains, helping SNNs focus on important features. [Fang *et al.*, 2021b; Yao *et al.*, 2022] developed neuronal variants to learn membrane parameters, expanding the expressiveness of SNNs. [Deng *et al.*, 2022; Guo *et al.*, 2022a] designed loss functions to regulate the distribution of spikes and membrane potentials along the temporal dimension to more accurately align the learning gradients.

2.2 Gradient Alignment

An essential component of SG learning is the suitable gradient flow [Zenke and Vogels, 2021]. To alleviate the problem of fixed SG not aligned with evolving MPD, [Guo *et al.*, 2022b] designed a membrane potential rectifier to redistribute potentials closer to the spikes. [Guo *et al.*, 2022c] introduced three regularization losses to penalize three undesired shifts of MPD. [Wang *et al.*, 2025a] quantified the inconsistency between actual distributions and targets, which was integrated into the overall network loss for joint optimization. Optimizing SG is another appealing approach. [Guo *et al.*, 2022a] approximated the gradient of spike activity by a differentiable asymptotic function evolving continuously, bridging the gap between pseudo and natural derivatives. [Che *et al.*, 2022] proposed a differentiable gradient search for parallel optimization of local SG. [Lian *et al.*, 2023] proposed a learnable SG to unlock the width limitation of SG. [Wang *et al.*, 2023] learned the accurate gradients of loss landscapes adaptively by fusing the learnable relaxation degree into a prototype network with random spike noise. [Wang *et al.*, 2025a] proposed a parametric SG strategy that can be iteratively updated. Considering the lack of synergy between these methods in matching SG and MPD, this motivates us to explore their correlations to maximize matching optimization.

3 Preliminary

3.1 Spiking Neural Model

Based on the essential electrophysiological properties of biological neurons, the leaky integrate-and-fire (LIF) model simulates the electrical activity of neurons in a simplified mathematical form, widely used in SNNs as the basis unit. For computational tractability, [Wu *et al.*, 2019] used the Euler formula to translate LIF into an iterative expression, the membrane potential evolves according to

$$I_i^n(t) = \sum_{j=1}^{l(n-1)} w_{ij}^n S_j^{n-1}(t), \tag{1}$$

$$V_i^n(t) = \tau V_i^n(t-1)(1 - S_i^n(t-1)) + I_i^n(t), \tag{2}$$

$$S_i^n(t) = \Theta(V_i^n(t)) = \begin{cases} 1, & V_i^n(t) \geq V_{th} \\ 0, & otherwise \end{cases} \tag{3}$$

where the superscript n, subscripts i and t denote the n-th layer, the i-th neuron and the t-th timestep, respectively. $l(n-1)$ denotes the number of neurons in the $(n-1)$-th layer. w_{ij}^n denotes the synapse weight from the j-th neuron in the $(n-1)$-th layer to the i-th neuron in the n-th layer. I, V, and S denote the pre-synaptic input, the membrane potential, and the binary spiking output of neurons, respectively. τ is the decay factor. V_{th} is the firing threshold.

3.2 Surrogate Gradient of SNNs

In Eq. 3, the activation function $\Theta(\cdot)$ of SNNs is a Heaviside step function. The derivative of output signals $\frac{\partial S}{\partial V}$ tends to infinity at the firing threshold V_{th} and zeros otherwise, i.e. Dirac function. SG learning allows gradient information to be backpropagated layer-wise along STD, which lays the foundation for developing general SNNs. In this work, we employ the rectangular SG [Wu *et al.*, 2018], which is defined as

$$\frac{\partial S_i^n(t)}{\partial V_i^n(t)} na \approx h(V_i^n(t)) = \frac{1}{\kappa} sign(|V_i^n(t) - V_{th}| < \frac{\kappa}{2}), \tag{4}$$

where hyperparameter κ controls the width of $h(\cdot)$ to ensure it integrates to 1, normally set to 1 [Wu *et al.*, 2018; Wu *et al.*, 2019]. The gradient $\frac{1}{\kappa}$ is available when the membrane potential $V_i^n(t)$ falls within the interval $[V_{th} - \frac{\kappa}{2}, V_{th} + \frac{\kappa}{2}]$.

3.3 Threshold-dependent Batch Normalization

There are some drawbacks to directly applying BN techniques in SNNs due to the inherent temporal dynamics of spiking neurons [Wu *et al.*, 2019]. To retain the advantages of BN in the channel dimension and capture the temporal dimension of SNN, [Zheng *et al.*, 2021] proposed threshold-dependent BN (tdBN), which normalized the pre-synaptic input I to the distribution of $N(0, (\alpha V_{th})^2)$ instead of $N(0,1)$. Let I_c^t represent the c-th channel feature maps of $I(t)$, then $I_c = (I_c^1, I_c^2, ..., I_c^T)$ will be normalized as

$$\hat{I}_c = \frac{\alpha V_{th}(I_c - \mathbb{E}[I_c])}{\sqrt{\mathbb{VAR}[I_c] + \epsilon}}, \quad \text{// normalize} \tag{5}$$

$$\bar{I}_c = \gamma_c \hat{I}_c + \beta_c, \quad \text{// scale and shift} \tag{6}$$

where $\mathbb{E}[I_c]$ and $\mathbb{VAR}[I_c]$ denote the expectation and variance of I_c, which are computed over the Mini-Batch. ϵ is a tiny constant. The hyperparameter α is to prevent overfire or underfire, normally set to 1 [Zheng *et al.*, 2021]. The pair of learnable parameters γ_c and β_c are initial to 1 and 0, for scaling and shifting the normalized $\hat{I}_c$.

4 Method

In this section, we introduce the MPD-AGL algorithm in detail and the overall training procedure.

4.1 Rethinking Pre-synaptic Input Distribution

To maintain the representation capacity of the layer, BN layers will normally perform a learnable affine transformation of the normalized pre-activations (Eq. 6). As shown in Fig. 2, the learnable parameters γ_c and β_c will evolve, and their discrepancy grows more pronounced during training. Thus, the pre-synaptic input normalized by tdBN [Zheng *et al.*, 2021] may not satisfy $I \nsim N(0, (V_{th})^2)$, which has not been considered in many previous studies. When SG uses the fixed gradient-available interval, unpredictable shifts in the membrane potential will naturally deviate from the optimal areas for gradient matching, resulting in performance limitations. As the membrane potential is directly computed from the pre-synaptic input, clarifying the distribution of pre-synaptic input helps to analyze the membrane potential shifts. For this respect, we propose **Theorem 1** to rethink the specific distribution of pre-synaptic input.

Theorem 1. *With the iterative LIF model and tdBN method, assuming normalized pre-synaptic input $I \sim N(0, (V_{th})^2)$, we have $\bar{I} \sim N(\bar{\beta}, (\bar{\gamma} V_{th})^2)$ after affine transformation, where $\bar{\beta} = \frac{1}{C}\sum_{c=1}^{C} \beta_c$ and $\bar{\gamma} = \frac{1}{C}\sum_{c=1}^{C} \gamma_c$, C is the channel size of tdBN layer.*

Proof. The proof of **Theorem 1** is presented in Supplementary A of the **Extended version**[1]. □

Theorem 1 proves that the distribution of pre-synaptic input normalized by tdBN is not only governed by threshold V_{th}, but also by the parameters γ_c and β_c of affine transformations. As the foundation of our work, it provides a new insight into the distribution of pre-synaptic input.

4.2 Adaptive Gradient Rule

To optimize SNN learning, we need to synchronously modify the gradient-available interval of SG to better align with the evolving MPD. Thus, it is essential to analyze the detailed dynamics of membrane potentials. [Zheng *et al.*, 2021] derived a high degree of similarity between the distribution of pre-synaptic input and membrane potential in neurons. [Lian *et al.*, 2023] extended this reasoning that for a given pre-synaptic input $I \sim N(0, (V_{th})^2)$, then the distribution of membrane potential is only determined by decay factor τ and satisfies $V \sim N(0, (1+\tau^2)(V_{th})^2)$. Based on the analysis in Section 4.1, it is known that the pre-synaptic input will deviate from the desired distribution after tdBN normalization. To this end, we further propose **Theorem 2** to express the relation between the pre-synaptic input, the decay factor, and the membrane potential at different timesteps.

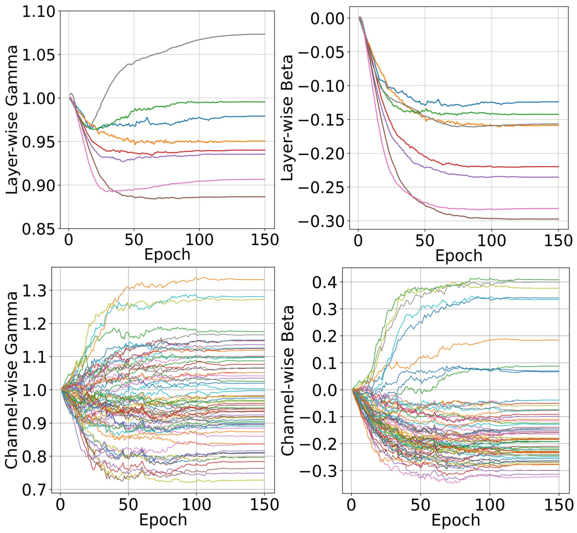

Figure 2: The affine transformation of tdBN in an 8-layer vanilla SNN. Top line is the variation curves of parameters γ and β for the average of all channels in each layer. Bottom line is the variation curves of parameters γ and β for all channels in the first layer.

Theorem 2. *Consider an SNN with T timesteps, the presynaptic input of neurons injected into the tdBN layer with affine transformation is normalized to satisfy $\bar{I} \sim N(\bar{\beta}, (\bar{\gamma}V_{th})^2)$, we have the membrane potential $\bar{V} \sim N(\bar{\beta}, (\bar{\gamma}V_{th})^2)$ when $t = 1$, and $\bar{V} \sim N((1+\tau)\bar{\beta}, (1+\tau^2)(\bar{\gamma}V_{th})^2)$ when $t > 1$, where $t \in T$.*

Proof. The proof of **Theorem 2** is also presented in Supplementary A of the **Extended version**[1]. □

Theorem 2 describes the dynamic distribution of membrane potential at different timesteps. [Lian *et al.*, 2023] observed a correlation between SG width κ and neuron decay factor τ and manually designed a proportional function (e.g. $\kappa = f(\tau)$) to describe it. As the decay factor also affects the membrane potential (Eq. 2), the proportional function can link SG width to MPD. From that view, we can avoid gradient information loss or redundancy by dynamically adjusting the SG width to control the alignment between the gradient-available interval and the evolving MPD. Then, we will concentrate on how to design the correlation function.

For a well-formed $f(\cdot)$, the key is to control the proportion of MPD in SG within a reasonable level. The variance reflects the dispersion of a distribution. An increase in variance indicates a more dispersed MPD, so the width needs to be enlarged to increase the proportion of neurons in SG to avoid gradient information loss. Conversely, a decrease in variance requires narrowing the width to reduce the proportion. Thus, a positive correlation arises between SG width and MPD. Here, we empirically set κ as 2 times the square root of $\mathbb{VAR}$ when $V_{th} = 0.5$ in our work, which ensures the initial width satisfies the standard rectangular SG setting (i.e. $k \approx 1$). Moreover, as PLIF neurons [Fang *et al.*, 2021b] can hierarchically learn the decay factor in SNNs, we also employ them to cooperate with the learnable affine transformation to control the evolving MPD together. It does not destroy the correlation function for scaling the SG width in response to MPD, but also enhances the expressiveness of SNNs. Finally, the correlation function $f(\cdot)$ can be formulated as

$$\kappa = f(\tau^n) = \begin{cases} 2 \times (\bar{\gamma}^n V_{th}), & t = 1 \\ 2 \times \sqrt{1 + (\tau^n)^2}(\bar{\gamma}^n V_{th}), & t > 1 \end{cases} \tag{7}$$

$$\tau^n = sigmoid(\rho^n) = \frac{1}{1 + e^{-\rho^n}}, \tag{8}$$

where learnable ρ^n is a layer-wise factor to ensure $\tau^n \in (0, 1)$. τ^n is initialized to 0.2 for all layers, which is adjusted when ρ^n is updated based on gradients (Eq. 11). In this way, SG can accurately capture the membrane potential shift and promptly update the gradient-available interval, effectively optimizing the loss landscape of SNNs.

4.3 The Overall Training Procedure

Employing the iterative LIF neurons in SNN has temporal dynamics in the spatial domain, which can well apply the spatio-temporal backpropagation algorithm (STBP) [Wu *et al.*, 2018] to update synapse weights. In the readout layer, we also only accumulate the membrane potential of output neurons without leakage and firing, as did in recent works [Rathi and Roy, 2023; Deng *et al.*, 2022], which can be described by

$$o_i^N = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{l(N-1)} w_{ij}^N S_j^{N-1}(t), \tag{9}$$

where N and T denote the number of layers and timesteps, respectively. Then, the gradient of synaptic weights w_{ij}^n and learnable ρ^n can be derived by the chain rule:

$$\begin{aligned} \frac{\partial L}{\partial w_{ij}^n} &= \sum_{t=1}^{T} \frac{\partial L}{\partial V_i^n(t)} \frac{\partial V_i^n(t)}{\partial I_i^n(t)} \frac{\partial I_i^n(t)}{\partial w_{ij}^n} \\ &= \sum_{t=1}^{T} \frac{\partial L}{\partial V_i^n(t)} \sum_{j=1}^{l(n-1)} S_j^{n-1}(t). \end{aligned} \tag{10}$$

$$\begin{aligned} \frac{\partial L}{\partial \rho^n} &= \sum_{t=1}^{T} \frac{\partial L}{\partial V_i^n(t)} \frac{\partial V_i^n(t)}{\partial \tau^n} \frac{\partial \tau^n}{\partial \rho^n} \\ &= \sum_{t=1}^{T} \frac{\partial L}{\partial V_i^n(t)} \frac{\partial V_i^n(t)}{\partial \tau^n} \tau^n(1 - \tau^n). \end{aligned} \tag{11}$$

As $V_i^n(t)$ not only contributes to the $S_i^n(t)$ but also governs the $V_i^n(t+1)$, it can be derived by

$$\frac{\partial L}{\partial V_i^n(t)} = \frac{\partial L}{\partial S_i^n(t)} \frac{\partial S_i^n(t)}{\partial V_i^n(t)} + \frac{\partial L}{\partial V_i^n(t+1)} \frac{\partial V_i^n(t+1)}{\partial V_i^n(t)}, \tag{12}$$

$$\begin{aligned} \frac{\partial L}{\partial S_i^n(t)} = &\sum_{j=1}^{l(n+1)} \frac{\partial L}{\partial V_j^{n+1}(t)} \frac{\partial V_j^{n+1}(t)}{\partial S_j^n(t)} \\ &+ \frac{\partial L}{\partial V_i^n(t+1)} \frac{\partial V_i^n(t+1)}{\partial S_i^n(t)}. \end{aligned} \tag{13}$$

Moreover, the pseudocode of the overall training procedure is briefed in **Algorithm 1**.

[1] Extended version: https://arxiv.org/abs/2505.11863

Algorithm 1 The overall training procedure of SNNs with MPD-AGL algorithm in one iteration

Input: Timestep: T; Threshold: V_{th}; Initial layer-wise decay: τ^n; input: $S(t), t \in T$; true-label vector: Y.
Output: updated the weight w_{ij}^n and learnable $\rho^n$of SNNs.

```
Forward:
for $n = 1$ to $N$ do
    if $n < N$ then
        Compute $I^n = w^n S^{n-1}$ // (1)
        $\bar{I}^n \leftarrow tdBN(I^n)$ // (4) and (5)
        for $t = 1$ to $T$ do
            Compute $\bar{V}^n(t), S^n(t)$ // (2) and (3)
            Compute the width of SG $\kappa$ // (7) and (8)
        end for
    else
        $o^N = \frac{1}{T}\sum_{t=1}^{T}(w^N S^{N-1}(t))$ // (9)
    end if
end for
$L \leftarrow CrossEntropy(o^N, Y)$
Backward:
for $n = 1$ to $N$ do
    for $t = 1$ to $T$ do
        $\frac{\partial L}{\partial V^n(t)} \leftarrow Grad(\frac{\partial L}{\partial S^n(t)}, \frac{\partial L}{\partial V^n(t+1)})$ // (12)
        $\frac{\partial L}{\partial S^n(t)} \leftarrow Grad(\frac{\partial L}{\partial V^{n+1}(t)}, \frac{\partial L}{\partial V^n(t+1)})$ // (13)
    end for
end for
Update the parameters $w_{ij}^n$ and $\rho^n$. // (10) and (11)
```

5 Experiment

In this section, we evaluate SNN with MPD-AGL for classification tasks on static CIFAR10/100, Tiny-ImageNet datasets, and the neuromorphic CIFAR10-DVS dataset.

5.1 Comparisons with Other Methods

As listed in Table 1, we compare the classification accuracy of the proposed method with other advanced methods. For **CIFAR10** dataset, MPD-AGL with ResNet-19 achieves 96.54% accuracy in 6 timesteps, significantly outperforming other methods. Notably, at ultra-low latency ($T = 2$), our method even slightly improves over all compared methods. For **CIFAR100** dataset, MPD-AGL still performs well and achieves the best accuracy of 80.49% in only 6 timesteps. Furthermore, our method outperforms LSG by an overwhelming margin of 2.52%, 2.87%, and 3.36%, respectively. The main reason is that LSG neglects the effect of affine transformation on the pre-synaptic input and membrane potential. As a result, the LSG-designed learnable SG cannot accurately capture the evolving MPD. For **CIFAR10-DVS** dataset, MPD-AGL with VGGSNN in 10 timesteps can reach an accuracy of 84.10% by using the TET loss [Deng *et al.*, 2022]. It even achieves the 82.50% accuracy w/o it, which is a greater improvement over other methods. For **Tiny-ImageNet** dataset, MPD-AGL with VGG-13 achieves the accuracy of 58.14% in 4 timesteps, outperforming ASGL by 1.57%.

5.2 Proportion of Gradient Available

To investigate whether the proposed method can effectively alleviate the gradient vanishing problem, we conducted experiments using ResNet-19 on the CIFAR10 dataset with 2 timesteps. MPD-AGL rethinks the distribution of pre-synaptic input in the tdBN method [Zheng *et al.*, 2021] and, inspired by LSG [Lian *et al.*, 2023], designs the correlation function to dynamically adjust SG. Therefore, we take STBP-tdBN and LSG as the benchmark algorithms. In Fig. 5(a) and Fig. 5(b), we compare the training loss and test accuracy of these three methods, where MPD-AGL can optimize the training loss to lower smooth values that have better generalization ability. Then we visualize the gradient-available proportion curve for layer 7 (Fig. 5(c)). The fixed width of SG in STBP-tdBN cannot effectively match the evolving MPD, which causes many neurons to fall outside the gradient-available interval and slow weight updates. Compared with STBP-tdBN, LSG can slightly alleviate this situation, but cannot respond to MPD timely. Specifically, LSG takes more epochs to make the proportion of neurons fall into the gradient-available interval to an appropriate level. Obviously, MPD-AGL can quickly capture the shifts in membrane potential and respond promptly. To reveal how our method helps SNNs for gradient propagation, we display the width of SG and the proportion of neurons that fall into the gradient-available interval in each layer. As illustrated in Fig. 5(d-e), the SG width in MPD-AGL is distributed mostly around 1.26, whereas LSG is 1.12. It means that MPD-AGL makes more neurons in deep layers have gradients, alleviating the gradient vanishing. Consequently, active neurons in all layers of MPD-AGL are higher than STBN-tdBN and LSG (Fig. 5(f)).

5.3 Effectiveness on SG Functions

To clarify the effectiveness of our method on other SG functions, we conducted experiments using ResNet-19 on the CIFAR100 dataset with 2 timesteps. Here, we choose three other widely used SG functions, i.e., triangular [Bellec *et al.*, 2018], sigmoid [Zenke and Vogels, 2021], and aTan [Fang *et al.*, 2021b]. Considering that optimal sharpness varies among different SGs, we scaled κ proportionally. As shown in Fig. 3, MPD-AGL achieves 76.43% accuracy with triangular SG, outperforming STBP-tdBN and LSG by 2.68% and 1.74%, respectively. While MPD-AGL still performs better

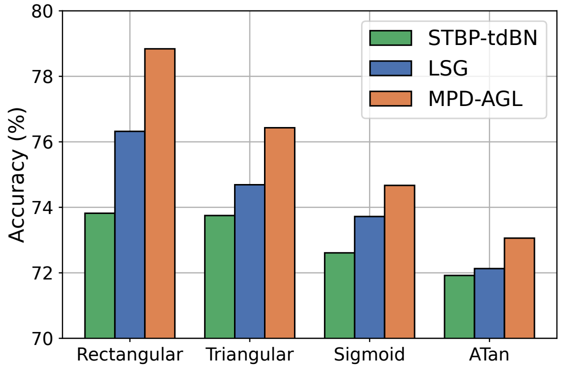

Figure 3: The effectiveness of other SG functions.

Dataset	Method	SG optimization	Architecture	Timestep	Accuary(%)
CIFAR10	TAB [Jiang *et al.*, 2024]	✗	ResNet-19	6 / 4 / 2	94.81 / 94.76 / 94.73
	ShortcutBP [Guo *et al.*, 2024]	✗	ResNet-19	2	95.19
	STAtten + [Lee *et al.*, 2025]	✗	SpikingReformer-6-384	4	95.26
	TCJA [Zhu *et al.*, 2025]	✗	MS-ResNet-18	4	95.60
	PSG [Wang *et al.*, 2025a]	✔	ResNet-19	6 / 4	95.00 / 95.12
	LSG [Lian *et al.*, 2023]	✔	ResNet-19	6 / 4 / 2	95.52 / 95.17 / 94.41
	DeepTAGE [Liu *et al.*, 2025]	✔	ResNet-18	4	95.86
	Ours	✔	**ResNet-19**	**6 / 4 / 2**	**96.54 / 96.35 / 96.18**
CIFAR100	TAB [Jiang *et al.*, 2024]	✗	ResNet-19	6 / 4 / 2	76.82 / 76.81 / 76.31
	IM-LIF [Lian *et al.*, 2024]	✗	ResNet-19	6 / 3	77.42 / 77.21
	TCJA [Zhu *et al.*, 2025]	✗	MS-ResNet-18	4	77.72
	STAtten + [Lee *et al.*, 2025]	✗	SpikingReformer-6-384	4	77.90
	PSG [Wang *et al.*, 2025a]	✔	ResNet-19	4	75.72
	LSG [Lian *et al.*, 2023]	✔	ResNet-19	6 / 4 / 2	77.13 / 76.85 / 76.32
	ASGL [Wang *et al.*, 2023]	✔	ResNet-18	4 / 2	77.74 / 76.59
	Ours	✔	**ResNet-19**	**6 / 4 / 2**	**80.49 / 79.72 / 78.84**
CIFAR10-DVS	IM-LIF [Lian *et al.*, 2024]	✗	VGGSNN	10	80.50
	IMPD-AGL [Jiang *et al.*, 2025]	✗	VGGSNN	10	77.20
	TET [Deng *et al.*, 2022]	✗	VGGSNN	10	77.33 / 83.17*
	STAtten + [Lee *et al.*, 2025]	✗	SpikingReformer-4-384	16	80.60
	PSG [Wang *et al.*, 2025a]	✔	ResNet-19	7	76.00
	LSG [Lian *et al.*, 2023]	✔	VGGSNN	10	77.90
	DeepTAGE[Liu *et al.*, 2025]	✔	VGG-11	10	81.23
	Ours	✔	**VGGSNN**	**10**	**82.50 / 84.10***
Tiny-ImageNet	Offline LTL [Yang *et al.*, 2022]	✗	VGG-13	16	55.37
	S3NN [Suetake *et al.*, 2023]	✗	ResNet-18	1	55.49
	IM-LIF [Lian *et al.*, 2024]	✗	ResNet-19	6 / 3	55.37 / 54.82
	AT [Ozdenizci and Legenstein, 2024]	✗	VGG-11	8	57.21
	ASGL [Wang *et al.*, 2023]	✔	VGG-13	8 / 4	56.81 / 56.57
	Ours	✔	**VGG-13**	**4**	**58.14**

Table 1: The comparison of classification performance on four benchmark datasets. * denotes using the Adam optimizer with $lr = 1e - 3$ and TET loss.

than STBP-tdBN and LSG on sigmoid and aTan SG functions, it does not perform as well as rectangular and triangular SG. This may be due to simply adjusting the sharpness of these asymptotic SG functions based on the evolving MPD, which leads to large oscillations in the gradient information. Instead, linear SG functions have a relatively smooth gradient estimation characteristic and thus exhibit stronger robustness.

5.4 Energy Efficiency

To validate the efficiency of SNNs in energy consumption, we conducted experiments using ResNet-19 on the CIFAR10 dataset. The theoretical energy consumption of SNNs can be estimated from the synaptic operations (SOPs) [Zhou *et al.*, 2023]. Due to the binarized and sparse nature of spikes, SNNs operate low-power AC operations only when neurons fire, and its required SOP varies with spike sparsity. In our model, real-valued images are directly fed into SNNs for encoding, and membrane potentials in the readout layer are used for prediction, so the SOPs contain AC operations and a few MAC operations. For the number of addition operations, we calculate it by $r^n \times T \times N^n_{AC}$, where r is the average firing rate of n-th layer and N^n_{AC} is the number of addition operations in n-th layer of an iso-architecture ANN. For the number of multiplication operations, it equals the number of multiplication operations at encoding and readout layers and scales by T [Yao *et al.*, 2023]. [Rathi and Roy, 2023] measured in 45 nm CMOS technology that an addition operation costs $0.9pJ$ and a MAC operation costs $4.6pJ$.

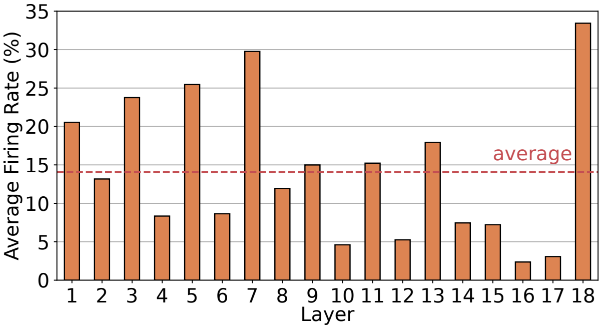

Figure 4: The average firing rate of each layer on CIFAR10 dataset.

Method	T	#Add.	#Multi.	Energy
ANN	-	2285.35M	2285.35M	10.51mJ
STBP-tdBN	2	890.20M	7.08M	0.83mJ
LSG	2	677.72M	7.08M	0.64mJ
MPD-AGL	2	579.33M	7.08M	0.55mJ
	4	1004.70M	14.16M	0.96mJ
	6	1303.21M	21.25M	1.25mJ

Table 2: The energy consumption on the CIFAR10 dataset.

As shown in Fig. 4, the average firing rate of each layer in spiking ResNet-19 does not exceed 34% (14% on average) when $T = 2$. In Table 2, we estimate the energy consumption during inference in different timesteps, and the proposed MPD-AGL is 19× lower compared to ANN in 2 timesteps.

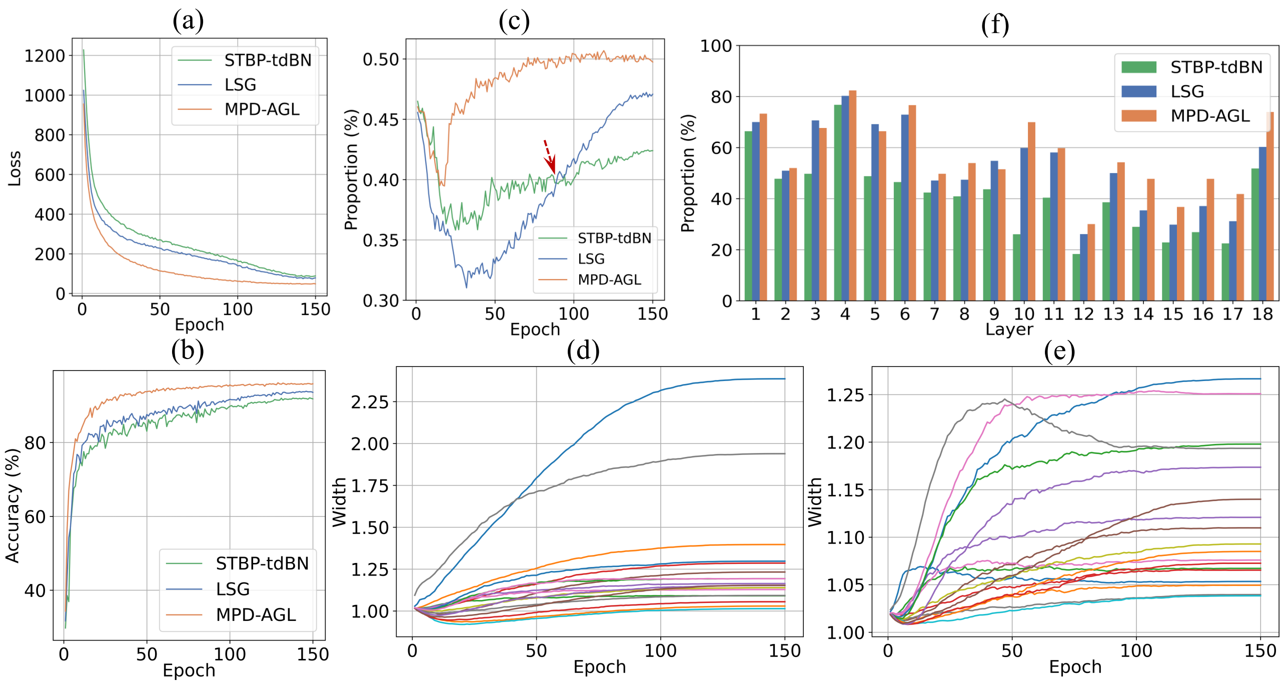

Figure 5: The comparison of different methods on the CIFAR10 dataset. (a) and (b) are the train loss and test accuracy, respectively. (c) and (f) are the proportion of neurons falling into the gradient-available interval in layer 7 and each layer of ResNet-19, respectively. (d) and (e) are the width of SG in each layer of MPD-AGL and LSG, respectively.

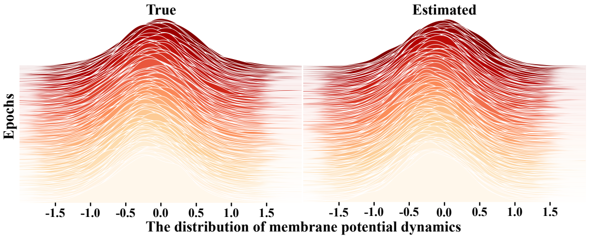

Figure 6: The proof of Theorem1 on CIFAR10-DVS dataset.

Method	Accuracy (%)	
	CIFAR10	CIFAR100
Vanilla	92.38	73.87
w/ trainable decay	93.15	74.12
w/ LSG [Lian *et al.*, 2023]	94.41	76.32
w/ AGR	95.93	78.47
w/ MPD-AGL	**96.18**	**78.84**

Table 3: The ablation study on CIFAR10/100 dataset. w/ AGR denotes using the proposed adaptive gradient rule.

5.5 Ablation Study

As shown in Fig. 6, the true mean and variance of presynaptic input are close to the estimated values reasoned from Theorem 1 during training, proving the correctness of Theorem 1. It also indicates that the affine transformation of normalization layers is the reason for the membrane potential shifts, limiting the performance of SG learning. As for Theorem 2, which follows [Zheng *et al.*, 2021; Lian *et al.*, 2023] by employing the factors of affine transformation, please refer to Supplementary A of the **Extended version** for detailed proofs. To evaluate the effectiveness of MPD-AGL algorithm, we also conducted experiments using ResNet-19 on the CIFAR10/100 datasets with 2 timesteps. In Table 3, applying the proposed adaptive gradient rule (AGR) achieves an accuracy of 95.93%/78.47% on the CIFAR10/100 datasets, surpassing the vanilla and LSG methods by 3.55%/4.60% and 1.52%/2.15%, respectively. When combined with AGR and PLIF neurons, it even reaches 96.18%/78.84%, which means that the trainable decay can indeed combine with the adaptive gradient rule to enhance SNN learning.

6 Conclusion

In this work, we present a new perspective on understanding the gradient vanishing or mismatch problems in directly training SNNs with SG learning. We identify that these issues primarily arise as the failure of fixed SG and evolving MPD to align, which is caused by the affine transformation in normalization layers. Here, we propose the MPD-AGL algorithm, which adaptively relaxes SG in a temporal-aligned manner to more accurately capture the evolving MPD at different timesteps. Experimental results and theoretical analysis on four datasets demonstrate the effectiveness and superiority of our approach. MPD-AGL unlocks the limitation of SG width and provides more flexible gradient estimation for SNNs. Importantly, it can naturally integrate into existing SNN architectures to further enhance performance without additional inference costs, hopefully promoting the application of SNNs in more complex tasks and wider scenarios.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (Grant No.: 2024YDLN0005) and the National Natural Science Foundation of China (Grant No.: 62276235)

References

[Bellec *et al.*, 2018] Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. *Advances in neural information processing systems*, 31, 2018.

[Che *et al.*, 2022] Kaiwei Che, Luziwei Leng, Kaixuan Zhang, Jianguo Zhang, Qinghu Meng, Jie Cheng, Qinghai Guo, and Jianxing Liao. Differentiable hierarchical and surrogate gradient search for spiking neural networks. *Advances in Neural Information Processing Systems*, 35:24975–24990, 2022.

[Dampfhoffer *et al.*, 2024] Manon Dampfhoffer, Thomas Mesquida, Alexandre Valentian, and Lorena Anghel. Backpropagation-based learning techniques for deep spiking neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):11906–11921, 2024.

[Deng *et al.*, 2022] Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting. In *International Conference on Learning Representations*, 2022.

[Fang *et al.*, 2021a] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34:21056–21069, 2021.

[Fang *et al.*, 2021b] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2661–2671, 2021.

[Guo *et al.*, 2022a] Yufei Guo, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Yinglei Wang, Xuhui Huang, and Zhe Ma. IM-Loss: Information maximization loss for spiking neural networks. *Advances in Neural Information Processing Systems*, 35:156–166, 2022.

[Guo *et al.*, 2022b] Yufei Guo, Yuanpei Chen, Liwen Zhang, YingLei Wang, Xiaode Liu, Xinyi Tong, Yuanyuan Ou, Xuhui Huang, and Zhe Ma. Reducing information loss for spiking neural networks. In *European Conference on Computer Vision*, pages 36–52. Springer, 2022.

[Guo *et al.*, 2022c] Yufei Guo, Xinyi Tong, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Zhe Ma, and Xuhui Huang. RecDis-SNN: Rectifying membrane potential distribution for directly training spiking neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 326–335, 2022.

[Guo *et al.*, 2024] Yufei Guo, Yuanpei Chen, Zecheng Hao, Weihang Peng, Zhou Jie, Yuhan Zhang, Xiaode Liu, and Zhe Ma. Take A Shortcut Back: Mitigating the gradient vanishing for training spiking neural networks. *Advances in Neural Information Processing Systems*, 37:24849–24867, 2024.

[Hu *et al.*, 2025] Yifan Hu, Lei Deng, Yujie Wu, Man Yao, and Guoqi Li. Advancing spiking neural networks toward deep residual learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2):2353–2367, 2025.

[Huh and Sejnowski, 2018] Dongsung Huh and Terrence J Sejnowski. Gradient descent for spiking neural networks. *Advances in neural information processing systems*, 31, 2018.

[Jiang *et al.*, 2024] Haiyan Jiang, Vincent Zoonekynd, Giulia De Masi, Bin Gu, and Huan Xiong. TAB: Temporal accumulated batch normalization in spiking neural networks. In *The Twelfth International Conference on Learning Representations*, 2024.

[Jiang *et al.*, 2025] Jiaqiang Jiang, Haohui Ding, Haixia Wang, and Rui Yan. Deep spiking neural networks driven by adaptive interval membrane potential for temporal credit assignment problem. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 9(1):717–728, 2025.

[Lee *et al.*, 2025] Donghyun Lee, Yuhang Li, Youngeun Kim, Shiting Xiao, and Priyadarshini Panda. Spiking transformer with spatial-temporal attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[Lian *et al.*, 2023] Shuang Lian, Jiangrong Shen, Qianhui Liu, Ziming Wang, Rui Yan, and Huajin Tang. Learnable surrogate gradient for direct training spiking neural networks. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 3002–3010, 2023.

[Lian *et al.*, 2024] Shuang Lian, Jiangrong Shen, Ziming Wang, and Huajin Tang. IM-LIF: Improved neuronal dynamics with attention mechanism for direct training deep spiking neural network. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 8(2):2075–2085, 2024.

[Liang *et al.*, 2024] Limei Liang, Runhao Jiang, Huajin Tang, and Rui Yan. An event-based feature representation method for event stream classification using deep spiking neural networks. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2024.

[Liu *et al.*, 2025] Wei Liu, Li Yang, Mingxuan Zhao, Shuxun Wang, Jin Gao, Wenjuan Li, Bing Li, and Weiming Hu. DeepTAGE: Deep temporal-aligned gradient enhancement for optimizing spiking neural networks. In *The Thirteenth International Conference on Learning Representations*, 2025.

[Ma *et al.*, 2024] De Ma, Xiaofei Jin, Shichun Sun, Yitao Li, Xundong Wu, Youneng Hu, Fangchao Yang, Huajin Tang, Xiaolei Zhu, Peng Lin, et al. Darwin3: a large-scale neuromorphic chip with a novel isa and on-chip learning. *National Science Review*, 11(5):nwae102, 2024.

[Ozdenizci and Legenstein, 2024] Ozan Ozdenizci and Robert Legenstein. Adversarially robust spiking neural networks through conversion. *Transactions on Machine Learning Research*, pages 2835–8856, 2024.

[Pei *et al.*, 2019] Jing Pei, Lei Deng, Sen Song, Mingguo Zhao, Youhui Zhang, Shuang Wu, Guanrui Wang, Zhe Zou, Zhenzhi Wu, Wei He, et al. Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature*, 572(7767):106–111, 2019.

[Qin *et al.*, 2023] Lang Qin, Rui Yan, and Huajin Tang. A low latency adaptive coding spike framework for deep reinforcement learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 3049–3057, 2023.

[Qin *et al.*, 2025] Lang Qin, Ziming Wang, Runhao Jiang, Rui Yan, and Huajin Tang. GRSN: Gated recurrent spiking neurons for pomdps and marl. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(2):1483–1491, 2025.

[Rathi and Roy, 2023] Nitin Rathi and Kaushik Roy. DIET-SNN: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 34(6):3174–3182, 2023.

[Suetake *et al.*, 2023] Kazuma Suetake, Shin-ichi Ikegawa, Ryuji Saiin, and Yoshihide Sawada. S3NN: Time step reduction of spiking surrogate gradients for training energy efficient single-step spiking neural networks. *Neural Networks*, 159:208–219, 2023.

[Wang *et al.*, 2023] Ziming Wang, Runhao Jiang, Shuang Lian, Rui Yan, and Huajin Tang. Adaptive smoothing gradient learning for spiking neural networks. In *International Conference on Machine Learning*, pages 35798–35816. PMLR, 2023.

[Wang *et al.*, 2025a] Siqi Wang, Tee Hiang Cheng, and Meng-Hiot Lim. Potential distribution adjustment and parametric surrogate gradient in spiking neural networks. *Neurocomputing*, 620:129189, 2025.

[Wang *et al.*, 2025b] Ziling Wang, Ziming Wang, Shuang Lian, Rui Yan, and Huajin Tang. Adaptive gradient-based timesurface for event-based detection. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

[Wang *et al.*, 2025c] Ziming Wang, Ziling Wang, Huaning Li, Lang Qin, Runhao Jiang, De Ma, and Huajin Tang. EAS-SNN: End-to-end adaptive sampling and representation for event-based detection with recurrent spiking neural networks. In *European Conference on Computer Vision*, pages 310–328. Springer, 2025.

[Wu *et al.*, 2018] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018.

[Wu *et al.*, 2019] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. Direct training for spiking neural networks: Faster, larger, better. *Proceedings of the AAAI conference on artificial intelligence*, 33(01):1311–1318, 2019.

[Yang *et al.*, 2022] Qu Yang, Jibin Wu, Malu Zhang, Yansong Chua, Xinchao Wang, and Haizhou Li. Training spiking neural networks with local tandem learning. *Advances in Neural Information Processing Systems*, 35:12662–12676, 2022.

[Yang *et al.*, 2024] Panpan Yang, Ziming Wang, Huajin Tang, and Rui Yan. Multi-scale harmonic mean time surfaces for event-based object classification. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2024.

[Yao *et al.*, 2022] Xingting Yao, Fanrong Li, Zitao Mo, and Jian Cheng. GLIF: A unified gated leaky integrate-and-fire neuron for spiking neural networks. *Advances in Neural Information Processing Systems*, 35:32160–32171, 2022.

[Yao *et al.*, 2023] Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, Bo Xu, and Guoqi Li. Attention spiking neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 45(8):9393–9410, 2023.

[Zenke and Vogels, 2021] Friedemann Zenke and Tim P Vogels. The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. *Neural computation*, 33(4):899–925, 2021.

[Zhang and Li, 2020] Wenrui Zhang and Peng Li. Temporal spike sequence learning via backpropagation for deep spiking neural networks. *Advances in neural information processing systems*, 33:12022–12033, 2020.

[Zheng *et al.*, 2021] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. *Proceedings of the AAAI conference on artificial intelligence*, 35(12):11062–11070, 2021.

[Zhou *et al.*, 2023] Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *The Eleventh International Conference on Learning Representations*, 2023.

[Zhu *et al.*, 2025] Rui-Jie Zhu, Malu Zhang, Qihang Zhao, Haoyu Deng, Yule Duan, and Liang-Jian Deng. TCJA-SNN: Temporal-channel joint attention for spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 36(3):5112–5125, 2025.