Neural Networks 170 (2024) 190–201

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Full Length Article

Exploiting nonlinear dendritic adaptive computation in training deep Spiking Neural Networks

Guobin Shen [a,c,1], Dongcheng Zhao [a,1], Yi Zeng [a,b,c,*]

[a] Brain-Inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China
[b] Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences (CAS), Shanghai, China
[c] School of Future Technology, University of Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Keywords:
Dendritic Nonlinearity
Dendritic Spatial Gating Module
Dendritic Temporal Adjust Module
Spiking Neural Networks

ABSTRACT

Inspired by the information transmission process in the brain, Spiking Neural Networks (SNNs) have gained considerable attention due to their event-driven nature. However, as the network structure grows complex, managing the spiking behavior within the network becomes challenging. Networks with excessively dense or sparse spikes fail to transmit sufficient information, inhibiting SNNs from exhibiting superior performance. Current SNNs linearly sum presynaptic information in postsynaptic neurons, overlooking the adaptive adjustment effect of dendrites on information processing. In this study, we introduce the Dendritic Spatial Gating Module (DSGM), which scales and translates the input, reducing the loss incurred when transforming the continuous membrane potential into discrete spikes. Simultaneously, by implementing the Dendritic Temporal Adjust Module (DTAM), dendrites assign different importance to inputs of different time steps, facilitating the establishment of the temporal dependency of spiking neurons and effectively integrating multi-step time information. The fusion of these two modules results in a more balanced spike representation within the network, significantly enhancing the neural network's performance. This approach has achieved state-of-the-art performance on static image datasets, including CIFAR10 and CIFAR100, as well as event datasets like DVS-CIFAR10, DVS-Gesture, and N-Caltech101. It also demonstrates competitive performance compared to the current state-of-the-art on the ImageNet dataset.

1. Introduction

As the third generation of Artificial Neural Networks (ANNs) (Maass, 1997), Spiking Neural Networks (SNNs) consist of biologically inspired spiking neurons and synaptic connections. Compared with the real-valued artificial neural network, the SNNs use binary spike sequences to transmit information, which is more in line with the information processing mechanism of the brain. Furthermore, it is more convenient to combine with neuromorphic sensors to perform efficient computing on neuromorphic hardware (Furber et al., 2012). Also, the rich spatio-temporal dynamic characteristics of spiking neurons make SNNs show great potential in brain-inspired artificial intelligence (Shen, Zhao, & Zeng, 2022a; Wu, Deng, Li, Zhu, & Shi, 2018; Zhao, Zeng, & Li, 2022) and brain simulation (Zeng, Zhao, Zhao, Shen, Dong, Lu, Zhang, Sun, Liang, Zhao, et al., 2023).

The non-differentiable characteristics of SNNs make the training of SNNs difficult, which attracts many researchers to study. Some brain-inspired synaptic plasticity learning algorithms such as Spike-Timing-Dependent Plasticity (STDP) (Bi & Poo, 1998) and Short Term Plasticity (STP) (Zucker & Regehr, 2002) have solved the training problem of SNNs to a certain extent. However, the performance of complex tasks is still poor. The surrogate gradient enables the SNNs to be trained using the backpropagation algorithm, which significantly promotes the development of SNNs. However, as the network deepens, the SNNs gradually become challenging to converge. Although there are some algorithms such as neuron normalization (Wu et al., 2019; Zheng, Wu, Deng, Hu, & Li, 2021), deep residual learning (Fang, Yu, Chen, Huang, et al., 2021), and network architecture search (Kim, Li, Park, Venkatesha, & Panda, 2022) have improved the training of SNNs from various aspects, most of these are borrowing experience from the optimization of ANNs, while ignoring the characteristics of the SNNs.

Although a single spiking neuron can achieve the performance of deep artificial neural networks (Beniaguev, Segev, & London, 2021), due to hardware limitations, the Leaky Integrate-and-Fire (LIF) and IF spiking neurons are often used when constructing large-scale deep SNNs. These neurons only passively sum the presynaptic neuron information linearly, accumulate the membrane potential, and fire spikes

* Corresponding author at: Brain-Inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China.
E-mail address: yi.zeng@ia.ac.cn (Y. Zeng).
[1] Guobin Shen, Dongcheng Zhao contributed equally to this work.

https://doi.org/10.1016/j.neunet.2023.10.056
Received 21 November 2022; Received in revised form 22 August 2023; Accepted 31 October 2023
Available online 10 November 2023

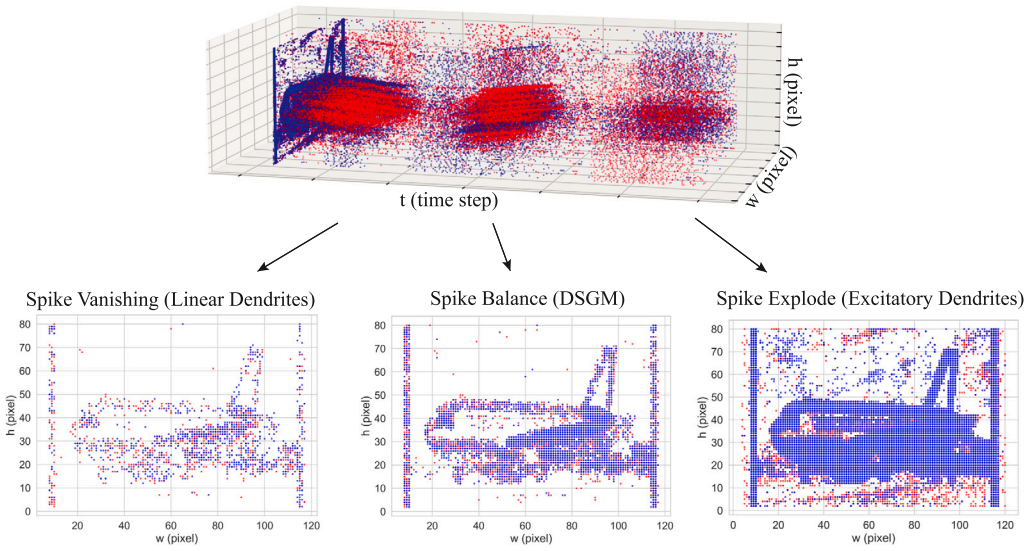

Fig. 1. Illustration of the relationship between different neuronal models and informativeness in the features.

when the membrane potentials reach the threshold. They often share the same hyperparameters and cannot dynamically adjust themselves to the input current to achieve the optimal fire rate. Fig. 1 shows the feature maps corresponding to the DVS data processed by different types of neurons. Different excitation patterns of neurons affect the informativeness of the features and greatly influence the final accuracy. We find that the plain linear synaptic model affects the excitatory pattern of neurons. When the number of layers deepens, the deep spiking neurons are under-activated for a long time and are in the spike vanishing state. In this state, the network cannot transmit sufficient information in the forward propagation, and the weights cannot be effectively updated in the backpropagation process. By simply lowering the threshold or strengthening the input of the spiking neurons, although the deep spiking neurons will be activated, it will cause the shallow spiking neurons to release too many spikes and be in a state of spike explosion. A large number of spikes will make it difficult for the networks to extract useful information, thus destroying the performance of SNNs. A spike-balanced state will enable the SNNs to learn richer representations and converge rapidly and stably.

In addition to the computational properties of the soma, dendrites, as powerful computational units, play a significant role in the information processing in the brain Acharya et al. (2021), Bicknell and Häusser (2021). Inspired by dendrites' powerful spatiotemporal information processing abilities, we construct the adaptive dendritic computing module from spatial and temporal dimensions to adjust the input so that the neuron firing states are more balanced, thus enriching the feature extraction ability. To illustrate the superiority of our model, we conduct experiments on several datasets, our model dramatically improves the performance of SNNs, and our contributions can be summarized as follows:

- Inspired by the dendrites' spatial information processing mechanism, we design the dendritic spatial gating module (DSGM) to translate and scale the input at the current moment adaptively. DSGM achieves multimode membrane potential distribution without external constraints and reduces the gap between membrane potential and spike distribution.
- Inspired by the temporal information processing mechanism of dendrites, we design the dendritic temporal adjustment module (DTAM) to adaptively adjust the contribution of the input current at each moment to the membrane potential.
- We evaluate our method on the static image datasets such as CIFAR10 and CIFAR100 and the event datasets such as DVS-CIFAR10, N-Caltech101, and DVS-Gesture. Experiments show that the proposed DSGM and DTAM can help balance the spiking firing rates and achieve state-of-the-art accuracy with very low latency. For the ImageNet dataset, our algorithm shows comparable performance with the current best algorithms under lighter structures and lower latency. Codes and data have been deposited in GitHub https://github.com/BrainCog-X/Brain-Cog.

2. Related works

The design of high-performance deep SNNs is mainly divided into two aspects, conversion-based SNNs and the directly trained SNNs based on surrogate gradient.

2.1. Conversion based

The conversion-based SNNs usually apply the parameter constraints on the well-trained ANNs, so the ReLU activation function in ANNs can be directly replaced with IF neurons. In order to improve the performance after conversion, many methods have been proposed, such as soft reset (Han, Srinivasan, & Roy, 2020), synchronous neurons (Li, Zhao, & Zeng, 2022), burst spikes (Li & Zeng, 2022). In order to reduce the conversion error, Bu et al. (2021) replaces the ReLU activation function of the ANN before training with the clip-floor-shift function. Although the conversion-based SNNs can achieve competitive results on large-scale datasets, most of these methods are limited to constructing SNNs using simple IF neurons and ignore the temporally rich dynamics of SNNs. These lead to SNNs obtained by conversion methods that do not handle event data well and have high latency.

2.2. Directly trained

To exploit the dynamic nature of SNNs in the temporal domain and to train SNNs end-to-end, some researchers have attempted to train SNNs directly from scratch. The main problem faced in training SNNs from scratch via gradient descent is the non-differentiability of the binary spikes. Wu et al. (2018, 2019) proposed the STBP algorithm to speed up the training process of SNNs based on the spatio-temporal dynamic properties of SNNs. Then the learnable neuron hyperparameters, such as the membrane time constant (Fang, Yu, Chen, Masquelier, et al., 2021), firing threshold (Shaban, Bezugam, & Suri, 2021). Then the attention mechanism (Liu, Xing, Feng, Tang, & Pan, 2022; Yao et al., 2021) is introduced to enhance the spatiotemporal information processing capability of SNNs. By adding an additional loss, Guo

et al. (Guo et al., 2022) achieved the conversion of membrane potential from single-peak to multi-peak distribution, reducing the gap between the distribution of membrane potential and spike and improving the performance of SNNs. Duan et al. (Duan, Ding, Chen, Yu, & Huang, 2022) proposed Temporal Effective Batch Normalization (TEBN) to make the temporal distribution of SNNs smoother and more even by assigning different weights to each moment. Yao et al. Yao, Li, Mo, and Cheng (2022) introduced gating factors to fuse different biological features and expand the characterization space of SNNs. However, in these methods, spiking neurons often passively receive presynaptic information while ignoring the regulatory role of dendrites as a powerful computing unit in information transmission.

3. Methods

In this section, first, we will give an introduction to the spiking neuron model. Then the dendritic spatial gating module and the dendritic temporal adjustment module will be discussed in detail.

3.1. Spiking neuron model

The most commonly used LIF spiking neuron model is adopted in our network. The LIF model is transformed into an efficient iterative expression for better simulation by solving the first-order differential equation with Euler's method.

$$\begin{aligned} u_{t+1,i} &= (\frac{1}{\tau}(\sum_{j=1}^{N} w_{ij}s_{t,j} - u_{t,i}) + u_t)(1 - s_{t,i}) + v_{rst}s_{t,i} \\ s_{t+1,i} &= H(u_{t+1,i} - v_{th}) \end{aligned} \tag{1}$$

In Eq. (1), $s_{t,j}$ represents the spike of the presynaptic neuron, w_{ij} is the synaptic weight between neuron j and neuron i, and $x_{t,i} = \sum_{j=1}^{N} w_{ij}s_{t,j}$ represents the input current calculated from presynaptic spikes. u_t denotes the membrane potential of the neuron at t, $\tau \in (1, +\infty)$ denotes the membrane time constant, and $s_t \in \{0, 1\}$ denotes the spike of the neuron at t. After a neuron delivers a spike, the membrane potential is reset to v_{rst}. $H(\cdot)$ denotes the Heaviside step function, which is used to realize the spike process of the neuron. v_{th} denotes the firing threshold.

As shown in Eq. (1), the process of neurons firing spikes is non-differentiable. Applying the surrogate gradient (Bohte, 2011) allows the backpropagation (Rumelhart, Hinton, & Williams, 1986) to be successfully applied to the optimization process of the network. The surrogate gradient we use is the same as Wu et al. (2018) and can be expressed as:

$$\frac{\partial s_t}{\partial u_t} = (\alpha - \alpha^2 |u_t|) sign(\frac{1}{\alpha} - |u_t|) \tag{2}$$

In Eq. (2), α controls the width of the surrogate gradient function, which is set to 2 in our experiments.

Neurons predominantly receive their inputs through dendrites. In the LIF model, dendrites play a linear summation function, ignoring the critical computational role played by dendrites. In real biological neurons, dendrites respond nonlinearly to the input stimuli of different intensities and can perform many functions such as temporal filtering (RoSE & Call, 1992), amplification (Szalai, Kevwitch, & McGrath, 2003). Here we take inspiration from the adaptive dendritic computation and propose the Dendritic Spatial Gating Module (DSGM) and Dendritic Temporal Adjust Module (DTAM) to translate and scale the input and adjust different importance to the input at different time steps.

3.2. Dendritic spatial gating module

The spiking neurons in SNNs control whether the neurons are excited and output spikes by comparing the membrane potential with the threshold. As shown in Eq. (1), converting continuous membrane potentials into discrete spikes makes SNNs more biologically plausible and energy efficient. However, this conversion process is inherently nonlinear and can lead to information losses, which negatively affect the performance of SNNs. When the input current is negative, the membrane potential of the neuron also becomes negative, keeping the neuron in an inhibited state and preventing it from firing. Therefore, the transformation error between the membrane potential and the spike of a neuron can be measured by Eq. (3):

$$\mathcal{E} = \sum_t [H(u_t - v_{th}) - u_t]^2 \cdot (u_t \geq 0) \tag{3}$$

Instead of directly imposing external constraints on the model, we hope to empower neurons with the ability to adapt to complex distributions by more biologically plausible means to reduce transformation errors.

Inspired by the nonlinear dendrites in biological neurons, we propose the Dendritic Spatial Gating Module (DSGM), which can be described as:

$$u_{t+1} = (\frac{1}{\tau}(f(x_t) - u_t) + u_t)(1 - s_t) + v_{rst}s_t \tag{4}$$

As shown in Eq. (4), in contrast to the basic LIF model in Eq. (1), we implement nonlinear modeling of dendrites by mapping the input currents nonlinearly with $f(\cdot)$. To achieve a continuous, smooth nonlinear response for different input currents, $f(\cdot)$ can be described as:

$$f(x) = \begin{cases} x - \gamma + \xi, & x \geqslant \gamma \\ \beta(e^{x-\gamma} - 1) + (1 - \beta)(x - \gamma) + \xi, & x < \gamma \end{cases} \tag{5}$$

As shown in Eq. (5), the γ and ξ control the translations in the input and output dimensions, respectively, and β controls the decay of inhibitory inputs. The learnable parameters in the DSGM enable nonlinear modeling of synapses and enable SNNs to achieve spike balance in response to external stimuli (see Figs. 2 and 3).

The DSGM facilitates a nonlinear reaction to presynaptic input and tailors the input current, which becomes critical for SNNs that only provide binary spike outputs. As described in Eq. (5), parameter γ adjusts the distribution of input currents, while parameter β modifies the inhibitory stimulus. It achieves this by scaling down the inhibitory input, which mitigates the intensity of such input, thus addressing the prevalent issue of spike disappearance that often plagues deep SNNs.

On the other hand, ξ serves to transform the modulated input currents, resulting in a shift in the output distribution. These parameters together effectuate a nonlinear mapping of neuronal input currents. This mapping permits neurons to convert the traditionally single-peaked membrane potential distribution into a more representative multi-peaked distribution, aligning better with the features of a discrete spike.

This proposed synaptic model, boasting a more comprehensive characterization capacity, is designed to minimize transformation errors typical in spiking neurons. In essence, it introduces a more dynamic, adaptable response to the synaptic input, enhancing the overall efficiency and accuracy of SNNs.

3.3. Dendritic temporal adjustment module

Dendrites possess the unique capability of processing information spatially as well as temporally, exhibiting varied responses to inputs at different time steps. The Leaky Integrate-and-Fire (LIF) neuron model, however, employs a fixed membrane time constant, preventing any dynamic changes during inference. This invariability in the membrane time constant leads to an exponential decay of input current over time,

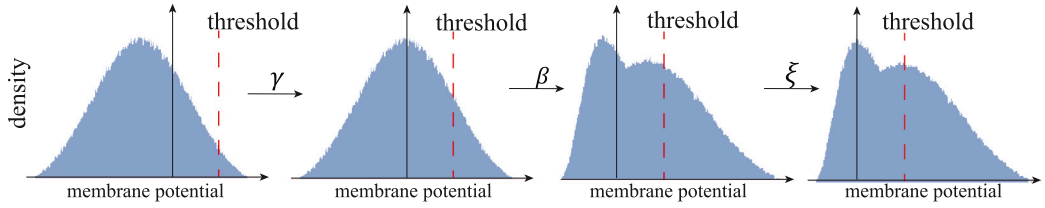

Fig. 2. Illustration of the distribution of postsynaptic membrane potential. DSGM achieves reshaping of membrane potential by nonlinear dendritic modeling.

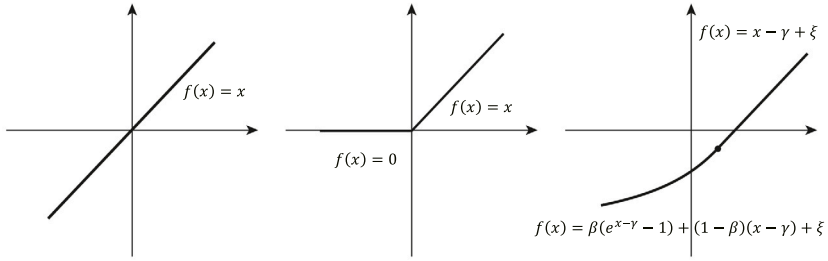

Fig. 3. The linear dendrities, excitatory dendrities, and our DSGM.

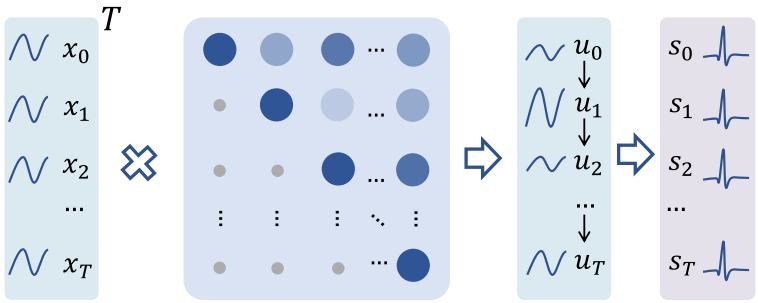

Fig. 4. Schematic of our Dendritic Temporal Adjustment Module (DTAM). This module enables attributing different significance to inputs at various time steps.

with the decay rate remaining the same across all time steps. Consequently, neurons are hindered from attributing significance to pivotal moments in time. This decay also inhibits neurons from establishing long-term dependencies and effectively integrating information from various moments.

To address these issues, we introduce the Dendritic Temporal Adjustment Module (DTAM) as depicted in Fig. 4. DTAM is designed to establish a correlation between dendrites and input currents at different moments. Upon employing DTAM, the dendritic state can be modulated by the input current at diverse time steps, thereby allowing the attribution of varying levels of importance to inputs at these different steps.

$$u_{t+1} = (\frac{1}{\tau}(g_{\boldsymbol{w}}(f(\boldsymbol{x}_t)) - u_t) + u_t)(1 - s_t) + v_{rst}s_t$$
$$\text{s.t.} \quad \boldsymbol{x}_t = [x_0, x_1, \ldots, x_t, 0, \ldots, 0, 0]^T \in \mathbb{R}^T \tag{6}$$

In Eq. (6), for an individual neuron, $\boldsymbol{x}_t$ encapsulates the input currents from time step 0 to t, while the remaining elements from $t+1$ to T are zero-padded. T denotes the total number of simulation steps. Additionally, $g_{\boldsymbol{w}}(\cdot)$ represents a function of the input current, which is weighted by an adjustable parameter. The role of $f(\boldsymbol{x}_t)$ in this equation is to summarize the input currents over time.

In our DTAM, the introduction of $g_{\boldsymbol{w}}$ allows us to adjust the weights of input currents from different moments, allowing neurons to assign different levels of importance to inputs at various time steps, thus enabling the establishment of long-term dependencies and more effective integration of information from different moments. The DTAM offers a novel approach to managing the temporal dynamics of input currents, enhancing the performance of SNNs by providing a more flexible and adaptive temporal processing mechanism.

4. Experiments

To verify the effectiveness and efficiency of the proposed method, we apply DSGM and DTAM to deep SNNs and validate them on classification tasks with static image and event datasets. We train deep SNNs using STBP (Wu et al., 2018) and use AdamW as the optimizer, with the initial learning rate set to 5×10^{-3}, and cosine decay to 0. The batch size is set to 128, and the training epoch is set to 600. For static datasets, the images are converted to spike sequences using direct encoding (Rathi & Roy, 2021; Wu et al., 2019), while for all datasets, the membrane potential of the last layer of neurons is used for direct output (Wu et al., 2019; Zheng et al., 2021). We use the LIF neuron as a baseline and apply DSGM and DTAM to the LIF neuron to test their effects on the performance of SNNs. The threshold voltage is set to 0.5, and the initial membrane time constant is set to 2. The surrogate gradient is applied to backpropagation to overcome the problem of the non-differentiability of binary spikes. As in Wu et al. (2018) we use triangular pulses to approximate the gradient of neurons. We assigned independent learnable parameters to the DSGMs in different channels. The γ and ξ in the DSGM are initialized to 0, and β is initialized to 0.75. For DTAM, the lower triangular matrix is used to describe the effects of synaptic currents at different moments and to ensure that previous moments were not influenced by subsequent moments. Moreover, each layer of neurons shares the same DTAM weights to prevent DTAM from significantly affecting the number of network parameters.

Table 1
Compare with existing works on static image datasets.

Dataset	Model	Methods	Architecture	Step	Accuracy
CIFAR10	Rathi, Srinivasan, Panda, and Roy (2019)	Hybrid training	ResNet-20	250	92.22
	Rathi and Roy (2021)	Diet-SNN	ResNet-20	10	92.54
	Wu et al. (2019)	STBP NeuNorm	CIFARNet	12	90.53
	Zhang and Li (2020)	TSSL-BP	CIFARNet	5	91.41
	Shen et al. (2022a)	STBP	7-layer-CNN	8	92.15
	Na et al. (2022)	STBP	NAS	16	93.15
	Zheng et al. (2021)	STBP-tdBN	ResNet-19	6	93.16
	Deng, Li, Zhang, and Gu (2021)	TET	ResNet-19	6	94.50
	Duan et al. (2022)	TEBN	ResNet-19	6	94.71
	Yao et al. (2022)	GLIF	ResNet-19	6	95.03
	our method	DSGM+DTAM	ResNet-19	4	**96.41** ± 0.03
CIFAR100	Rathi et al. (2019)	Hybrid training	VGG-11	125	67.87
	Rathi and Roy (2021)	Diet-SNN	ResNet-20	5	64.07
	Shen et al. (2022a)	STBP	ResNet34	8	69.32
	Kim et al. (2022)	STBP	NAS	5	73.04
	Deng et al. (2021)	TET	ResNet-19	6	74.72
	Duan et al. (2022)	TEBN	ResNet-19	6	76.41
	Yao et al. (2022)	GLIF	ResNet-19	6	77.35
	our method	DSGM+DTAM	ResNet-19	4	**78.87** ± 0.18
ImageNet	Rathi et al. (2019)	Hybrid training	ResNet-34	250	61.48
	Sengupta, Ye, Wang, Liu, and Roy (2019)	SPIKE-NORM	ResNet-34	2500	69.96
	Zheng et al. (2021)	STBP-tdBN	Spiking-ResNet-34	6	63.72
	Fang, Yu, Chen, Masquelier, et al. (2021)	SEW ResNet	SEW-ResNet-34	4	67.04
	Deng et al. (2021)	TET	SEW-ResNet-34	4	68.00
	Fang, Yu, Chen, Masquelier, et al. (2021)	SEW ResNet	SEW-ResNet-152	4	69.26
	Yao et al. (2022)	GLIF	ResNet-34	6	69.09
	Yao et al. (2022)	TEBN	ResNet-34	4	64.29
	our method	DSGM+DTAM	Spiking-ResNet-34	4	68.47
		DSGM+DTAM	SEW-ResNet-34	4	**69.36**

4.1. Static datasets

To verify the performance of our DSGM and DTAM, we first conduct experiments on the commonly used static classification datasets, CIFAR10, CIFAR100, and ImageNet. All static datasets are based on the direct input encoding used in Wu et al. (2019).

4.1.1. CIFAR

The CIFAR series mainly include two categories: CIFAR10 (Krizhevsky et al., 2009) and CIFAR100 dataset (Xu, Wang, Chen, & Li, 2015). They are widely used to verify multiple visual recognition algorithms. The CIFAR10 dataset contains ten categories of image samples, with a total of 50,000 training samples and 10,000 test samples. CIFAR100 is based on CIFAR10 to classify images more finely, with 100 categories, making it more challenging. The samples in the dataset are color datasets of size 32×32. $T = 4$ are used for training and validation on the CIFAR dataset.

4.1.2. ImageNet

ImageNet (Krizhevsky, Sutskever, & Hinton, 2012) is a large computer vision dataset with a total of 1000 categories, containing more than 1250,000 training samples and 50,000 validation samples. The samples are cropped to size of 224×224, and apply standard data augmentation for training. Similar to CIFAR10, the simulation step is $T = 4$.

The results are compared with several existing works on the structures ResNet19 (Zheng et al., 2021), Spiking-ResNet34 (Zheng et al., 2021), SNN5 (Deng et al., 2021) and VGGSNN (Deng et al., 2021). As shown in Table 1, for CIFAR10 and CIFAR100 datasets, our model achieved performance with 96.41% and 78.87%, which achieves state-of-the-art performance compared with other superior work (Deng et al., 2021; Zheng et al., 2021) on the same network architecture. For the more complex dataset CIFAR100, our model improves by nearly 4.4% over the current state-of-the-art algorithm with the same simulation length and network architecture. On ImageNet, we also achieve competitive results using a simpler network structure than other methods. We achieve a 69.36% accuracy compared to methods using the same structure on ImageNet.

4.2. Event datasets

We further conduct experiments on the event datasets, DVS-CIFAR10 (Li, Liu, Ji, Li, & Shi, 2017), DVS-Gesture (Amir et al., 2017), and N-Caltech101 (Orchard, Jayawant, Cohen, & Thakor, 2015) to demonstrate the powerful information representation ability.

4.2.1. DVS-CIFAR10

DVS-CIFAR10 is the event stream version of CIFAR10. There are ten categories, including 10,000 samples in CIFAR10. Since the original DVS-CIFAR10 does not divide the training set and test set, we use the first 90% of each category as the training set and the remaining 10% as the test set. The samples are resized to 48×48, and random horizontal flips, rotations, and crops are applied for data argumentation. We set the simulation set $T = 10$.

4.2.2. DVS-gesture

DVS-Gesture is a real-world gesture recognition dataset collected by the dynamic vision sensor (DVS). It contains 11 categories of gestures, comprising 1342 different samples. The samples are resized to 48×48 with no data augmentation applied. The training set and validation set are divided by 8:2.

4.2.3. N-Caltech101

The N-Caltech101 is a neuromorphic version of the Caltech101, obtained by capturing images of the Caltech101 displayed on an LCD monitor with a freely moving event camera. We resize samples to 48×48, and use the same data argumentation strategy as DVS-CIFAR10. The training and validation sets are split 9:1. The simulation step is set to $T = 10$, and $T = 16$ to test the performance of SNNs with different latencies.

As shown in Table 2, our model shows competitive results on the event dataset. On DVS-CIFAR10, we achieve an accuracy improvement of 1.05% compared to other methods. After applying EventMix (Shen, Zhao, & Zeng, 2022b), we achieved an accuracy improvement of 2.35% compared with SNNs of the same structure. Using only 80% of the training data, we achieved an accuracy that exceeded those of other methods

Table 2
Compare with existing works on event datasets.

Dataset	Model	Methods	Architecture	Step	Accuracy
DVS-CIFAR10	Zheng et al. (2021)	STBP-tdBN	ResNet-19	10	67.8
	Kugele, Pfeil, Pfeiffer, and Chicca (2020)	Streaming Rollout	DenseNet	10	66.8
	Wu et al. (2021)	Conv3D	LIAF-Net	10	71.70
	Wu et al. (2021)	LIAF	LIAF-Net	10	70.40
	Na et al. (2022)	STBP	NAS	16	72.50
	Shen et al. (2022a)	STBP	5-layer-CNN	16	78.95
	Duan et al. (2022)	TEBN	7-layer CNN	10	75.10
	Deng et al. (2021)	TET	VGGSNN	10	83.17
	Yao et al. (2022)	GLIF	7B-wideNet	16	78.10
	our method	DSGM+DTAM	VGGSNN	10	84.23 ± 0.48
		DSGM+DTAM	VGGSNN	10	**85.53**[b] ± 0.39
DVS-Gesture	Xing, Di Caterina, and Soraghan (2020)	SLAYER	5-Layer-CNN	20	92.01
	Shrestha and Orchard (2018)	SLAYER	16-layer-CNN	300	93.64
	Fang, Yu, Chen, Masquelier, et al. (2021)	STBP	BP	20	97.57
	Na et al. (2022)	STBP	NAS	16	96.53
	our method	DSGM + DTAM	SNN5	10	**98.36** ± 0.53
		DSGM + DTAM	SNN5	10	**96.69**[a] ± 0.40
		DSGM + DTAM	SNN5	16	**98.58**[a] ± 0.49
N-Caltech101	Kugele et al. (2020)	STBP	VGG11	20	55.0
	Ramesh et al. (2019)	N/A	N/A	N/A	66.8
	our method	DSGM + DTAM	VGGSNN	10	74.48 ± 0.31
		DSGM + DTAM	VGGSNN	10	76.39[b] ± 0.39
		DSGM + DTAM	VGGSNN	16	84.42 ± 0.43
		DSGM + DTAM	VGGSNN	16	**85.26**[b] ± 0.14

[a] Denotes that the training and test sets are divided by 8:2, while other works uses 9:1.
[b] Indicates using EventMix.

using 90% data for training. On the more complex N-Caltech101, we achieved an accuracy of 85.26%.

4.3. Result analysis

In this section, we present an advanced analysis of the proposed method in terms of the membrane potential distribution, spike sparsity, and the effect of different dendritic models on the model performance and loss landscape.

4.3.1. Membrane potential distribution

In SNNs, neurons use binary spikes to represent different information. When the membrane potential is above the threshold, the neuron is excited and outputs 1, while when the membrane potential is below the threshold, the neuron is resting and outputs 0. Therefore, if the membrane potential can be concentrated around the neuron's resting potential and threshold voltage, then the binary spikes can more accurately approximate the membrane potential distribution. To further illustrate that nonlinear dendrites can bring better characterization ability and more minor transformation errors to SNNs, we analyze the distribution of membrane potentials of neurons at different depths on the DVS-Gesture dataset and the corresponding transformation errors.

As shown in Fig. 5, we conduct experiments on DVS-Gesture with SNN5 structure, and we demonstrate the effect of our DSGM and DTAM on the distribution of membrane potentials in different layers. Such distribution makes it impossible to efficiently characterize the distribution of membrane potentials by binary spikes, which creates a gap between membrane potentials and spikes and affects the performance of SNNs. When using linear and excitatory synaptic models, the membrane potential tends to show a single-peaked distribution. In contrast, neurons with DSGM and DTAM show more excellent adaptation to external stimuli, with the membrane potential showing a multimodal distribution and concentrated near the resting potential and threshold voltage. The adaptability of neurons with DSGM and DTAM enables binary spikes to characterize the membrane potential of neurons better and improve the information transmission efficiency of SNNs.

To verify the adaptability of DSGM, we also test on SNNs using soft-reset neurons, as shown in Eq. (7).

$$u_{t+1,i} = (\frac{1}{\tau}(x_t - u_{t,i}) + u_t) - v_{th}s_{t,i} \tag{7}$$

Unlike the hard-reset neuron in Eq. (1), the soft-reset neuron is soft-reset by subtracting the threshold voltage after the spike is delivered. As shown in the last row of Fig. 5, we find that for the soft-reset neurons, by applying DSGM, the membrane potential can also show a multi-peak distribution and reduce the transformation error of SNNs.

As shown in Fig. 6, we analyzed the effect of different dendritic models on the transformation error of SNNs for all channels in the model corresponding to Fig. 5. For different depth neurons, DSGM brings less transformation error to the SNNs.

4.3.2. Spike sparsity analysis

Fig. 7 shows the firing rate of neurons in different layers related to the dendrite type during training on the DVS-Gesture dataset. In the initial stage of training, the shallow layer neurons have a higher firing rate. This is due to the random initialization of the synaptic weights, which makes it difficult to activate the deep neurons, while the nonlinear design of DSGM enables the deep neurons of SNNs to have higher firing rates, which facilitates the rapid convergence of the network. As the training progresses, in the fifth epoch, the neurons of different layers have a higher firing rate than the initial stage to pass the information from the shallow layers to the deeper neurons. Subsequently, the firing rate of the shallow neurons decreases to better adapt to the external stimuli.

At different stages of training, excitatory dendritic models result in higher firing rates, which can lead to more significant energy overhead. The linear dendritic model, like the excitatory dendritic model, has a considerable variation in the firing rate of different layers as the training progresses, which can easily lead to spike disappearance and spike explosion, which can affect the performance of SNNs. The DSGM, on the other hand, has a more flexible dendritic model design, which can ensure a relatively stable firing rate of neurons in different layers at different training stages and thus performs better.

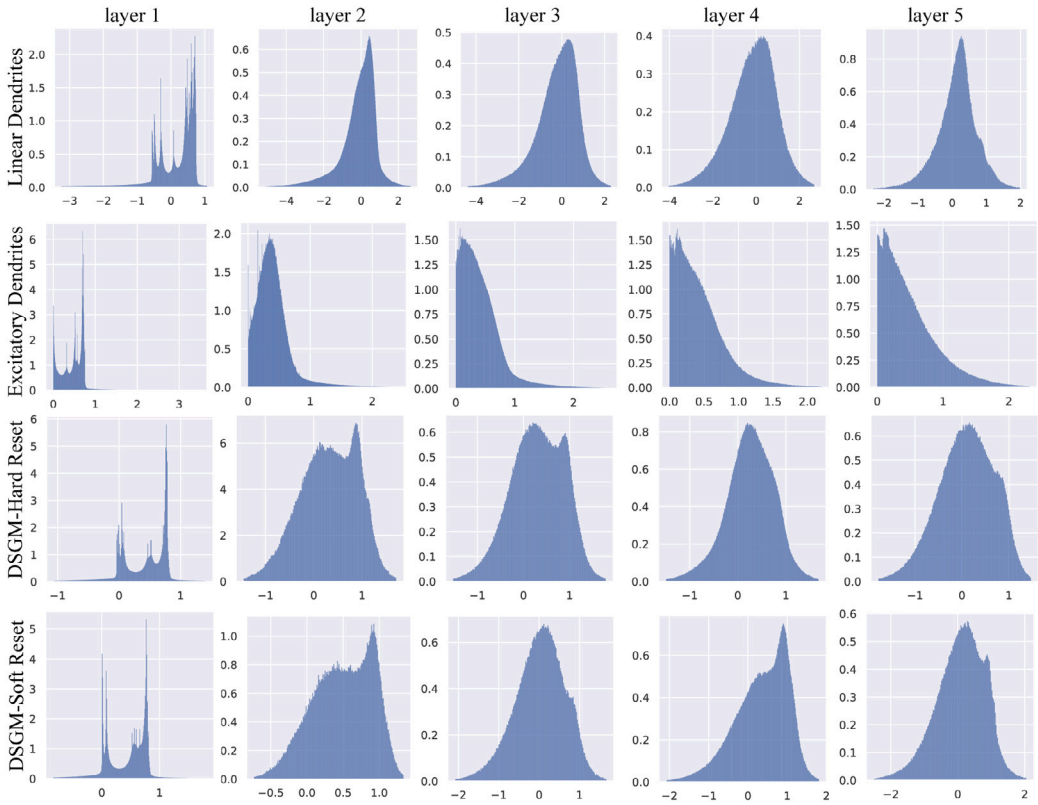

Fig. 5. Distribution of postsynaptic neuronal membrane potential at different depths on the DVS-Gesture.

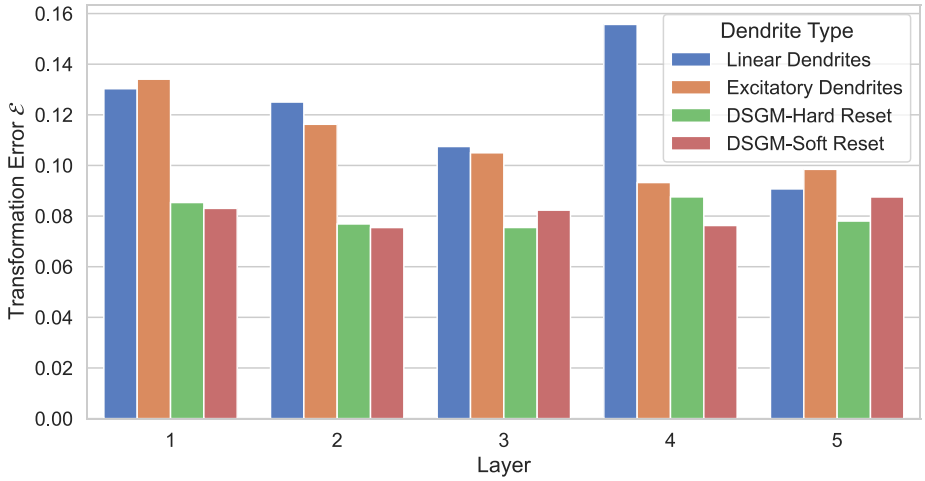

Fig. 6. Transformation errors of neurons with different dendritic types at different depths.

4.3.3. Comparison of dendrite models

The DSGM can give the spiking neuron a larger representation space and realize the reshaping of the input current. To further illustrate the effectiveness of DSGM, we compared DSGM with the nonlinear activation functions commonly used in ANNs such as ReLU (Glorot, Bordes, & Bengio, 2011), ELU (Clevert, Unterthiner, & Hochreiter, 2015), and GeRU (Hendrycks & Gimpel, 2016), and the results are shown in Table 3.

Table 3 shows the performance of dendritic models with different activation functions on the CIFAR10 and DVS-CIFAR10 datasets. Compared with the other dendritic models, DSGM confers greater adaptability to the spiking neuron, allowing it to better adapt to the input current's shape and make more targeted adjustments to the membrane potential of the postsynaptic neuron. The other nonlinear functions do not significantly improve the model's performance compared to the linear dendritic model. This is because these functions cannot adjust to the gain of the input current, making the neuron more likely to deliver more spikes, thus affecting the performance of the model.

4.3.4. Loss landscapes

To further examine the 2D landscapes (Li, Xu, Taylor, Studer, & Goldstein, 2018) of SNNs near the local minima to demonstrate the effect of DSGM and DTAM on the model's generalization ability. As shown in Fig. 8, DSGM and DTAM can help SNNs find flatter minima.

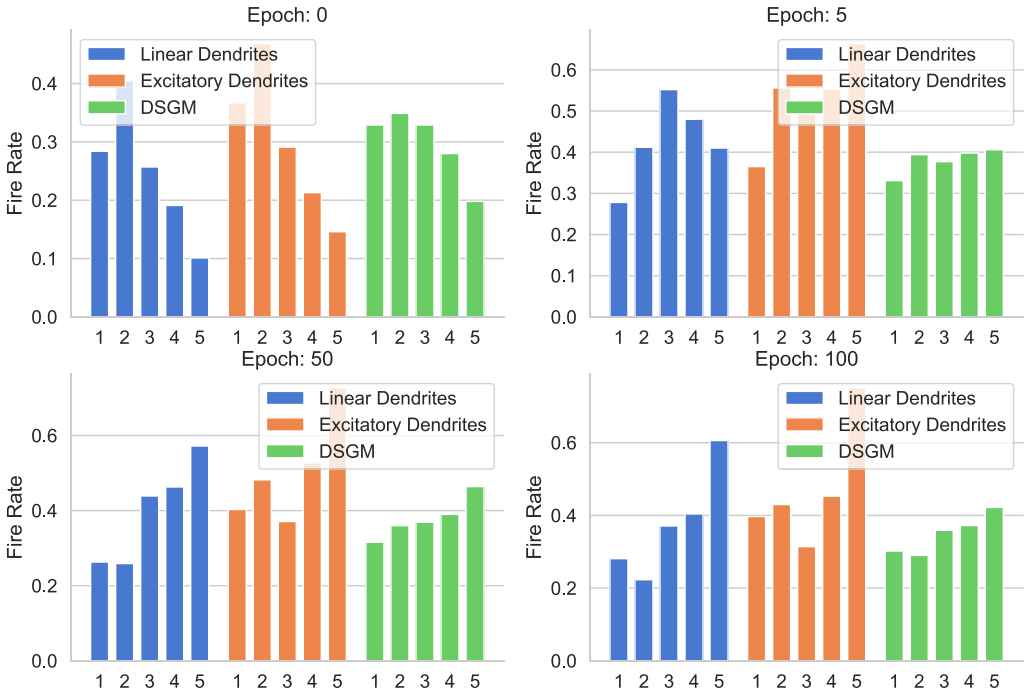

Fig. 7. Firing rates of different dendritic models at different training stages.

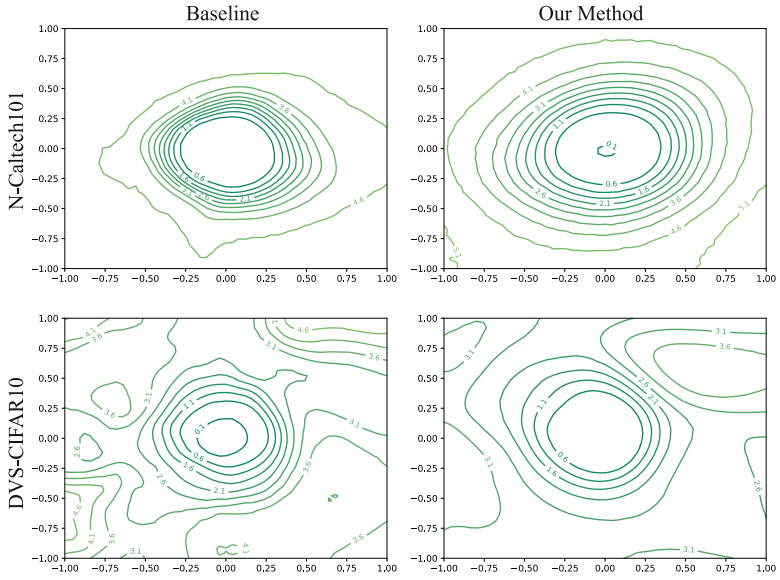

Fig. 8. Loss landscape of VGGSNN with/without DSGM and DTAM. Validation is performed on N-Caltech101 and DVS-CIFAR10.

Table 3
Comparison of different dendrite models.

Dataset	Model	Linear	ReLU	ELU	GeLU	DSGM
CIFAR10	ResNet-19	95.98	96.01	95.86	96.12	**96.41**
DVS-CIFAR10	VGGSNN	84.80	84.53	84.86	84.72	**85.53**

4.4. Ablation studies

We perform ablation studies on several datasets to better illustrate the role of DSGM and DTAM. The qualitative and quantitative analyses will be given.

4.4.1. Qualitative analysis

Our model includes DSGM, which is guided by three adjustable parameters: γ, ξ, and β. Each layer and channel in the network is assigned its unique DSGM configuration. To obtain robust evidence for the changing trends of DSGM during the training process, we conducted five runs using the SNN5 architecture on the DVS-CIFAR10 dataset and visualized the variations of different parameters across all channels, as depicted in Fig. 9.

Throughout the training process, parameters γ and ξ converge swiftly, suggesting that the DSGM rapidly adjusts the polarity of the input current at the initial stages of training. Subsequently, β is delicately adjusted to regulate the inhibitory input's stimulus intensity over time. These recurring observations demonstrate the ability of these

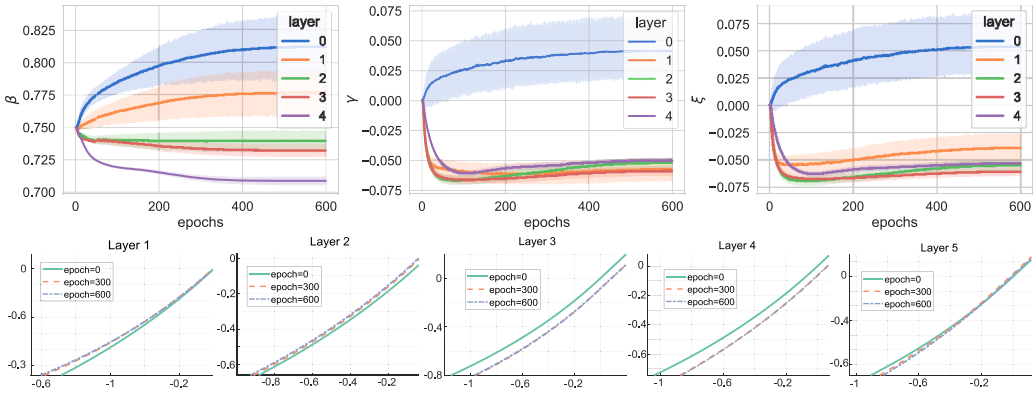

Fig. 9. Variation of parameters in DSGM during training.

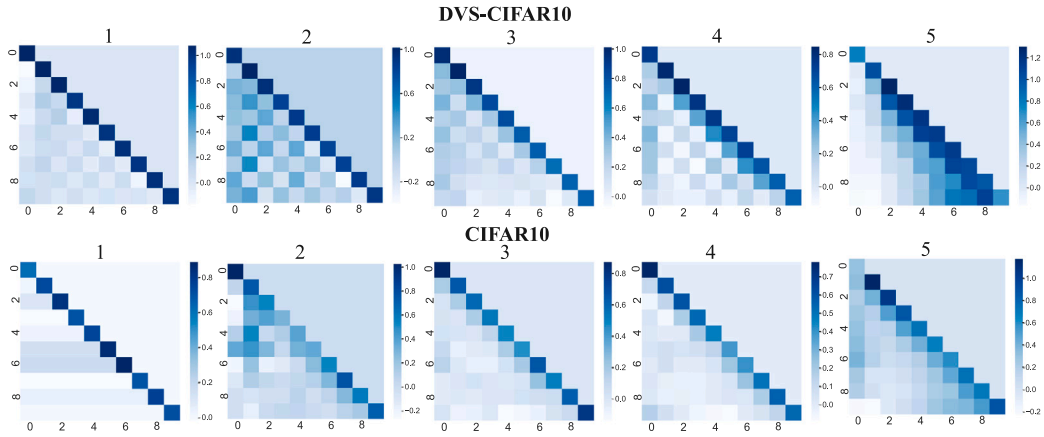

Fig. 10. DTAM weight matrices on the DVS-CIFAR10 and CIFAR10 datasets.

parameters to maintain spike distributions in a relatively balanced state.

As depicted in Fig. 10, we present the visualizations of the DTAM matrices on the CIFAR10 and DVS-CIFAR10 datasets. The DTAM dynamically adjusts the influence of input currents on membrane potentials for different time steps, endowing significance to inputs at various moments. This enables a more precise encoding of spike representation.

To further demonstrate the attention mechanism of DTAM in the temporal dimension, we retrain the model on DVS-Gesture with a longer sequence of length 64, 128, 256, and 512, and visualize the DTAM weights, as shown in Fig. 11.

Fig. 11 shows the distribution of DTAM weights for different lengths of sequences excluding the current moment. At longer sequences, the distribution of DTAM is more pronounced, and the neurons pay more attention to the input currents at closer moments.

The attention mechanism is one of the common advanced cognitive functions in the brain, and it plays a crucial role in many deep learning models such as Transformer (Vaswani et al., 2017). Transformer utilizes a self-attention mechanism to adaptively adjust the weights of different spatial positions by calculating matrices for Query, Key, and Value. This method enhances representational power without significantly increasing the number of parameters. However, it incurs computational costs during inference due to the need for online computation of self-attention matrices.

In contrast, DTAM assigns dynamic importance to different time steps by utilizing a pre-defined matrix. This matrix is updated concurrently with other parameters during model training, but remains fixed during inference. This enables DTAM to assign different weights to input currents at different historical time steps, thereby influencing the behavior of neurons. As shown in Eq. (6), DTAM is a linear operation in the temporal dimension, resulting in relatively low computational overhead. Furthermore, each layer of neurons in the network shares the same DTAM matrix, leading to improved energy efficiency. Importantly, DTAM is designed with consideration for the temporal characteristics of SNNs, where early time steps should not be influenced by subsequent ones. Therefore, DTAM better simulates the information processing mechanisms in the brain, thus enhancing the performance and efficiency of the model.

4.4.2. Quantitative analysis

As shown in Table 4, to investigate the effect of DSGM and DTAM on the performance and convergence speed of SNNs, we conducted an ablation study on CIFAR10, DVS-CIFAR10, and N-Caltech101.

Both DSGM and DTAM contribute to the improvement of model performance. Due to less dynamic characteristics and shorter simulation time, DTAM does not contribute significantly to the classification task of SNNs on static datasets due to less dynamic characteristics and shorter simulation time. Both DSGM and DTAM effectively improve the model performance for event datasets with richer temporal dynamics.

Fig. 12 shows the training curve when using different modules. DSGM and DTAM enable the neuron's membrane potential to adapt to different input currents at different moments, thus enabling faster adaptation to external stimuli.

The nonlinear mapping of the DSGM for the input current can be divided into two parts: linear and exponential. To further verify the role of these two components in DSGM, we perform an ablation analysis on the components of DSGM, and the results are shown in Table 5.

We decompose the mapping of DSGM for inhibitory input currents into linear and exponential components and conduct ablation experiments on the CIFAR10 and DVS-CIFAR10 datasets, as shown in Table 5.

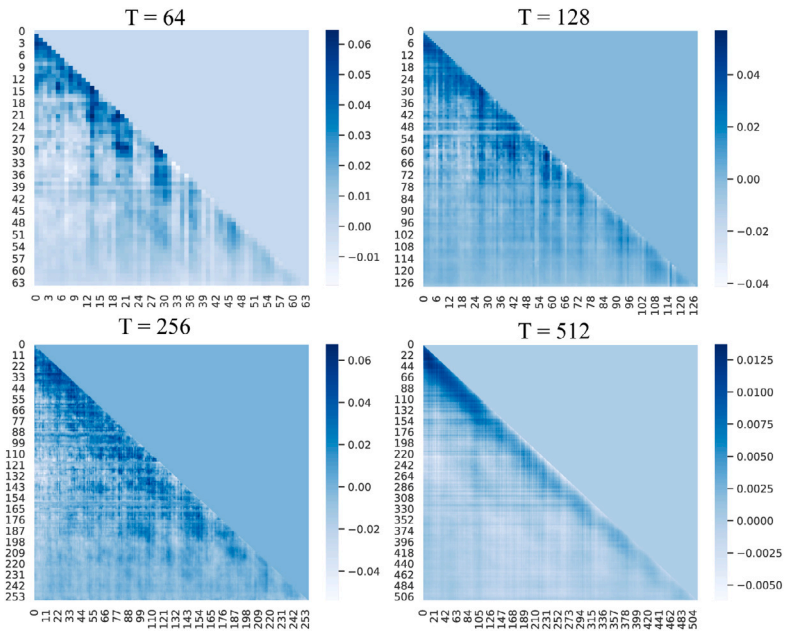

Fig. 11. DTAM weight matrices on the DVS-Gesture dataset with long sequences.

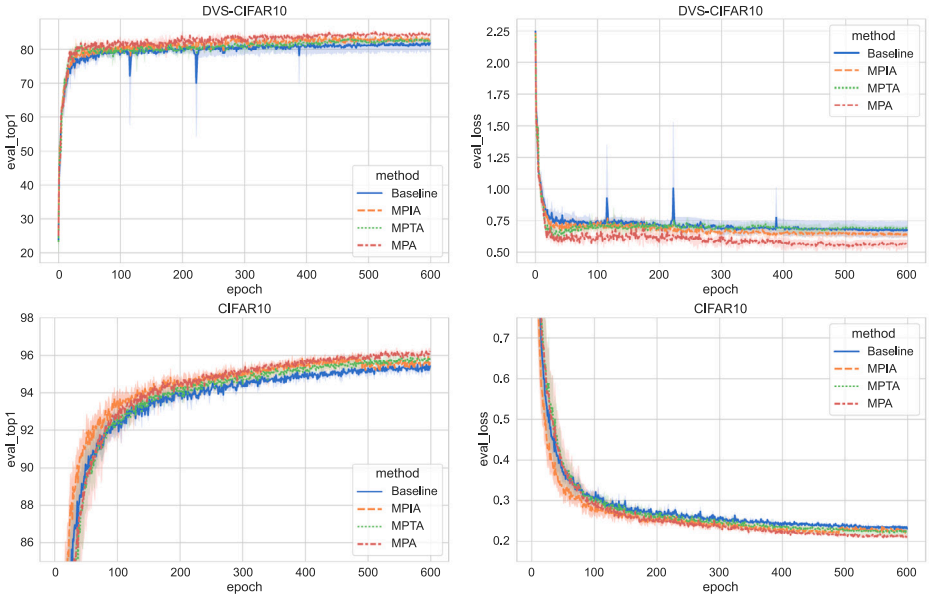

Fig. 12. Training curves on CIFAR10 and DVS-CIFAR10.

Table 4
Ablation study on DSGM and DTAM.

Method	CIFAR10	DVS-CIFAR10	N-Caltech101
Baseline	96.03	83.4	84.37
DSGM	96.32	84.6	85.23
DTAM	95.98	84.8	85.11
DSGM+DTAM	**96.41**	**85.53**	**85.26**

Table 5
Ablation analysis on the components of DSGM.

Dataset	Model	$\beta(x-\gamma)+\xi$	$\beta(e^{x-\gamma})+\xi$	DSGM
CIFAR10	ResNet-19	96.11	95.92	**96.41**
DVS-CIFAR10	VGGSNN	85.23	84.96	**85.53**

The experimental results show that the DSGM outperforms the dendritic model using only exponential and linear mappings on both static and DVS datasets. This is because DSGM achieves a continuous and smooth transformation of the input currents while maintaining the nonlinearity of the dendritic model.

5. Conclusion

The SNNs is widely used in modeling various cognitive functions because of its binary spike train information transmission. The sparse information in the network makes SNNs in the spike vanishing state when it is extended to the deep structure and cannot transmit enough information. When the input stimulus is strengthened, there will be a state of spike explosion, which cannot transmit adequate information. Dendrites perform a simple linear summation of presynaptic information in the current information transmission of SNNs, ignoring the

nonlinear computing role. This paper proposes the dendritic spatial gating mechanism and dendritic temporal adjust mechanism due to dendrites' adaptive spatiotemporal processing ability for input information. Through the spatial translation and scaling of the input information and assigning different importance to the input information at different time steps, the spikes of the network are in a relatively balanced state. We conduct experiments on the CIFAR10, CIFAR100, DVS-CIFAR10, DVS-Gesture, and N-Caltech101 datasets and achieve state-of-the-art performance. At the same time, on ImageNet, it achieves a performance comparable to the current state-of-the-art algorithm under a lower latency and lighter network structure.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Key Research and Development Program (Grant No. 2020AAA0107800).

References

Acharya, J., Basu, A., Legenstein, R., Limbacher, T., Poirazi, P., & Wu, X. (2021). Dendritic computing: Branching deeper into machine learning. *Neuroscience*.

Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., 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).

Beniaguev, D., Segev, I., & London, M. (2021). Single cortical neurons as deep artificial neural networks. *Neuron*, *109*(17), 2727–2739.

Bi, G.-q., & Poo, M.-m. (1998). Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience*, *18*(24), 10464–10472.

Bicknell, B. A., & Häusser, M. (2021). A synaptic learning rule for exploiting nonlinear dendritic computation. *Neuron*, *109*(24), 4001–4017.

Bohte, S. M. (2011). Error-backpropagation in networks of fractionally predictive spiking neurons. In *Artificial neural networks and machine learning–ICANN 2011: 21st international conference on artificial neural networks, Espoo, Finland, June 14-17, 2011, proceedings, Part I 21* (pp. 60–68). Springer.

Bu, T., Fang, W., Ding, J., Dai, P., Yu, Z., & Huang, T. (2021). Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks. In *International conference on learning representations*.

Clevert, D.-A., Unterthiner, T., & Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289.

Deng, S., Li, Y., Zhang, S., & Gu, S. (2021). Temporal efficient training of spiking neural network via gradient re-weighting. In *International conference on learning representations*.

Duan, C., Ding, J., Chen, S., Yu, Z., & Huang, T. (2022). Temporal effective batch normalization in spiking neural networks. *Adv. Neural Inf. Process. Syst.*, *35*, 34377–34390.

Fang, W., Yu, Z., Chen, Y., Huang, T., Masquelier, T., & Tian, Y. (2021). Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, *34*, 21056–21069.

Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., & 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).

Furber, S. B., Lester, D. R., Plana, L. A., Garside, J. D., Painkras, E., Temple, S., et al. (2012). Overview of the SpiNNaker system architecture. *IEEE Transactions on Computers*, *62*(12), 2454–2467.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 315–323). JMLR Workshop and Conference Proceedings.

Guo, Y., Tong, X., Chen, Y., Zhang, L., Liu, X., Ma, Z., et al. (2022). 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* (pp. 326–335).

Han, B., Srinivasan, G., & Roy, K. (2020). Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13558–13567).

Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415.

Kim, Y., Li, Y., Park, H., Venkatesha, Y., & Panda, P. (2022). Neural architecture search for spiking neural networks. In *European Conference on Computer Vision* (pp. 36–56). Springer.

Krizhevsky, A., Hinton, G., et al. (2009). *Learning multiple layers of features from tiny images*. Citeseer.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems, vol. 25*.

Kugele, A., Pfeil, T., Pfeiffer, M., & Chicca, E. (2020). Efficient processing of spatio-temporal data streams with spiking neural networks. *Frontiers in Neuroscience*, *14*, 439.

Li, H., Liu, H., Ji, X., Li, G., & Shi, L. (2017). Cifar10-dvs: An event-stream dataset for object classification. *Frontiers in Neuroscience*, *11*, 309.

Li, H., Xu, Z., Taylor, G., Studer, C., & Goldstein, T. (2018). Visualizing the loss landscape of neural nets. In *Advances in neural information processing systems, vol. 31*.

Li, Y., & Zeng, Y. (2022). Efficient and accurate conversion of spiking neural network with burst spikes. arXiv preprint arXiv:2204.13271.

Li, Y., Zhao, D., & Zeng, Y. (2022). BSNN: Towards faster and better conversion of artificial neural networks to spiking neural networks with bistable neurons. *Frontiers in neuroscience*, *16*, 991851.

Liu, Q., Xing, D., Feng, L., Tang, H., & Pan, G. (2022). Event-based multimodal spiking neural network with attention mechanism. In *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing* (pp. 8922–8926). IEEE.

Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, *10*(9), 1659–1671.

Na, B., Mok, J., Park, S., Lee, D., Choe, H., & Yoon, S. (2022). Autosnn: Towards energy-efficient spiking neural networks. In *International Conference on Machine Learning* (pp. 16253–16269). PMLR.

Orchard, G., Jayawant, A., Cohen, G. K., & Thakor, N. (2015). Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in Neuroscience*, *9*, 437.

Ramesh, B., Yang, H., Orchard, G., Le Thi, N. A., Zhang, S., & Xiang, C. (2019). Dart: Distribution aware retinal transform for event-based cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(11), 2767–2780.

Rathi, N., & Roy, K. (2021). 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*.

Rathi, N., Srinivasan, G., Panda, P., & Roy, K. (2019). Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation. In *International conference on learning representations*.

RoSE, G. J., & Call, S. J. (1992). Evidence for the role of dendritic spines in the temporal filtering properties of neurons: The decoding problem and beyond. *Proceedings of the National Academy of Sciences*, *89*(20), 9662–9665.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Sengupta, A., Ye, Y., Wang, R., Liu, C., & Roy, K. (2019). Going deeper in spiking neural networks: VGG and residual architectures. *Frontiers in Neuroscience*, *13*, 95.

Shaban, A., Bezugam, S. S., & Suri, M. (2021). An adaptive threshold neuron for recurrent spiking neural networks with nanodevice hardware implementation. *Nature Communications*, *12*(1), 1–11.

Shen, G., Zhao, D., & Zeng, Y. (2022a). Backpropagation with biologically plausible spatiotemporal adjustment for training deep spiking neural networks. *Patterns*, Article 100522.

Shen, G., Zhao, D., & Zeng, Y. (2022b). Eventmix: An efficient augmentation strategy for event-based data. *arXiv preprint arXiv:2205.12054*.

Shrestha, S. B., & Orchard, G. (2018). Slayer: Spike layer error reassignment in time. In *Advances in neural information processing systems, vol. 31*.

Szalai, M. L., Kevwitch, R. M., & McGrath, D. V. (2003). Geometric disassembly of dendrimers: Dendritic amplification. *Journal of the American Chemical Society*, *125*(51), 15688–15689.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Advances in neural information processing systems, vol. 30*.

Wu, Y., Deng, L., Li, G., Zhu, J., & Shi, L. (2018). Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience*, *12*, 331.

Wu, Y., Deng, L., Li, G., Zhu, J., Xie, Y., & Shi, L. (2019). Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 01* (pp. 1311–1318).

Wu, Z., Zhang, H., Lin, Y., Li, G., Wang, M., & Tang, Y. (2021). Liaf-net: Leaky integrate and analog fire network for lightweight and efficient spatiotemporal information processing. *IEEE Transactions on Neural Networks and Learning Systems*.

Xing, Y., Di Caterina, G., & Soraghan, J. (2020). A new spiking convolutional recurrent neural network (SCRNN) with applications to event-based hand gesture recognition. *Frontiers in Neuroscience*, *14*, 1143.

Xu, B., Wang, N., Chen, T., & Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853.

Yao, M., Gao, H., Zhao, G., Wang, D., Lin, Y., Yang, Z., et al. (2021). Temporal-wise attention spiking neural networks for event streams classification. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10221–10230).

Yao, X., Li, F., Mo, Z., & Cheng, J. (2022). Glif: A unified gated leaky integrate-and-fire neuron for spiking neural networks. *Adv. Neural Inf. Process. Syst.*, *35*, 32160–32171.

Zeng, Y., Zhao, D., Zhao, F., Shen, G., Dong, Y., Lu, E., et al. (2023). Braincog: a spiking neural network based, brain-inspired cognitive intelligence engine for brain-inspired ai and brain simulation. *Patterns*, *4*(8).

Zhang, W., & Li, P. (2020). Temporal spike sequence learning via backpropagation for deep spiking neural networks. *Advances in Neural Information Processing Systems*, *33*, 12022–12033.

Zhao, D., Zeng, Y., & Li, Y. (2022). BackEISNN: A deep spiking neural network with adaptive self-feedback and balanced excitatory–inhibitory neurons. *Neural Networks*.

Zheng, H., Wu, Y., Deng, L., Hu, Y., & Li, G. (2021). Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence, vol. 35, no. 12* (pp. 11062–11070).

Zucker, R. S., & Regehr, W. G. (2002). Short-term synaptic plasticity. *Annual Review of Physiology*, *64*(1), 355–405.