frontiers | Frontiers in Neuroscience

TYPE Review
PUBLISHED 31 July 2024
DOI 10.3389/fnins.2024.1383844

Direct training high-performance deep spiking neural networks: a review of theories and methods

Chenlin Zhou[1†], Han Zhang[1,2†], Liutao Yu[1†], Yumin Ye[1], Zhaokun Zhou[1,3], Liwei Huang[1,4], Zhengyu Ma[1*], Xiaopeng Fan[1,2], Huihui Zhou[1] and Yonghong Tian[1,3,4]

[1]Peng Cheng Laboratory, Shenzhen, China, [2]Faculty of Computing, Harbin Institute of Technology, Harbin, China, [3]School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, Shenzhen, China, [4]National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University, Beijing, China

EDITED BY
Priyadarshini Panda,
Yale University, United States

REVIEWED BY
Lei Deng,
Tsinghua University, China
Chenglong Zou,
Peking University, China

*CORRESPONDENCE
Zhengyu Ma
✉ mazhy@pcl.ac.cn

[†]These authors have contributed equally to this work and share first authorship

RECEIVED 08 February 2024
ACCEPTED 03 July 2024
PUBLISHED 31 July 2024

CITATION
Zhou C, Zhang H, Yu L, Ye Y, Zhou Z, Huang L, Ma Z, Fan X, Zhou H and Tian Y (2024) Direct training high-performance deep spiking neural networks: a review of theories and methods. *Front. Neurosci.* 18:1383844.
doi: 10.3389/fnins.2024.1383844

Spiking neural networks (SNNs) offer a promising energy-efficient alternative to artificial neural networks (ANNs), in virtue of their high biological plausibility, rich spatial-temporal dynamics, and event-driven computation. The direct training algorithms based on the surrogate gradient method provide sufficient flexibility to design novel SNN architectures and explore the spatial-temporal dynamics of SNNs. According to previous studies, the performance of models is highly dependent on their sizes. Recently, direct training deep SNNs have achieved great progress on both neuromorphic datasets and large-scale static datasets. Notably, transformer-based SNNs show comparable performance with their ANN counterparts. In this paper, we provide a new perspective to summarize the theories and methods for training deep SNNs with high performance in a systematic and comprehensive way, including theory fundamentals, spiking neuron models, advanced SNN models and residual architectures, software frameworks and neuromorphic hardware, applications, and future trends.

KEYWORDS

deep spiking neural network, direct training, transformer-based SNNs, residual connection, energy efficiency, high performance

1 Introduction

Regarded as the third generation of neural network (Maass, 1997), the brain-inspired spiking neural networks (SNNs) are potential competitors to traditional artificial neural networks (ANNs) in virtue of their high biological plausibility, and low power consumption when implemented on neuromorphic hardware (Roy et al., 2019). In particular, the utilization of binary spikes allows SNNs to adopt low-power accumulation (AC) instead of the traditional high-power multiply-accumulation (MAC), leading to significantly enhanced energy efficiency and making SNNs increasingly popular (Chen et al., 2023).

There are two mainstream pathways to obtain deep SNNs: ANN-to-SNN conversion and direct training through the surrogate gradient method. Firstly, in ANN-to-SNN conversion (Cao et al., 2015; Hunsberger and Eliasmith, 2015; Rueckauer et al., 2017; Bu et al., 2022; Meng et al., 2022; Wang Y. et al., 2022), a pre-trained ANN is converted to an SNN by replacing the ReLU activation layers with spiking neurons and adding scaling operations like weight normalization and threshold balancing. This conversion process suffers from long converting time steps, which causes high computational consumption in practice. In addition, the converted SNNs obtained in this way are constrained by the original ANNs' architecture and are hard to adapt to dynamic signal (DVS, DAVIS, ATIS data) processing. Thus, the direct exploration of the virtues of SNNs is limited in ANN-to-SNN conversion. Secondly, in the field of direct training, SNNs are unfolded over simulation time steps and trained with backpropagation through time (Lee et al., 2016; Shrestha and Orchard, 2018). Due to the non-differentiability of spiking neurons, the surrogate gradient method is employed for backpropagation (Neftci et al., 2019; Lee et al., 2020b; Fang et al., 2021a,b; Zhou Z. et al., 2023). On one hand, this direct training method can handle temporal data and also achieve decent performance on large-scale static datasets, with only a few time steps. On the other hand, it can provide sufficient flexibility for designing novel architectures specifically for SNNs and exploring the properties of SNNs directly. Therefore, the direct training method has received more attention recently.

Given the significant benefits and rapid advancement of directly trained deep SNNs, particularly the emergence of high-performance transformer-based SNNs, this review systematically and comprehensively summarizes the theories and methods for directly trained deep SNNs. Combining theory fundamentals, spiking neuron models, advanced SNN models and residual architectures, software frameworks and neuromorphic hardware, applications, and future trends, this article offers fresh perspectives into the field of SNNs. This review is structured as follows: Section 2 presents the evolution and recent advancements in spiking neuron models. Section 3 introduces the fundamental principles of spiking neural networks. Section 4 focuses on the most recent advanced SNN models and architectures, especially transformer-based SNNs. Section 5 concludes the software frameworks for training SNNs and the development of neuromorphic hardware. Section 6 summarizes the applications of deep SNNs. Finally, Section 7 points out future research trends and concludes this review.

2 Spiking neuron models

LIF (Leaky Integrate-and-Fire) neuron is one of the most commonly used neurons in SNNs (Zhou et al., 2023a,b; Zhou Z. et al., 2023), which is simple but retains biological characteristics (Figure 1A). The dynamics of LIF are described as Equations (1–3):

$$H[t] = V[t-1] + \frac{1}{\tau}\left(X[t] - \left(V[t-1] - V_{reset}\right)\right), \qquad (1)$$

$$S[t] = \Theta\left(H[t] - V_{th}\right), \qquad (2)$$

$$V[t] = H[t]\left(1 - S[t]\right) + V_{reset}S[t], \qquad (3)$$

where τ in Equation (1) is the membrane time constant, $X[t]$ is the input current at time step t. V_{reset} represents the reset potential, V_{th} represents the spike firing threshold, $H[t]$ and $V[t]$ represent the membrane potential before and after spike firing at time step t, respectively. $\Theta(v)$ is the Heaviside step function, if $v \geq 0$ then $\Theta(v) = 1$, meaning a spike is generated; otherwise $\Theta(v) = 0$. $S[t]$ represents whether a neuron fires a spike at time step t.

LIF also comes with notable limitations in practical applications. For instance, LIF needs to manually adjust the hyperparameters, such as membrane time constant τ and firing threshold V_{th}, which constrains its expressiveness. In addition, LIF is simple in modeling, which limits the range of neuronal dynamics. Overall, there is a lack of diversity and flexibility in LIF, which calls for more advanced neuron models to enhance SNNs' performance and broaden their applications. Table 1 lists some recently developed spiking neuron models and their performance on typical tasks.

2.1 Spiking neurons with trainable parameters

Based on LIF, many improved spiking neuron models with trainable parameters have been proposed, which expand the representation space of neurons through parameter learning and improve the expression ability of SNNs. Fang et al. proposed Parametric LIF (PLIF) (Fang et al., 2021b) by using trainable membrane time constant as follows:

$$H[t] = V[t-1] + k(a)\left(X[t] - \left(V[t-1] - V_{reset}\right)\right), \qquad (4)$$

where $k(a)$ in Equation (4) denotes a clamp function and $k(a) = \frac{1}{1+exp(-a)} \in (0,1)$. The trainable membrane-related parameter of PLIF is biologically plausible, as neurons in the brain are heterogeneous. LTMD (Wang S. et al., 2022) also leverages this biological plausibility but approaches it differently by employing learnable firing thresholds. An increase in the threshold of LTMD results in a reduction of output spikes, making an SNN less sensitive to its input and thus more robust. On the contrary, a decrease in the threshold leads to an increment of output spikes, making an SNN more sensitive to its input, which is particularly beneficial for processing transient small signals. Therefore, the learnable threshold $V_{th} = \text{tank}(k)$, of which k is trainable, can lead to the optimal sensitivity of an SNN.

Diet-SNN (Rathi and Roy, 2023) adopts an end-to-end gradient descent optimization algorithm to train the membrane-related parameters and firing thresholds of LIF neurons while optimizing the network weights. The trained neuron parameters selectively reduce the membrane potential, making spikes in the network sparser, thereby improving the computational efficiency of SNN. Spiking neurons with dynamic thresholds are adopted in LSNN (Bellec et al., 2018). After firing a spike each time, the firing threshold of a neuron will increase by a fixed amount, and then it will decay exponentially according to the time constant. Adaptive spiking neuron (ASN) (Yin et al., 2020) was proposed for sequence and streaming media tasks. In ASN, the time constant of membrane potential is trainable. In addition, similar to LSNN, the firing threshold will increase after each spike of the neuron, thus improving sparsity and efficiency.

In KLIF (Jiang and Zhang, 2023), a trainable scaling factor k and a nonlinear ReLU activation function are inserted between charging and firing. The dynamics of KLIF can be described by Equations (1), (5–7).

$$F[t] = \text{ReLU}\left(kH[t]\right), \qquad (5)$$

$$S[t] = \Theta\left(F[t] - V_{th}\right), \qquad (6)$$

$$V[t] = F[t]\left(1 - S[t]\right) + V_{reset}S[t]. \qquad (7)$$

Compared with LIF, KLIF can automatically adjust the membrane potential and the gradient of backpropagation within

TABLE 1 Overview of spiking neurons for direct training and their performance.

Method	Architecture	Dataset	Acc (%)	Training
PLIF (Fang et al., 2021b)	PLIF-Net	CIFAR10	93.50	Time dependent
LTMD (Wang S. et al., 2022)	DenseNet	CIFAR10	94.19	Time dependent
GLIF (Yao et al., 2022)	ResNet-34	CIFAR10	95.03	Time dependent
MLF (Feng L. et al., 2022)	DS ResNet	CIFAR10	94.25	Time dependent
LIFB (Shen et al., 2023)	ResNet-19	CIFAR10	96.32	Time dependent
Deit-SNN (Rathi and Roy, 2023)	VGG16	CIFAR10	93.44	Time dependent
KLIF (Jiang and Zhang, 2023)	CNN	CIFAR10	92.52	Time dependent
MT-SNN (Wang X. et al., 2023)	MT-VGG9	CIFAR10	94.74	Time dependent
PSN (Fang et al., 2023b)	PLIF-Net	CIFAR10	95.32	Parallel
GLIF (Yao et al., 2022)	ResNet-34	CIFAR100	77.35	Time dependent
Deit-SNN (Rathi and Roy, 2023)	VGG16	CIFAR100	69.67	Time dependent
LIFB (Shen et al., 2023)	ResNet-19	CIFAR100	78.31	Time dependent
MT-SNN (Wang X. et al., 2023)	MT-VGG9	CIFAR100	75.53	Time dependent
GLIF (Yao et al., 2022)	ResNet-34	ImageNet	69.09	Time dependent
Deit-SNN (Rathi and Roy, 2023)	VGG16	ImageNet	69.00	Time dependent
LIFB (Shen et al., 2023)	SEW ResNet-34	ImageNet	70.02	Time dependent
PSN (Fang et al., 2023b)	SEW ResNet-34	ImageNet	70.54	Parallel
PLIF (Fang et al., 2021b)	PLIF-Net	CIFAR10-DVS	74.80	Time dependent
GLIF (Yao et al., 2022)	ResNet-34	CIFAR10-DVS	78.10	Time dependent
MLF (Feng L. et al., 2022)	DS ResNet	CIFAR10-DVS	70.36	Time dependent
LTMD (Wang S. et al., 2022)	DenseNet	CIFAR10-DVS	73.30	Time dependent
KLIF (Jiang and Zhang, 2023)	CNN	CIFAR10-DVS	70.90	Time dependent
MT-SNN (Wang X. et al., 2023)	MT-VGG9	CIFAR10-DVS	76.30	Time dependent
PSN (Fang et al., 2023b)	VGG	CIFAR10-DVS	85.90	Parallel
PLIF (Fang et al., 2021b)	PLIF-Net	DVS128-Gesture	97.57	Time dependent
MLF (Feng L. et al., 2022)	DS ResNet	DVS128-Gesture	97.29	Time dependent
KLIF (Jiang and Zhang, 2023)	CNN	DVS128-Gesture	94.10	Time dependent
LSNN (Bellec et al., 2018)	LSTM	Sequential MNIST	96.40	Time dependent
ASN (Yin et al., 2020)	RNN	PS-MNIST	97.90	Time dependent
SPSN (Yarga and Wood, 2023)	MLP	SHD	86.89	Parallel

the neuron. GLIF (Yao et al., 2022) introduces a gating unit that fuses multiple biometric features, with the ratio of these features adjusted by a trainable gating factor. Moreover, inspired by various spiking patterns of brain neurons, LIFB (Shen et al., 2023) has three modes: resting, regular spiking, and burst spiking. The density of the burst spiking can be learned automatically, which greatly enriches the representation capability of neurons.

In addition, there are other studies trying to improve performance by multi-level firing thresholds instead of trainable parameters. To reduce the performance loss caused by the transmission of binarized spikes in the network, MT-SNN (Wang X. et al., 2023) introduces multi-level firing thresholds. MT-SNN performs convolution operations on the binarized spikes generated by different firing thresholds and then sums them up. Similarly, MLF (Feng L. et al., 2022) can also fire spikes under different firing thresholds, thus improving the performance of SNNs.

2.2 Parallel spiking neurons

A typical neuron model like LIF is time-dependent, that is, its state at time t relies on its state at time $t-1$, resulting in a high computation load. Fang et al. (2023b) proposed a parallel spiking neuron (PSN) to accelerate the computation by parallel computing. By eliminating the resetting process, they represent the charging

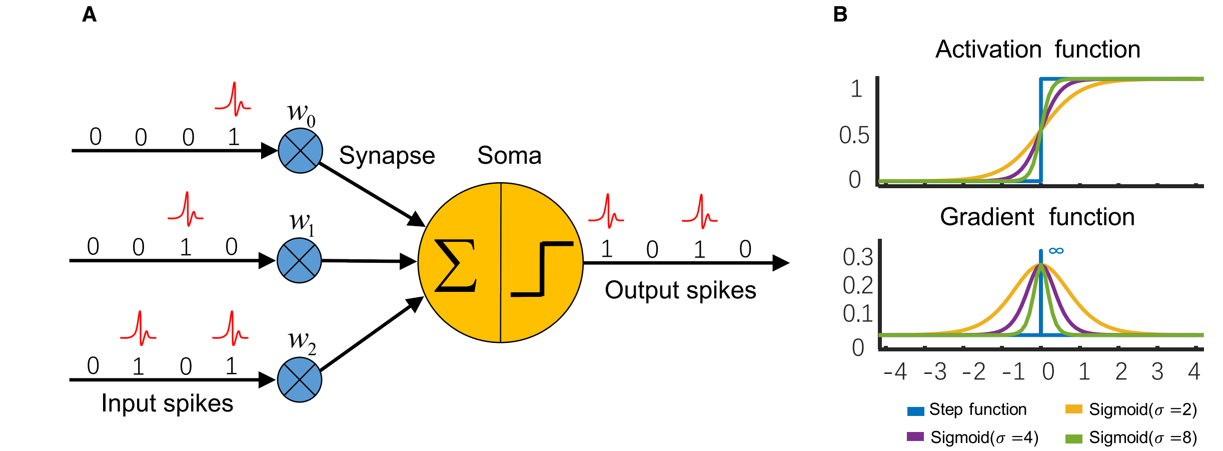

FIGURE 1
(A) The scheme of a spiking neuron, of which the input and output are both binary spikes. **(B)** The sigmoid function approximates the Heaviside activation function of a spiking neuron, and its derivative can be utilized to calculate gradients during backpropagation.

process of PSN by a non-iterative equation as Equation (8):

$$H[t] = \sum_{i=0}^{T-1} W_{t,i} \cdot X[i], \tag{8}$$

where $W_{t,i}$ is the weight between input $X[i]$ and membrane potential $H[t]$. For LIF neuron, $W_{t,i} = \frac{1}{\tau_m}(1-\frac{1}{\tau_m})^{t-i}$. The dynamics of PSN are as Equations (9, 10):

$$\boldsymbol{H} = \boldsymbol{WX}, \quad \boldsymbol{W} \in \mathbb{R}^{T\times T}, \boldsymbol{X} \in \mathbb{R}^{T\times N} \tag{9}$$

$$\boldsymbol{S} = \Theta(\boldsymbol{H} - \boldsymbol{B}), \quad \boldsymbol{B} \in \mathbb{R}^{T}, \boldsymbol{S} \in \{0,1\}^{T\times N} \tag{10}$$

where $\boldsymbol{X}$ is the input, $\boldsymbol{W}$ and $\boldsymbol{B}$ are trainable weights and trainable firing thresholds, respectively. $\boldsymbol{H}$ is the membrane potential after charging, and $\boldsymbol{S}$ denotes whether a neuron spikes. N and T are the batch size and the number of time steps, respectively. For step-by-step serial forward computation and variable-length sequence processing, the masked PSN and the sliding PSN are also derived.

The stochastic parallel spiking neuron (SPSN) (Yarga and Wood, 2023) adopts an idea similar to PSN, by removing the resetting mechanism. The neuronal dynamics of SPSN contains two parts, namely parallel leaky integrator and stochastic firing. The leaky integrator is a linear time-invariant system, which can be transformed into the Fourier domain to realize parallel computation. Stochastic firing adaptively adjusts the firing probability through trainable parameters, enhancing the network's capability to process information in a dynamic and efficient manner.

3 Fundamentals of spiking neural networks

3.1 Information coding

To process image data through SNNs, it is essential to first encode the data into spike trains. Rate coding (Adrian and Zotterman, 1926) is the most commonly used information coding method in SNNs, in which the firing rate is proportional to the intensity of the input signal and spikes are typically generated by a Poisson process (Wiener and Richmond, 2003). To encode information more accurately, rate coding requires a longer time window, which leads to a slower information transmission rate. In contrast, utilizing a shorter time window may result in loss of information during encoding, presenting a trade-off between speed and accuracy in information transmission.

Different from rate coding, temporal coding represents information through the timing of spikes. Time-to-first-spike (TTFS) (Park et al., 2020; Guo W. et al., 2021) stands out for its simplicity and efficiency in temporal coding, which uses the time of the first spike fired by the neuron to represent the input signal. TTFS effectively reduces the total number of spikes, thereby accelerating the computation of SNNs. TTFS algorithm can be described as Equation (11):

$$S[t] = \begin{cases} 1, & \text{if } t = \left(\frac{X_{max}-X}{X_{max}}\right) t_{max} \\ 0, & \text{otherwise} \end{cases}, \tag{11}$$

where $S[t]$ represents whether a spike is fired at time t after encoding, t_{max} denotes the maximum time allowed during encoding, X and X_{max} represent the input signal and its maximum value, respectively. In the TTFS encoding method, larger values of the input signal lead to earlier firing of spikes.

3.2 Network training

3.2.1 Surrogate gradient

As the core components of SNNs, neurons are essential for information processing and transmission, since spikes are fired by neurons. However, the firing of spikes involves the non-differentiable Heaviside step function, which presents a significant challenge in the direct training of SNNs. To address the non-differentiability of the Heaviside step function, Neftci et al. (2019)

TABLE 2 Overview of direct training deep SNNs and their performance on ImageNet, CIFAR10, CIFAR100, DVS128-Gesture, CIFAR10-DVS.

Method	Architecture	Param (M)	Time steps	Dataset	Top-1 Acc (%)
Spiking ResNet (Hu et al., 2021a)	ResNet-50	25.56	350	ImageNet	72.75
SEW ResNet (Fang et al., 2021a)	SEW-ResNet-152	60.19	4	ImageNet	69.26
MS-ResNet (Hu et al., 2021b)	MS-ResNet-104	77.28	5	ImageNet	76.02
Att MS-ResNet (Yao et al., 2023b)	Att-MS-ResNet-104	78.37	4	ImageNet	77.08
Spikformer (Zhou Z. et al., 2023)	Spikformer-8-768	66.34	4	ImageNet	74.81
Spikingformer (Zhou et al., 2023a)	Spikingformer-8-768	66.34	4	ImageNet	75.85
CML (Zhou et al., 2023b)	Spikformer-8-768	66.34	4	ImageNet	77.34
Spike-driven Transformer (Yao et al., 2023a)	S-Transformer-8-768	66.34	4	ImageNet	77.07
SpikingResformer (Shi et al., 2024)	SpikingResformer-L	60.38	4	ImageNet	79.40
Spike-driven Transformer V2 (Yao et al., 2024)	Meta-SpikeFormer	55.40	4	ImageNet	80.00
Spikformer V2 (Zhou Z. et al., 2024)	Spikformer V2-8-512	51.55	4	ImageNet	80.38
SGLFormer (Zhang et al., 2024)	SGLFormer-8-768	64.02	4	ImageNet	83.73
QKFormer (Zhou C. et al., 2024)	HST-10-768	64.96	4	ImageNet	85.65
Hybrid training (Rathi et al., 2020)	VGG-11	9.27	125	CIFAR10	92.22
STBP-tdBN (Zheng et al., 2021)	ResNet-19	12.63	4	CIFAR10	92.92
TET (Deng et al., 2022)	ResNet-19	12.63	4	CIFAR10	94.44
MS-ResNet (Hu et al., 2021b)	MS-ResNet-110	–	4	CIFAR10	92.12
Spikformer (Zhou Z. et al., 2023)	Spikformer-4-384	9.32	4	CIFAR10	95.51
Spikingformer (Zhou et al., 2023a)	Spikingformer-4-384	9.32	4	CIFAR10	95.81
CML (Zhou et al., 2023b)	Spikformer-4-384	9.32	4	CIFAR10	96.04
Spike-driven Transformer (Yao et al., 2023a)	S-Transformer-2-512	10.23	4	CIFAR10	95.60
SGLFormer (Zhang et al., 2024)	SGLFormer-4-384	8.85	4	CIFAR10	96.76
Hybrid training (Rathi et al., 2020)	VGG-11	9.27	125	CIFAR100	67.87
STBP-tdBN (Zheng et al., 2021)	ResNet-19	12.63	4	CIFAR100	70.86
TET (Deng et al., 2022)	ResNet-19	12.63	4	CIFAR100	74.47
Spikformer (Zhou Z. et al., 2023)	Spikformer-4-384	9.32	4	CIFAR100	78.21
Spikingformer (Zhou et al., 2023a)	Spikingformer-4-384	9.32	4	CIFAR100	79.21
CML (Zhou et al., 2023b)	Spikformer-4-384	9.32	4	CIFAR100	80.02
Spike-driven Transformer (Yao et al., 2023a)	S-Transformer-2-512	10.28	4	CIFAR100	78.4
SGLFormer (Zhang et al., 2024)	SGLFormer-4-384	8.88	4	CIFAR100	82.26
SEW-ResNet (Hu et al., 2021b)	SEW-ResNet	–	16	DVS128-Gesture	97.9
tdBN (Zheng et al., 2021)	ResNet	–	40	DVS128-Gesture	96.9
Spikformer (Zhou Z. et al., 2023)	Spikformer-2-256	2.57	16	DVS128-Gesture	98.3
Spikingformer (Zhou et al., 2023a)	Spikingformer-2-256	2.57	16	DVS128-Gesture	98.3
CML (Zhou et al., 2023b)	Spikformer-2-256	2.57	16	DVS128-Gesture	98.6
Spike-driven Transformer (Yao et al., 2023a)	S-Transformer-2-256	2.57	16	DVS128-Gesture	99.3
STSA (Wang Y. et al., 2023)	STSFormer-2-256	1.99	16	DVS128-Gesture	98.72

(Continued)

TABLE 2 (Continued)

Method	Architecture	Param (M)	Time steps	Dataset	Top-1 Acc (%)
SGLFormer (Zhang et al., 2024)	SGLFormer-3-256	2.17	16	DVS128-Gesture	98.6
SEW-ResNet (Hu et al., 2021b)	SEW-ResNet	–	16	CIFAR10-DVS	74.4
Spikformer (Zhou Z. et al., 2023)	Spikformer-2-256	2.57	16	CIFAR10-DVS	80.9
Spikingformer (Zhou et al., 2023a)	Spikingformer-2-256	2.57	16	CIFAR10-DVS	81.3
CML (Zhou et al., 2023b)	Spikformer-2-256	2.57	16	CIFAR10-DVS	80.9
Spike-driven Transformer (Yao et al., 2023a)	S-Transformer-2-256	2.57	16	CIFAR10-DVS	80.0
STSA (Wang Y. et al., 2023)	STSFormer-2-256	1.99	16	CIFAR10-DVS	79.93
SGLFormer (Zhang et al., 2024)	SGLFormer-3-256	2.58	10	CIFAR10-DVS	82.9

proposed the Surrogate Gradient (SG) algorithm. In SG, the Heaviside step function is adopted to generate spikes during forward propagation, and differentiable functions are adopted for gradient calculation during backpropagation. Notably, SG functions could vary according to the networks. For instance, the SG function used in SEW ResNet (Fang et al., 2021a) is the derivative of the arctan function as follows:

$$\sigma(x) = \frac{1}{\pi}\arctan(\frac{\pi}{2}\alpha x) + \frac{1}{2}\,, \tag{12}$$

$$\sigma'(x) = \frac{\alpha}{2(1 + (\frac{\pi}{2}\alpha x)^2)}\,. \tag{13}$$

Equation (13) is the derivative of Equation (12). In addition, SG could be the derivative of Sigmoid (Figure 1B) (Zhou et al., 2023a,b; Zhou Z. et al., 2023), tanh (Guo et al., 2022a), or rectangular (Wu et al., 2018, 2019) functions, etc. To address the problem of gradient vanishing caused by a surrogate gradient function with fixed parameters, Lian et al. (2023) proposed the Learnable Surrogate Gradient (LSG), in which a learnable parameter is used to adjust the gradient-available interval.

Li et al. (2021) proposed Differentiable Spike (Dspike) as another approach to overcome the non-differentiable problem of the Heaviside function. Based on the hyperbolic tangent function, Dspike can be described as Equation (14):

$$\text{Dspike}(x, b) = \frac{\tanh(b(x - 0.5)) + \tanh(b/2)}{2(\tanh(b/2))}, \text{if } 0 \le x \le 1 \tag{14}$$

By adjusting the parameter b, different backpropagation gradients can be obtained. Differentiation on Spike Representation (DSR) proposed by Meng et al. (2022) encodes spike trains and represents them as sub-differentiable mapping, which also avoids the non-differentiable problem during backpropagation.

3.2.2 Loss function and backpropagation

Loss function is the key to neural network training, and different loss functions have been proposed to enhance the performance of SNNs. IM-Loss (Guo et al., 2022a), for example, aims to maximize the information flow in the network. The total loss function consists of two parts, cross-entropy loss, and IM-Loss, as Equations (15, 16):

$$\mathcal{L}_{Total} = \mathcal{L}_{CE} + \lambda\mathcal{L}_{IM}, \tag{15}$$

$$\mathcal{L}_{IM} = \sum_{l=0}^{L}(\bar{U}_l - V_{th})^2/L, \tag{16}$$

where $\bar{U}_l$ is the averaged membrane potential at all time steps of the l-th layer, and L is the total number of layers. To alleviate the information loss in SNNs and reduce the quantization error, RMP-Loss (Guo et al., 2023a) is proposed to adjust the distribution of membrane potential. RecDis-SNN (Guo et al., 2022b) adopts MDP-Loss that also adjusts the membrane potential distribution to overcome the distribution shift during network training. In addition, to improve the generalization ability of SNNs, Deng et al. proposed temporal efficient training (TET) (Deng et al., 2022) loss function to make the network output closer to the target distribution.

Distinct from ANNs, there's an additional dimension in SNNs, the temporal domain. For spiking neurons, the membrane potential in the current step depends on the membrane potential in the previous time step, that is, there is a time dependence. Thus, backpropagation in ANNs does not apply to SNNs. Backpropagation Through Time (BPTT) (Werbos, 1990; Bird and Polivoda, 2021), originally developed for recurrent neural networks (RNNs), is applied to SNNs due to their similar characteristics to those of RNNs. The combination of BPTT and surrogate gradient is the basic approach in SNNs. Spatio-temporal backpropagation (STBP) (Wu et al., 2018), proposed by Wu et al., takes the gradient update in both the spatial domain and temporal domain into account to train SNNs. However, the additional time dimension exposes BPTT and STBP to the problem of requiring a large amount of training memory and training time. Therefore, Xiao et al. (2022) proposed an online training through time (OTTT) algorithm derived from BPTT, which only requires constant training memory consumption agnostic to time steps, and reduces the significant memory costs compared to BPTT. The backward of BPTT and OTTT are shown in Figure 2. Another efficient backpropagation method, Spatial Learning Through Time (SLTT) (Meng et al., 2023), ignores the unimportant routes in the computational graph

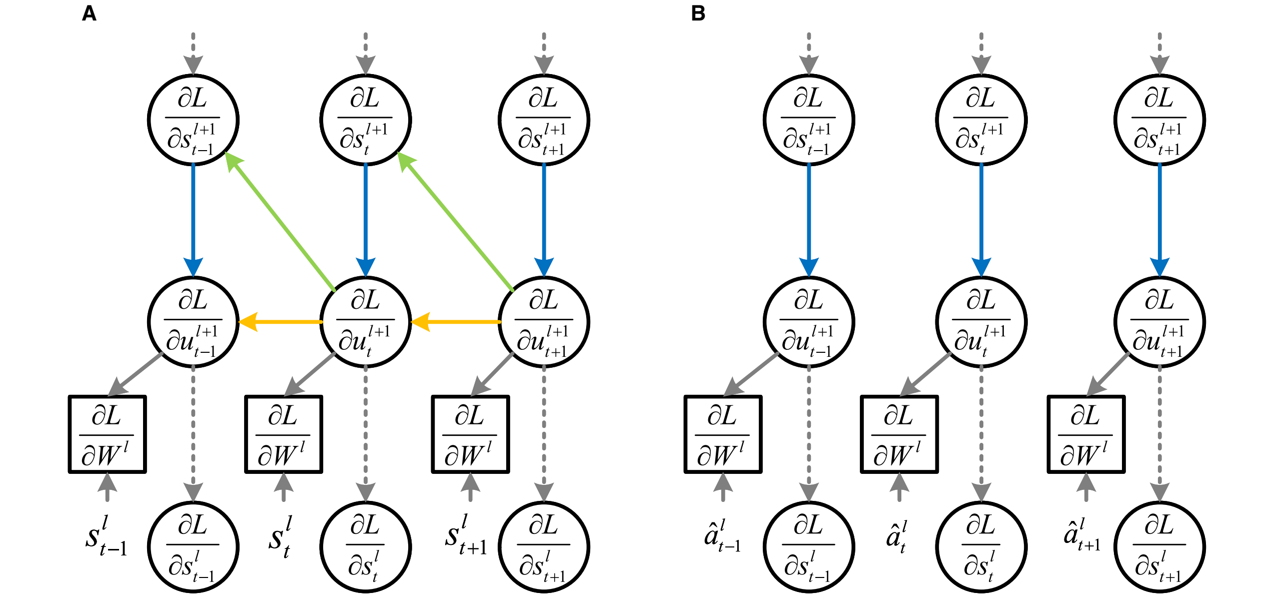
A
B
$\frac{\partial L}{\partial s^{l+1}_{t-1}}$
$\frac{\partial L}{\partial s^{l+1}_{t}}$
$\frac{\partial L}{\partial s^{l+1}_{t+1}}$
$\frac{\partial L}{\partial u^{l+1}_{t-1}}$
$\frac{\partial L}{\partial u^{l+1}_{t}}$
$\frac{\partial L}{\partial u^{l+1}_{t+1}}$
$\frac{\partial L}{\partial W^{l}}$
s^l_{t-1}
s^l_{t}
s^l_{t+1}
$\frac{\partial L}{\partial s^{l}_{t-1}}$
$\frac{\partial L}{\partial s^{l}_{t}}$
$\frac{\partial L}{\partial s^{l}_{t+1}}$
$\hat{a}^l_{t-1}$
$\hat{a}^l_{t}$
$\hat{a}^l_{t+1}$

FIGURE 2
The backward of **(A)** BPTT and **(B)** OTTT.

during backpropagation, to reduce training memory consumption and training time. However, although OTTT and SLTT show better training memory consumption than BPTT, direct training high-performance SNNs are still dominated by the combination of BPTT and surrogate gradient, such as SGLFormer (Zhang et al., 2024), Spikformer (Zhou Z. et al., 2023), etc. Thus, it's essential to investigate direct training methods offering both high effectiveness and efficiency.

3.2.3 Batch normalization

In SNNs, batch normalization is an indispensable component, especially in the context that deep SNNs are difficult to train and converge, compared to ANNs. To mitigate the degradation problems of SNNs, Zheng et al. (2021) proposed threshold-dependent batch normalization (tdBN), which is described as Equation (17):

$$\hat{X}_k = \gamma_k \frac{\alpha V_{th}(X_k - \mu)}{\sqrt{\sigma^2 + \epsilon}} + \beta_k, \tag{17}$$

where α is a hyperparameter, V_{th} is the firing threshold of the neuron, X_k is the feature of the k-th channel, γ_k and β_k are trainable parameters, μ and σ^2 are mean and variance, respectively, ϵ is a tiny constant. Temporal effective batch normalization (TEBN) (Duan et al., 2022) regularizes the temporal distribution, by adopting batch normalization with different parameters at different time steps. Batch normalization through time (BNTT) proposed by Kim and Panda (2021) is similar to TEBN, which also adopts different batch normalization parameters for feature maps at different time steps. Moreover, Guo et al. (2023b) applied batch normalization inside the LIF neuron to normalize the distribution of membrane potentials before firing spikes.

4 SNN architecture developments

This review focuses on the most recent SNN models. Recently, the evolution of residual blocks enhances both the size and performance of deep SNNs significantly. In addition, combining SNNs with transformer architecture has broken the bottleneck of SNNs' performance. Therefore, this review focuses on the application of two kinds of architectures in direct training deep SNNs: transformer structures (Section 4.1) and the residual connections (Section 4.2). Table 2 summarizes their performance on mainstream datasets (ImageNet-1K, CIFAR10, CIFAR100, DVS128 Gesture, CIFAR10-DVS).

4.1 Transformer-based spiking neural networks

Transformer, originally designed for natural language processing (Vaswani et al., 2017), has achieved great success in many computer vision tasks, including image classification (Dosovitskiy et al., 2021; Yuan et al., 2021), object detection (Carion et al., 2020; Liu et al., 2021; Zhu X. et al., 2021), and semantic segmentation (Wang et al., 2021; Yuan et al., 2022). While convolution-based models mainly rely on inductive bias and focus on adjacent pixels, transformer structures use self-attention to capture the relation among spiking features globally, which enhances the performance effectively.

To adopt transformer structure in SNNs, Zhou Z. et al. (2023) designed a novel spike-form self-attention named Spiking Self Attention (SSA), using sparse spike-form Query, Key and Value without softmax operation. The calculation process of SSA is formulated as Equations (18–20):

$$Q = \mathrm{SN}_Q\left(\mathrm{BN}\left(XW_Q\right)\right), K = \mathrm{SN}_K\left(\mathrm{BN}\left(XW_K\right)\right),$$
$$V = \mathrm{SN}_V\left(\mathrm{BN}\left(XW_V\right)\right), \tag{18}$$

$$\mathrm{SSA}'(Q, K, V) = \mathrm{SN}\left(QK^{\mathrm{T}}V * s\right), \tag{19}$$

$$\mathrm{SSA}(Q, K, V) = \mathrm{SN}\left(\mathrm{BN}\left(\mathrm{Linear}\left(\mathrm{SSA}'(Q, K, V)\right)\right)\right), \tag{20}$$

where $Q, K, V \in \mathbb{R}^{T\times N\times D}$. The spike-form Query ($Q$), Key ($K$), and Value ($V$) are computed by learnable layers. s is a scaling factor, which can be fused into the next spiking neuron in practice. Therefore, the calculation of SSA avoids multiplication, meeting the property of SNNs. Based on the SSA, Zhou Z. et al. (2023) developed a spiking transformer named Spikformer, which is shown in Figure 3. As the first transformer-based SNN model, Spikformer achieves 74% accuracy on ImageNet-1k, showing great performance potential.

Zhou et al. (2023a) discussed the non-spike computation problem (integer-float multiplications) of Spikformer (Zhou Z. et al., 2023) and SEW-ResNet (Fang et al., 2021a), which is caused by Activation-after-addition shortcut. Spikingformer (Zhou et al., 2023a) was proposed with the Pre-activation shortcut to avoid the non-spike computation problem in synaptic computing. Experimental Analysis has shown that Spikingformer has only about 43% energy consumption compared with Spikformer in synaptic computing, with only accumulation operations and lower fire rates. CML (Zhou et al., 2023b) designed a downsampling structure specifically for SNNs to solve the imprecise gradient backpropagation problem of most state-of-the-art deep SNNs (including Spikformer). CML achieved 77.34% on ImageNet, significantly enhancing the performance of transformer-based SNNs. All the architectures above are based on SSA with computational complexity of $O(N^2d)$ or $O(Nd^2)$, while Yao et al. (2023a) designed a novel Spike-Driven Self-Attention (SDSA) with linear complexity regarding both the number of tokens and channels. SDSA uses only mask and addition operations without any multiplication, thus having up to 87.2× lower computation energy than the vanilla SSA. In addition, the Spike-driven Transformer based on SDSA has achieved 77.1% accuracy on ImageNet-1k. Wang Y. et al. (2023) proposed an SNN-based spatial-temporal self-attention (STSA) mechanism, which could calculate the feature dependence across the time and space domains. Shi et al. (2024) proposed Dual Spike Self-Attention (DSSA) with a reasonable scaling method, achieving 79.40% top-1 accuracy on ImageNet-1K. Yao et al. (2024) proposed Spike-driven Transformer v2 which explored the impact of structure, spike-driven self-attention, and skip connection on its performance to inspire the next-generation transformer-based neuromorphic chip designs. Zhou Z. et al. (2024) developed a Spiking Convolutional Stem (SCS) with supplementary layers to enhance the architecture of Spikformer, achieving 80.38% accuracy on ImageNet-1k. Zhang et al. (2024) proposed a Spiking Global-Local-Fusion Transformer (SGLFormer), which enables efficient information processing on both global and local scales, by integrating transformer and convolution structures in SNNs. SGLFormer achieved a groundbreaking top-1 accuracy of 83.73% on ImageNet-1k with 64M parameters. Zhou C. et al. (2024) proposed QKFormer, a novel hierarchical spiking transformer using Q-K attention, which can easily model the importance of token or channel dimensions with binary values and has linear complexity to #tokens (or #channels). QKFormer achieved a significant milestone, surpassing 85% top-1 accuracy on ImageNet with 4 time steps using the direct training approach.

Biological realistic models tend to model neural networks with high biological plausibility to simulate the complex biological mechanism of the brain. It often lacks the consideration of computational efficiency and performance optimization on general application tasks. Traditional ANNs often prioritize task performance over biological realism and computational energy consumption. SNNs have great potential to own the characteristics of biological plausibility, low computational energy consumption, and high task performance simultaneously. Especially, several direct training Transformer-based SNNs have broken through 80% top-1 accuracy on ImageNet-1K, which instills great optimism in the application of SNNs.

4.2 Residual architectures in spiking neural networks

Residual block is the fundamental block in both deep ANNs and SNNs. As shown in Figure 4, there are mainly three residual shortcut types in SNNs: Activation-after-addition, Activation-before-addition, and Pre-activation. Both advantages and disadvantages of these three types are concluded in Table 3. **Activation-after-addition shortcut** simply replaces ReLU activation layers in the standard residual block with spiking neurons, such as Spiking ResNet (Hu et al., 2021a) and MPBN (Guo et al., 2023b). SNNs with this simple design suffer from performance degradation and gradient vanishing/exploding. For example, the deeper 34-layer Spiking ResNet has lower test accuracy than the shallower 18-layer Spiking ResNet. As the layer increases, the test accuracy of Spiking ResNet decreases (Fang et al., 2021a). To solve the degradation problem in the Activation-after-addition shortcut, **Activation-before-addition shortcut** is proposed in SEW-ResNet (Fang et al., 2021a), which extended directly trained SNNs to 100 layers for the first time. This structure has been widely used, such as in Spikformer (Zhou Z. et al., 2023), PLIF (Fang et al., 2021b), PSN (Fang et al., 2023b). This design mitigates the vanishing/exploding gradient problem and could train deeper SNN. However, the blocks in this shortcut will result in positive integers, which leads to non-spike computations (integer-float multiplications) in synaptic computing (like convolutional layer, linear layer) (Zhou et al., 2023a). **Pre-activation shortcut** could be traced back to the *Activation-Conv-Bn* paradigm, which is a fundamental building block in Binary Neural Networks (BNNs) (Liu et al., 2018, 2020; Guo N. et al., 2021; Zhang Y. et al., 2022). Some representative SNNs that use the Pre-activation shortcut include MS-ResNet (Hu et al., 2021b), Spikingformer (Zhou et al., 2023a), Spike-driven transformer (Yao et al., 2023a). MS-ResNet directly trained convolution-based SNNs to successfully extend

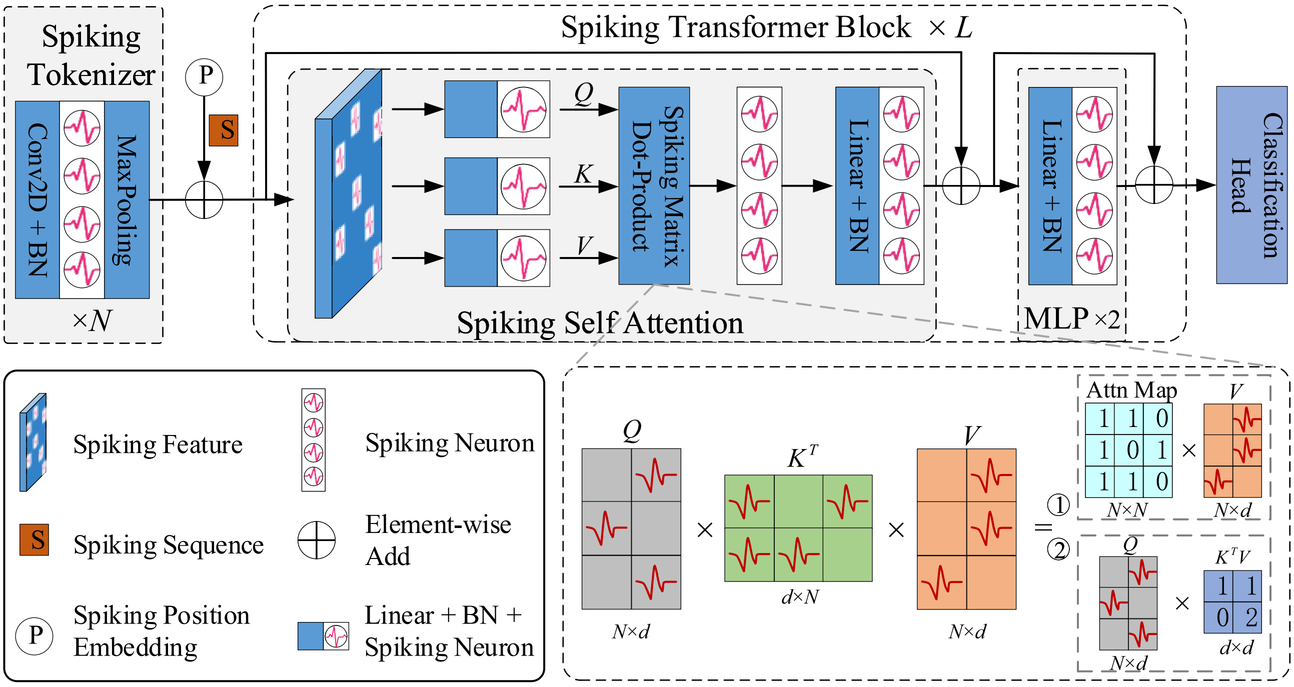

FIGURE 3
The overview of spiking transformer (Spikformer).

TABLE 3 Features of various residual learning architectures.

Features	Activation-after-addition	Activation-before-addition	Pre-activation
Element addition	Spike added to floats	Integer added to spike	Floats added to floats
Gradient vanishing/exploding	Yes	No	No
Synaptic computing type	Spike computing	Multiplication between sparse integer and floats	Spike computing
Data transmission in residual branch	Spike	Sparse integer	Floats

the depth up to 482 layers on CIFAR10 without experiencing degradation problems, effectively verifying the feasibility of this way. Spikingformer (Zhou et al., 2023a) showed that the Pre-activation shortcut can effectively avoid non-spike computations, and thus has lower energy consumption than the previous shortcut in synaptic computing, through avoiding integer-float multiplication problems and with a lower firing rate. However, the Pre-activation shortcut requires dense transmission of floats in the residual branch.

Overall, the residual learning suitable for the properties of SNNs needs further exploration. In our opinion, the Activation-after-addition shortcut with gradient problem is not suitable for directly training deep SNNs, but is feasible in the field of ANN-to-SNN conversion. Activation-before-addition shortcut has some alternatives to ensure the properties of SNNs by slightly sacrificing the performance, such as using AND or IAND to replace ADD in the aggregation operation. Pre-activation shortcut needs further analyses of the effects of float transmission, and more efforts to exploit its advantages through collaborative hardware optimization and design.

4.3 Others

Besides the above-mentioned architectures, some other interesting research topics are also worthy of attention, such as Spiking RNN/LSTM, LSM, etc. Deep Liquid State Machine (LSM) (Wang and Li, 2016) explored the power of recurrent spiking networks and deep architectures. Soures and Kudithipudi (2019) proposed a novel deep LSM to capture dynamic information over multiple time-scales with a combination of randomly connected layers and unsupervised layers. Hamilton et al. (2019) demonstrated the nonlinear dynamics of spiking neurons can be used to implement low-level graph operations. Zhu Z. et al. (2022) proposed end-to-end Spiking Graph Convolutional Networks (GCNs) that integrate the embedding of GCNs with the biofidelity characteristics of SNNs. Bellec et al. (2020) and Bohnstingl et al. (2022) explored the architectures and online-training methods of recurrent spiking neural networks. Ren H. et al. (2023) proposed a novel end-to-end point-based SNN architecture, which excels at processing sparse event cloud data, effectively extracting both global and local features through a singular-stage structure.

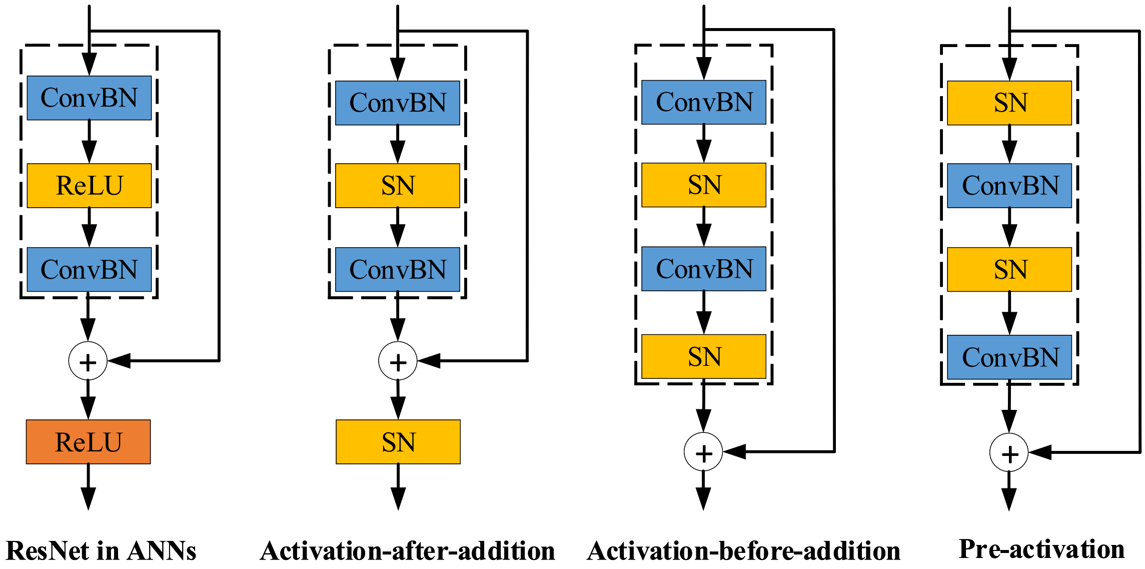

FIGURE 4
The overview of residual learning architectures.

5 Software frameworks and neuromorphic hardware for spiking neural networks

5.1 Software frameworks for training spiking neural networks

Software frameworks play a crucial role in propelling the advancement of deep learning. Deep learning frameworks such as PyTorch (Paszke et al., 2019) and TensorFlow (Abadi et al., 2016) leverage low-level languages like C++ libraries for high-performance acceleration on the backend, while offering user-friendly front-end application programming interfaces (APIs) implemented in high-level languages like Python. These frameworks significantly ease the workload of constructing and training ANNs, making substantial contributions to the growth of deep learning research. However, these deep learning frameworks are primarily designed for ANNs. With the development of large-scale brain-inspired neural networks, many related frameworks have emerged, facilitating the modeling and efficient computation of large-scale SNNs.

One category of frameworks includes brain simulators such as NEURON (Hines and Carnevale, 1997) and Brian (Goodman and Brette, 2009), which not only enhance the scalability and computational efficiency of models but also encompass cognitive functions such as perception, decision-making, and reasoning. The SNNs constructed by these frameworks exhibit a high degree of biological plausibility, making them suitable for studying the functionalities of real neural systems. They support biologically interpretable learning rules such as Spike-Timing-Dependent Plasticity (STDP) (Bi and Poo, 1998), playing a significant role in advancing the field of neuroscience. However, these frameworks lack core computational functionalities required for deep learning, such as automatic differentiation, rendering them incapable of performing machine learning tasks.

Another category of brain-inspired computing frameworks comprises deep spiking computation frameworks. Deep SNNs involve a substantial amount of matrix operations across spatial and temporal dimensions, a variety of neurons, neuromorphic datasets, and deployments on neuromorphic chips. The modeling and application processes are complex, and achieving high-performance acceleration is challenging. To address these issues, spiking deep learning frameworks need to support the construction, training, and deployment of deep SNNs, and be capable of acceleration based on spike operations. Frameworks such as BindsNET (Hazan et al., 2018), NengoDL (Rasmussen, 2019), SpykeTorch (Mozafari et al., 2019), Norse, SpyTorch, SNNTorch (Eshraghian et al., 2023), and SpikingJelly (Fang et al., 2023a) have been developed. They utilize simple spiking neurons to reduce computational complexity, making them suitable for machine learning research. Among them, BindsNet (Hazan et al., 2018) primarily focuses on machine learning and reinforcement learning; NengoDL (Rasmussen, 2019) converts ANNs to obtain deep SNNs but does not support direct training of SNNs using surrogate gradient methods; SpyTorch is a demonstrative framework that only provides basic surrogate gradient examples; SpyTorch (Mozafari et al., 2019) introduces a new type of surrogate gradient method named SuperSpike. These frameworks can implement some simple machine learning and reinforcement learning models, but they still lack deep learning capabilities for SNNs. Norse is attempting to introduce the sparse and event-driven characteristics of SNNs and supports many typical spiking neuron models. It is in the development stage and has not been officially released yet. SNNTorch supports some variants of online backpropagation algorithms that are more biologically plausible and support large-scale SNN computation. SpikingJelly (Fang et al., 2023a) is a full-stack toolkit for preprocessing neuromorphic datasets, building deep SNNs, optimizing their parameters, and deploying SNNs on neuromorphic chips, which shows remarkable extensibility and flexibility, enabling users to accelerate custom models at low costs through multilevel inheritance and semiautomatic code generation. In summary, the development of existing software frameworks is essentially in its early stages, and there is still a long way to go in terms of functionality enhancement and performance optimization.

5.2 Neuromorphic hardware for spiking neural networks

Neuromorphic hardware provides computational power for neural network models, playing a crucial role in large-scale brain-like neural networks. Efficient hardware can significantly accelerate the training, evaluation, iteration, and real-world applications of large-scale brain-like models. In comparison to general-purpose processors, deep learning chips and brain-like chips are specialized chips that focus on the computational efficiency of deep learning tasks and brain-like computing tasks, aiming to achieve better power/performance/area ratios. Current deep learning chips, like general CPUs, are based on the Von Neumann architecture, with separate computing and storage units. Brain-like chips enhance computational efficiency by designing efficient storage and computation hierarchy, enabling parallel data flow and efficient reuse, thus improving computational efficiency. From an architectural perspective, current brain-like chips can be mainly divided into two categories: analog-digital hybrid circuits and fully digital circuits (Table 4).

Inspired by the simultaneous computation and storage capabilities of the brain's neural system, brain-inspired chips often adopt near-memory (NMA) or compute-in-memory architectures (CIM), incorporating closely coupled computational and storage resources within each computing core (Akopyan et al., 2015; Pei et al., 2019). Efficient intra-chip and inter-chip interconnects enable large-scale computational parallelism and high local memory, reducing computational power consumption.

The near-memory computing architecture refers to the separation of memory storage and computation in each processing unit, but with proximity. Key chips in this category include IBM's TrueNorth (Akopyan et al., 2015), Intel's Loihi (Davies et al., 2018; Davies, 2021), the University of Manchester's SpiNNaker (Painkras et al., 2013), Stanford University's Neurogrid (Benjamin et al., 2014), Heidelberg University's BrainScaleS (Schemmel et al., 2010; Pehle et al., 2022), Tsinghua University's Tianji Chip (Pei et al., 2019), and Zhejiang University's Darwin Chip (Shen et al., 2016; Ma et al., 2017, 2024). They utilize characteristics of brain-like spiking computation such as sparsity, spike summation, and asynchronous event-driven processing to achieve ultra-low power consumption, currently mainly supporting model inference and local online learning based on STDP, Hebb, etc. For instance, ROLLS (Qiao et al., 2015), ODIN (Frenkel et al., 2018), and MorphIC (Frenkel et al., 2019) support spike-driven synaptic plasticity (SDSP) rules, and Loihi adds a learning module for STDP rules. In SpiNNaker (Painkras et al., 2013) and BrainScaleS (Schemmel et al., 2010), STDP learning is exhibited through timestamp recording and learning circuits. In their next generations (Pehle et al., 2022), more flexible learning rules are possible due to the presence of embedded programmable units. Tsinghua University's Tianji Chip, as the first chip to support the fusion of SNN and ANN computation, improves accuracy based on ANN, and achieves rich dynamics, high efficiency, and robustness based on SNN. This mode is also adopted by BrainScaleS-2 (Pehle et al., 2022), SpiNNaker-2, and Loihi-2 (Davies, 2021). Recently, BPTT has been applied to SNNs, achieving higher accuracy compared to local learning rules (Wu et al., 2018, 2019). Some works, like H2Learn (Liang et al., 2021) and SATA (Yin et al., 2022), have designed specific architectures for BPTT learning in SNNs. In the future, the integration of learning rules will become increasingly important for exploring large and complex neuromorphic models in brain-inspired computing (BIC) chips.

Another important type of BIC architecture is the compute-in-memory architecture, where in-core processing units and on-chip storage are physically integrated, performing synaptic integration matrix operations in synaptic memory. Compute-in-memory chips can be divided into two categories based on the materials: traditional or emerging memories. Traditional memories (such as SRAM, DRAM, and Flash) can be redesigned to support specific logical operations (Wu et al., 2020). Their advantages include a mature ecosystem, easy simulation, and manufacturing. Emerging memories mainly refer to storage devices based on memristors. Synaptic weight storage, multiplication calculations, and presynaptic inputs are performed at the same crosspoint in the memristor, integrating computation and storage. Zhang et al. (2021) designed a hybrid spiking neuron combining a memristor with simple digital circuits to enhance neuron functions. Further, they demonstrated a full hardware spiking neural network with the hybrid neurons and memristive synapses for the first time, and achieved *in-situ* Hebbian learning. Brain-inspired computing hardware based on memristors involves multiple levels of material and architectural designs, which is currently still in a small-scale phase due to manufacturing process limitations.

Multiple types of brain-like chips have shown remarkable developments, demonstrating significant advantages in terms of biological simulation and low-power inference. However, they still face numerous challenges in practical applications. When it comes to handling high-level intelligence tasks, the superiority of brain-like chips compared to GPUs and ANN accelerators has not been fully established. Currently, to optimize their performance, some designs draw inspiration from ANN accelerators for improvements. It's worth noting that current brain-like chips do not yet support the training of large-scale SNNs and require special architectural designs to accommodate the training process for SNNs. To further support large-scale SNNs, it is necessary to enhance brain-like systems from both a software and hardware perspective in a more collaborative manner.

5.3 Software and hardware interplay

The deployment of algorithms for SNNs onto neuromorphic chips typically requires certain software frameworks. The computational software frameworks mentioned in Section 5.1 usually support simulations on mainstream CPUs and GPUs, without clear mention of deployment on neuromorphic chips.

TABLE 4 Overview of typical neuromorphic hardware.

Chip	Developer	Network	Function	Arch	Scale
Hybrid digital-analog					
BrainScaleS (Schemmel et al., 2010)	Heidelberg Uni	SNNs	Training	NMA	Large
Neurogrid (Benjamin et al., 2014)	Stanford	SNNs	Inference	NMA	Large
ROLLS (Qiao et al., 2015)	UZH	SNNs	Training	NMA	Small
DYNAPs (Moradi et al., 2017)	UZH	SNNs	Inference	NMA	Small
Memristor-based (Zhang et al., 2021)	–	ANNs/SNNs	Inference	CIM	Small
BrainScaleS-2 (Pehle et al., 2022)	Heidelberg Uni	ANNs/SNNs	Training	NMA	Large
Digital					
SpiNNaker (Painkras et al., 2013)	UoM	SNNs	Training	NMA	Large
SpiNNaker 2	UoM	ANNs/SNNs	Training	NMA	Large
TrueNorth (Akopyan et al., 2015)	IBM	SNNs	Inference	NMA	Large
Darwin (Shen et al., 2016)	ZJU	SNNs	Inference	NMA	Small
Darwin II (Ma et al., 2017)	ZJU	SNNs	Training	NMA	Large
Darwin III (Ma et al., 2024)	ZJU	SNNs	Training	NMA	Large
DeepSouth (Wang et al., 2017)	Westwell	SNNs	Inference	NMA	Large
Intel SNN chip (Chen et al., 2018)	Intel	SNNs	Training	NMA	Large
ODIN (Frenkel et al., 2018)	K.U.Leuven	SNNs	Training	NMA	Small
Loihi (Davies et al., 2018)	Intel	SNNs	Training	NMA	Large
Tianjic (Pei et al., 2019)	Tsinghua	ANNs/SNNs	Training	NMA	Large
MorphIC (Frenkel et al., 2019)	UZH	SNNs	Training	NMA	Small
Flash-based (Wu et al., 2020)	–	ANNs/SNNs	Inference	CIM	Small
Loihi II (Davies, 2021)	Intel	SNNs	Training	NMA	Large
Y. Kuang et al. (Kuang et al., 2021)	PKU	ANNs/SNNs	Inference	NMA	Large
H2Learn (Liang et al., 2021)	UCSB	SNNs	Training	SNN TA	Large
SATA (Yin et al., 2022)	Yale	SNNs	Training	SNN TA	Small

SNN TA, SNN training accelerator; Arch, architecture.

Meanwhile, among the previously mentioned neuromorphic chips in Section 5.2, only about 27% of them are connected to application software packages (Schuman et al., 2022). Typically, these application software packages contain model construction tools, simulators, and optimization tools. Model construction tools are used to define the structure and parameters of neural networks, including neuron types, connection patterns. Simulators are applied to simulate and debug neural network models on the chip. Optimization tools are adopted to train the network parameters and optimize its performance. Here are some typical examples. The Neurogrid chip is paired with the Neurogrid Software Framework (Benjamin et al., 2014), allowing users to specify neuronal models in the Python programming environment. The software framework for the BrainScaleS chip is the BrainScaleS-Software-Stack (Pehle et al., 2022), which supports training neural networks on the chip using the PyTorch framework. The IBM TrueNorth chip typically utilizes a software framework called the TrueNorth Ecosystem, which is developed by the TrueNorth native Corelet language (Akopyan et al., 2015). IBM NorthPole (Modha et al., 2023) is a brain-inspired memory-near-compute chip with a software development kit, but this chip does not emulate spiking communication. Tianjic's software toolchain supports both ANN-to-SNN conversion and direct training for SNNs, and supports automatically transforming a pretrained model into an equivalent network that meets the Tianjic hardware constraints for non-spiking ANNs (Pei et al., 2019). The latest Darwin3 builds a specialized instruction set architecture (ISA) (Ma et al., 2024), which is close to machine code tailored for efficient neuromorphic computing. These software frameworks enable users to conveniently construct, simulate, and optimize neural network models on neuromorphic chips, facilitating efficient research and application development.

6 Applications of deep spiking neural networks

SNNs offer powerful computation capability due to their event-driven nature and temporal processing property. Theoretically, SNNs could be applied to any field where conventional deep neural networks (DNNs) are applied. As the training methods and programming frameworks of deep SNNs become more powerful, deep SNNs are increasingly drawing more attention and being applied to more fields, mainly including computer vision, reinforcement learning and autonomous robotics, biological visual system modeling, biological signal processing, natural language processing, equipment safety monitoring, and so on. It should be noted that this paper only lists some typical examples in recent years for some common application fields, not aiming to fully review all related studies.

6.1 Applications in computer vision

As traditional DNNs, the most common applications of SNNs lay in computer vision tasks. There are mainly two types of visual inputs for SNNs, i.e., RGB frames from traditional cameras or events from neuromorphic vision sensors. Neuromorphic vision sensors display great potential for computer vision tasks under high-speed and low-light conditions (Li and Tian, 2021). SNNs are excellent candidates for processing neuromorphic signals due to their event-driven nature and energy-efficient computing.

Recognition task plays an important role in the rapid progress of deep SNNs. SNNs are usually tested on both static datasets such as CIFAR10, CIFAR100, ImageNet, and neuromorphic datasets such as CIFAR10-DVS and DVS128-Gesture. Table 2 lists the performances of some recently proposed architectures. Besides common recognition tasks, deep SNNs are increasingly applied to more computer vision tasks, including object detection/tracking, image denoising/generation, image/video reconstruction, video action recognition, image segmentation, and so on.

6.1.1 Object detection and object tracking

The first spike-based object detection model Spiking-YOLO was obtained through the ANN-to-SNN conversion method, achieving comparable performances to tiny-YOLO on PASCAL VOC and MS-COCO dataset with 3,500 time steps (Kim et al., 2020). Later, a spike calibration (SpiCalib) method was proposed to reduce the time steps to hundreds (Li et al., 2022). Kugele et al. (2021) and Cordone et al. (2022) combined some spiking backbones with an SSD detection head for event cameras. Su et al. (2023) proposed EMS-YOLO, the first directly trained deep SNNs for object detection, achieving comparable performance to the ANN counterpart while consuming less energy on both the frame-based MS-COCO dataset and the event-based Gen1 dataset in only four time steps.

Considering that the Siamese networks have achieved remarkable performances in object tracking, SiamSNN was constructed by conversion to achieve short latency and low precision degradation on several benchmarks (Luo et al., 2022). Similarly, the directly trained Spiking SiamFC++ showed a small precision loss compared to the original SiamFC++ (Xiang et al., 2022). A spiking transformer network called STNet was developed for event-based single-object tracking, demonstrating competitive tracking accuracy and speed on three event-based datasets (Zhang et al., 2022a). To process frames and events simultaneously, Yang et al. (2019) proposed DashNet, achieving good tracking performance with a surprising tracking speed of 2,083 FPS on neuromorphic chips.

6.1.2 Image generation/denoising and image/video reconstruction

Generation tasks are increasingly explored in SNNs. Comşa et al. (2021) introduced a directly trained spiking autoencoder to reconstruct images with high fidelity on MNIST and FMNIST datasets. Kamata et al. (2022) constructed a fully spiking variational autoencoder (FSVAE), generating images with competitive quality compared to conventional ANNs. Liu et al. (2023) proposed a Spiking-Diffusion model, outperforming the existing SNN-based generation models on several datasets. Castagnetti et al. (2023) developed an image denoising solution based on a directly trained spiking autoencoder, achieving a competitive signal-to-noise ratio on the Set12 dataset with significantly lower energy.

Visual information reconstruction is important for neuromorphic vision sensors, because humans cannot directly perceive visual scenes from events. Zhu and Tian (2023) provided a comprehensive review of visual reconstruction methods for events. Duwek et al. (2021) proposed a hybrid ANN-SNN model, accomplishing image reconstruction for simple scenes from N-MNIST and N-Caltech101 datasets. Zhu L. et al. (2021) proposed an image reconstruction algorithm that combines DVS and Vidar signals, leveraging the high dynamic range of DVS to improve reconstruction effectiveness. Subsequently, they developed a deep SNN with an encoder-decoder structure for event-based video reconstruction, achieving performance comparable to ANN counterparts with only 0.05x energy consumption (Zhu L. et al., 2022).

6.1.3 Others

Besides the above-mentioned tasks, deep SNNs are also applied in some other computer vision tasks, including video action recognition (Panda and Srinivasa, 2018; Wang et al., 2019; Zhang et al., 2022c; Chakraborty and Mukhopadhyay, 2023; Yu et al., 2024), image segmentation (Parameshwara et al., 2021; Kim et al., 2022; Liang et al., 2022; Zhang H. et al., 2023), optical flow estimation (Lee et al., 2020a; Cuadrado et al., 2023; Kosta and Roy, 2023), depth prediction (Rançon et al., 2022; Wu et al., 2022; Zhang et al., 2022b), point clouds processing (Zhou et al., 2020; Ren D. et al., 2023), human pose tracking (Zou et al., 2023), lip-reading (Bulzomi et al., 2023), emotion/expression recognition (Wang B. et al., 2022; Barchid et al., 2023), medical image classification (Shan et al., 2022; Qasim Gilani et al., 2023), and so on.

6.2 Applications in other fields

Besides computer vision tasks, SNNs are showing gradually expanding application prospects in many fields, including reinforcement learning and autonomous robotics, biological visual system modeling, biological signal processing, natural language processing, equipment safety monitoring, and so on.

6.2.1 Reinforcement learning and autonomous robotics

As reinforcement learning (RL) is critical for the survival of humans and animals, there is increasing interest in applying brain-inspired SNNs to reinforcement learning. Asgari et al. (2020) implemented an SNN with RL capability on hardware to learn stimulus-response associations, performing well in controlling a robot through a closed sensory-motor loop. Chen et al. (2022) constructed the deep spiking Q-network (DSQN) to directly learn robust policies from high-dimensional sensory inputs using end-to-end deep RL, which was tested on 17 Atari games, outperforming the ANN-based deep Q-network (DQN) in most games. To reduce the latency of spiking RL, Qin et al. (2023) applied learnable matrix multiplication to encode and decode spikes.

Due to the good biological plausibility and high energy efficiency, SNNs have been applied to autonomous robotics for a long time, which is still a flourishing research direction, mainly including pattern generation (walk, trot, or run), motor control, and navigation (simultaneous localization and mapping, SLAM). Yamazaki et al. (2022) have already provided a good review of relevant studies, we do not go into more detail about this topic in this review which mainly focuses on deep SNNs.

6.2.2 Biological visual system modeling and biological signal processing

ANNs play important roles in modeling biological visual pathways. However, SNNs are more biologically plausible models due to the use of temporal spike sequences. Therefore, several studies adopted SNNs to model the biological visual cortex. Huang et al. (2023a) applied deep SNNs to model the visual cortex for the first time, of which the similarity scores are higher than their ANN counterparts, showing SNNs' effectiveness. Further, they added a brain-inspired recurrent module into deep SNNs, outperforming the forward deep SNNs under natural movie stimuli (Huang et al., 2023b). Zhang J. et al. (2023) compared performances of deep SNNs and CNNs in the prediction of visual responses to naturalistic stimuli in three brain areas. Ma et al. (2023) presented a temporal conditioning spiking latent variable model to produce more realistic spike activities.

Due to the intrinsic dynamics, SNNs are also applied to process biological signals. Xiong et al. (2021) proposed a convolutional SNN for odor recognition of electronic noses. Feng Y. et al. (2022) applied the conversion method to obtain a 14-layer SNN model for ECG (electrocardiogram) classification. Li X. et al. (2023) reviewed recent studies applying SNNs on signal classification and disease diagnosis based on biological signals including EEG, ECG, EMG, and so on.

6.2.3 Others

SNNs were also applied to natural language processing, equipment safety monitoring, semantic communication, multi-modal information processing, and so on. To ease the heavy energy cost of ANN-based large language models, some studies applied SNN-based architectures, including SpikBERT (Lv et al., 2023), SpikingBERT (Bal and Sengupta, 2024), SpikeGPT (Zhu et al., 2023), and SpikeLM (Xing et al., 2024). Applications regarding equipment safety monitoring mainly include battery health monitoring (Wang et al., 2023a,b), autonomous vehicle sensors fault diagnosis (Wang and Li, 2023), and bearing fault diagnosis (Xu et al., 2022). Applications in semantic communication mainly tried to mitigate the limitation of transmission bandwidth (Wang M. et al., 2023). Applications in multi-modal information processing currently show up in audio-visual zero-shot learning tasks (Li et al., 2023a,b).

6.3 Discussion on SNN applications

Deep SNNs have achieved great success in many fields in recent years, but there still exist some limits that need to be addressed. Firstly, although many studies demonstrated that deep SNNs achieved comparable accuracy to their ANN counterparts on many tasks, they still lag behind conventional ANN SOTA, especially for large datasets like ImageNet, which asks for more endeavors. Secondly, many studies claimed that the proposed SNNs consumed much less energy compared to ANN counterparts, through calculating the number of addition operations, without considering the cost of other operations like data movement. Therefore, it is meaningful to deploy well-performed SNNs on neuromorphic chips or corresponding simulators to fully exploit the event-driven nature and measure the actual energy cost. Thirdly, as for applications requiring high processing speed and low power consumption, like robotics, it is promising to adopt neuromorphic vision/audio sensors and neuromorphic processing chips due to their event-driven nature, besides network pruning and weight quantization. Meanwhile, to fully exploit the advantages of events, it deserves more efforts to explore how to directly process neuromorphic sensing events using SNNs, without converting events into frames as current studies usually do. Fourthly, as for transformer-based SNNs used in language or video processing, how to choose the input clip for one simulation step, to reconcile the temporal resolution of the input sequence and the simulation step of SNNs, is worth studying. Last but not least, as SNNs have an additional temporal dimension, how to achieve the speed-accuracy trade-off as humans is a problem worth of study. In other words, how to assign a suitable simulation duration or how to decide when to make a choice, are important questions to realize the balance between computation cost and prediction accuracy.

7 Future trends and conclusions

This article provides an overview of the current developments in various theories and methods of deep SNNs, including relevant fundamentals, various spiking neuron models, advanced models, and architectures, booming software tools and hardware platforms, as well as applications in various fields. However, there are still many limitations and challenges.

(1) Currently, only a few aspects of the intelligent brains have been applied to instruct the construction and training of SNNs, lacking enough biological plausibility. Therefore, to improve SNNs' capability, it is necessary to introduce more types

of spiking neurons, rich connection structures, multiscale local-global-cooperative learning rules, system homeostasis, etc., into SNNs to more accurately mimic the cognitive and intelligent characteristics emerging in the brains. For example, it deserves more efforts to train SNNs with self-supervised or unsupervised learning (Zhou Z. et al., 2024), as children mainly receive unlabeled data during growth. Besides, the brain is actually a complex network, thus it is worthy of more effort to study graph SNNs, although some attempts already exist (Li et al., 2024; Yin et al., 2024).

(2) Recent neuroscience studies have found that astrocytes can naturally realize Transformer operations (Kozachkov et al., 2023), which provides a new direction for the improvement of SNNs. In addition, astrocytes have the function of regulating neuronal firing activity and synaptic pruning (Lee et al., 2021; Liu et al., 2022), which provides ideas for the performance improvement and lightweight of SNNs in the future.

(3) Information encoding methods and training algorithms for SNNs are mostly based on average firing rates, lacking the ability to represent temporal dynamics adequately. There should be more exploration of time-dependent information encoding strategies and corresponding training algorithms, to further enhance the spatiotemporal dynamic characteristics of SNNs and strengthen their temporal processing capability.

(4) The training of SNNs mainly employs time-dependent methods, like BPTT, which greatly increases the training cost, compared to conventional DNNs. Thus, there is a need to develop brain-like SNNs that can be trained in parallel, and dedicated software and hardware that support their computation, reducing training time and power consumption.

(5) As there are obstacles to conversion and interaction between different neuromorphic platforms, it is needed to establish a common standard to improve interoperability. Further, more brain-inspired principles or technologies should be incorporated into the design of neuromorphic systems, to enhance the computational performance of the chips, in terms of processing speed and energy efficiency.

(6) Large-scale SNNs are mainly applied to classification tasks. Their potential in handling tasks that need to process continuous input streams, such as videos, languages, events from neuromorphic vision sensors, etc., has not been fully explored. Moreover, the introduction of various neuromorphic sensors and neuromorphic chips into autonomous robotics, cooperating with conventional sensors and processing chips, might be an efficient and effective way to achieve embodied intelligence. Further studies are needed to fully leverage the features and advantages of SNNs.

In summary, studies and applications of SNNs are growing rapidly, but there is still great potential to improve the effectiveness and efficiency of SNNs. Efforts should be made in multiple directions, including model architectures, training algorithms, software frameworks, and hardware platforms, to promote the coordinated progress of models, software, and hardware.

Author contributions

CZ: Writing – original draft, Writing – review & editing. HZha: Writing – original draft, Writing – review & editing. LY: Writing – original draft, Writing – review & editing. YY: Writing – original draft, Writing – review & editing. ZZ: Writing – review & editing. LH: Writing – review & editing. ZM: Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing. XF: Writing – review & editing, Resources. HZho: Writing – review & editing, Resources. YT: Writing – review & editing, Resources.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. The study was funded by the National Natural Science Foundation of China under contracts Nos. 62206141, 62236009, 62332002, 62027804, and 61825101, and the major key project of the Peng Cheng Laboratory (PCL2021A13). Computing support was provided by Pengcheng Cloudbrain.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). "Tensorflow: a system for large-scale machine learning," in *Symposium on Operating Systems Design and Implementation (OSDI)* (Savannah, GA), 265–283.

Adrian, E. D., and Zotterman, Y. (1926). The impulses produced by sensory nerve endings: Part 3. Impulses set up by touch and pressure. *J. Physiol.* 61:465. doi: 10.1113/jphysiol.1926.sp002273

Akopyan, F., Sawada, J., Cassidy, A., Alvarez-Icaza, R., Arthur, J., Merolla, P., et al. (2015). Truenorth: design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip. *IEEE Transact. Comp. Aided Des. Integr. Circ. Syst.* 34, 1537–1557. doi: 10.1109/TCAD.2015.2474396

Asgari, H., Maybodi, B. M.-N., Kreiser, R., and Sandamirskaya, Y. (2020). Digital multiplier-less spiking neural network architecture of reinforcement learning in a context-dependent task. *IEEE J. Emerg. Select. Top. Circ. Syst.* 10, 498–511. doi: 10.1109/JETCAS.2020.3031040

Bal, M., and Sengupta, A. (2024). pikingbert: distilling bert to train spiking language models using implicit differentiation. *Proc. AAAI Conf. Artif. Intell.* 38, 10998–11006. doi: 10.1609/aaai.v38i10.28975

Barchid, S., Allaert, B., Aissaoui, A., Mennesson, J., and Djeraba, C. C. (2023). "Spiking-fer: spiking neural network for facial expression recognition with event cameras," in *International Conference on Content-based Multimedia Indexing (CBMI)* (Orleans), 1–7.

Bellec, G., Salaj, D., Subramoney, A., Legenstein, R. A., and Maass, W. (2018). "Long short-term memory and learning-to-learn in networks of spiking neurons," in *NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems, Vol. 31* (Montreal, QC).

Bellec, G., Scherr, F., Subramoney, A., Hajek, E., Salaj, D., Legenstein, R., et al. (2020). A solution to the learning dilemma for recurrent networks of spiking neurons. *Nat. Commun.* 11:3625. doi: 10.1038/s41467-020-17236-y

Benjamin, B. V., Gao, P., McQuinn, E., Choudhary, S., Chandrasekaran, A. R., Bussat, J.-M., et al. (2014). Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations. *Proc. IEEE* 102, 699–716. doi: 10.1109/JPROC.2014.2313565

Bi, G.-Q., and Poo, M.-M. (1998). Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci.* 18, 10464–10472. doi: 10.1523/JNEUROSCI.18-24-10464.1998

Bird, G. M., and Polivoda, M. E. (2021). Backpropagation through time for networks with long-term dependencies. *arXiv [Preprint]*. doi: 10.48550/arXiv.2103.15589

Bohnstingl, T., Šurina, A., Fabre, M., Demirağ, Y., Frenkel, C., Payvand, M., et al. (2022). "Biologically-inspired training of spiking recurrent neural networks with neuromorphic hardware," in *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)* (Incheon: IEEE), 218–221.

Bu, T., Fang, W., Ding, J., Dai, P., Yu, Z., and Huang, T. (2022). "Optimal ann-snn conversion for high-accuracy and ultra-low-latency spiking neural networks," in *International Conference on Learning Representations (ICLR)*.

Bulzomi, H., Schweiker, M., Gruel, A., and Martinet, J. (2023). "End-to-end neuromorphic lip-reading," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Vancouver, BC), 4100–4107.

Cao, Y., Chen, Y., and Khosla, D. (2015). Spiking deep convolutional neural networks for energy-efficient object recognition. *Int. J. Comput. Vis.* 113, 54–66. doi: 10.1007/s11263-014-0788-3

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. *Proc. Eur. Conf. Comp. Vis.* 12346, 213–229. doi: 10.1007/978-3-030-58452-8_13

Castagnetti, A., Pegatoquet, A., and Miramond, B. (2023). Spiden: deep spiking neural networks for efficient image denoising. *Front. Neurosci.* 17:1224457. doi: 10.3389/fnins.2023.1224457

Chakraborty, B., and Mukhopadhyay, S. (2023). Heterogeneous recurrent spiking neural network for spatio-temporal classification. *Front. Neurosci.* 17:994517. doi: 10.3389/fnins.2023.994517

Chen, D., Peng, P., Huang, T., and Tian, Y. (2022). Deep reinforcement learning with spiking q-learning. *arXiv [Preprint]*. doi: 10.48550/arXiv.2201.09754

Chen, G., Peng, P., Li, G., and Tian, Y. (2023). Training full spike neural networks via auxiliary accumulation pathway. *arXiv [Preprint]*. doi: 10.48550/arXiv.2301.11929

Chen, G. K., Kumar, R., Sumbul, H. E., Knag, P. C., and Krishnamurthy, R. K. (2018). A 4096-neuron 1M-synapse 3.8-pJ/SOP spiking neural network with on-chip STDP learning and sparse weights in 10-nm FinFET CMOS. *IEEE J. Solid State Circ.* 54, 992–1002. doi: 10.1109/JSSC.2018.2884901

Comşa, I.-M., Versari, L., Fischbacher, T., and Alakuijala, J. (2021). Spiking autoencoders with temporal coding. *Front. Neurosci.* 15:712667. doi: 10.3389/fnins.2021.712667

Cordone, L., Miramond, B., and Thierion, P. (2022). "Object detection with spiking neural networks on automotive event data," in *International Joint Conference on Neural Networks (IJCNN)* (Padua), 1–8.

Cuadrado, J., Rançon, U., Cottereau, B. R., Barranco, F., and Masquelier, T. (2023). Optical flow estimation from event-based cameras and spiking neural networks. *Front. Neurosci.* 17:1160034. doi: 10.3389/fnins.2023.1160034

Davies, M. (2021). Taking neuromorphic computing to the next level with Loihi2. *Intel Labs Loihi* 2, 1–7.

Davies, M., Srinivasa, N., Lin, T.-H., Chinya, G., Cao, Y., Choday, S. H., et al. (2018). Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 82–99. doi: 10.1109/MM.2018.112130359

Deng, S., Li, Y., Zhang, S., and Gu, S. (2022). "Temporal efficient training of spiking neural network via gradient re-weighting," in *International Conference on Learning Representations (ICLR)*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2021). "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*.

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

Duwek, H. C., Shalumov, A., and Tsur, E. E. (2021). "Image reconstruction from neuromorphic event cameras using laplacian-prediction and poisson integration with spiking and artificial neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, 1333–1341.

Eshraghian, J. K., Ward, M., Neftci, E. O., Wang, X., Lenz, G., Dwivedi, G., et al. (2023). Training spiking neural networks using lessons from deep learning. *Proc. IEEE* 111, 1016–1054. doi: 10.1109/JPROC.2023.3308088

Fang, W., Chen, Y., Ding, J., Yu, Z., Masquelier, T., Chen, D., et al. (2023a). Spikingjelly: an open-source machine learning infrastructure platform for spike-based intelligence. *Sci. Adv.* 9:eadi1480. doi: 10.1126/sciadv.adi1480

Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., and Tian, Y. (2021b). "Incorporating learnable membrane time constant to enhance learning of spiking neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (Montreal, QC), 2661–2671.

Fang, W., Yu, Z., Zhou, Z., Chen, D., Chen, Y., Ma, Z., et al. (2023b). "Parallel spiking neurons with high efficiency and ability to learn long-term dependencies," in *NIPS '23: Proceedings of the 37th International Conference on Neural Information Processing Systems, Vol. 36* (New Orleans, LA).

Fang, W., Yu, Z., Chen, Y., Huang, T., Masquelier, T., and Tian, Y. (2021a). Deep residual learning in spiking neural networks. *Adv. Neural Inf. Process. Syst.* 34, 21056–21069.

Feng, L., Liu, Q., Tang, H., Ma, D., and Pan, G. (2022). "Multi-level firing with spiking ds-resnet: Enabling better and deeper directly-trained spiking neural networks," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*, 2471–2477.

Feng, Y., Geng, S., Chu, J., Fu, Z., and Hong, S. (2022). Building and training a deep spiking neural network for ecg classification. *Biomed. Signal Process. Control* 77:103749. doi: 10.1016/j.bspc.2022.103749

Frenkel, C., Lefebvre, M., Legat, J.-D., and Bol, D. (2018). A 0.086-mm^2 12.7-pJ/SOP 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm cmos. *IEEE Trans. Biomed. Circuits Syst.* 13, 145–158.

Frenkel, C., Legat, J.-D., and Bol, D. (2019). Morphic: a 65-nm 738k-synapse/mm^2 quad-core binary-weight digital neuromorphic processor with stochastic spike-driven online learning. *IEEE Trans. Biomed. Circuits Syst.* 13, 999–1010. doi: 10.1109/TBCAS.2019.2928793

Goodman, D. F., and Brette, R. (2009). The brian simulator. *Front. Neurosci.* 3:643. doi: 10.3389/neuro.01.026.2009

Guo, N., Bethge, J., Yang, H., Zhong, K., Ning, X., Meinel, C., et al. (2021). Boolnet: minimizing the energy consumption of binary neural networks. *arXiv [Preprint]*. doi: 10.48550/arXiv.2106.06991

Guo, W., Fouda, M. E., Eltawil, A. M., and Salama, K. N. (2021). Neural coding in spiking neural networks: a comparative study for robust neuromorphic systems. *Front. Neurosci.* 15:638474. doi: 10.3389/fnins.2021.638474

Guo, Y., Chen, Y., Zhang, L., Liu, X., Wang, Y., Huang, X., et al. (2022a). IM-loss: information maximization loss for spiking neural networks. *Adv. Neural Inf. Process. Syst.* 35, 156–166. doi: 10.20944/preprints202312.1318.v1

Guo, Y., Tong, X., Chen, Y., Zhang, L., Liu, X., Ma, Z., et al. (2022b). "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 (CVPR)* (New Orleans, LA), 326–335.

Guo, Y., Zhang, Y., Chen, Y., Peng, W., Liu, X., Zhang, L., et al. (2023b). "Membrane potential batch normalization for spiking neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (Paris), 19420–19430.

Guo, Y., Liu, X., Chen, Y., Zhang, L., Peng, W., Zhang, Y., et al. (2023a). "RMP-loss: Regularizing membrane potential distribution for spiking neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (Paris), 17391–17401.

Hamilton, K. E., Mintz, T. M., and Schuman, C. D. (2019). Spike-based primitives for graph algorithms. *arXiv [Preprint]*. doi: 10.48550/arXiv.1903.10574

Hazan, H., Saunders, D. J., Khan, H., Patel, D., Sanghavi, D. T., Siegelmann, H. T., et al. (2018). BINDSNET: a machine learning-oriented spiking neural networks library in python. *Front. Neuroinform.* 12:89. doi: 10.3389/fninf.2018.00089

Hines, M. L., and Carnevale, N. T. (1997). The neuron simulation environment. *Neural Comput.* 9, 1179–1209. doi: 10.1162/neco.1997.9.6.1179

Hu, Y., Tang, H., and Pan, G. (2021a). Spiking deep residual networks. *IEEE Transact. Neural Netw. Learn. Syst.* 34, 5200–5205. doi: 10.1109/TNNLS.2021.3119238

Hu, Y., Wu, Y., Deng, L., and Li, G. (2021b). Advancing residual learning towards powerful deep spiking neural networks. *arXiv*. doi: 10.48550/arXiv.2112.08954

Huang, L., Ma, Z., Zhou, H., and Tian, Y. (2023b). Deep recurrent spiking neural networks capture both static and dynamic representations of the visual cortex under movie stimuli. *arXiv [Preprint]*. doi: 10.48550/arXiv.2306.01354

Huang, L., Ma, Z., Yu, L., Zhou, H., and Tian, Y. (2023a). "Deep spiking neural networks with high representation similarity model visual pathways of macaque and mouse," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)* (Washington, DC), 31–39.

Hunsberger, E., and Eliasmith, C. (2015). Spiking deep networks with lif neurons. *arXiv [Preprint]*. doi: 10.48550/arXiv.1510.08829

Jiang, C., and Zhang, Y. (2023). KLIF: an optimized spiking neuron unit for tuning surrogate gradient slope and membrane potential. *arXiv [Preprint]*. doi: 10.48550/arXiv.2302.09238

Kamata, H., Mukuta, Y., and Harada, T. (2022). "Fully spiking variational autoencoder," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 7059–7067.

Kim, S., Park, S., Na, B., and Yoon, S. (2020). "Spiking-yolo: spiking neural network for energy-efficient object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)* (New York, NY), 11270–11277.

Kim, Y., Chough, J., and Panda, P. (2022). Beyond classification: directly training spiking neural networks for semantic segmentation. *Neuromorp. Comp. Eng.* 2:044015. doi: 10.1088/2634-4386/ac9b86

Kim, Y., and Panda, P. (2021). Revisiting batch normalization for training low-latency deep spiking neural networks from scratch. *Front. Neurosci.* 15:773954. doi: 10.3389/fnins.2021.773954

Kosta, A. K., and Roy, K. (2023). "Adaptive-spikenet: event-based optical flow estimation using spiking neural networks with learnable neuronal dynamics," in *International Conference on Robotics and Automation (ICRA)* (London), 6021–6027.

Kozachkov, L., Kastanenka, K. V., and Krotov, D. (2023). Building transformers from neurons and astrocytes. *Proc. Nat. Acad. Sci. U. S. A.* 120:e2219150120. doi: 10.1073/pnas.2219150120

Kuang, Y., Cui, X., Zhong, Y., Liu, K., Zou, C., Dai, Z., et al. (2021). A 64K-neuron 64M-1b-synapse 2.64 pJ/SOP neuromorphic chip with all memory on chip for spike-based models in 65nm CMOS. *IEEE Transact. Circ. Syst. II* 68, 2655–2659. doi: 10.1109/TCSII.2021.3052172

Kugele, A., Pfeil, T., Pfeiffer, M., and Chicca, E. (2021). "Hybrid SNN-ANN: energy-efficient classification and object detection for event-based vision," in *DAGM German Conference on Pattern Recognition (GCPR)* (Bonn), 297–312.

Lee, C., Kosta, A. K., Zhu, A. Z., Chaney, K., Daniilidis, K., and Roy, K. (2020a). Spike-flownet: event-based optical flow estimation with energy-efficient hybrid neural networks. *Proc. Eur. Conf. Comp. Vis.* 12374, 366–382. doi: 10.1007/978-3-030-58526-6_22

Lee, C., Sarwar, S. S., Panda, P., Srinivasan, G., and Roy, K. (2020b). Enabling spike-based backpropagation for training deep neural network architectures. *Front. Neurosci.* 14:119. doi: 10.3389/fnins.2020.00119

Lee, J.-H., Kim, J.-Y., Noh, S., Lee, H., Lee, S. Y., Mun, J. Y., et al. (2021). Astrocytes phagocytose adult hippocampal synapses for circuit homeostasis. *Nature* 590, 612–617. doi: 10.1038/s41586-020-03060-3

Lee, J. H., Delbruck, T., and Pfeiffer, M. (2016). Training deep spiking neural networks using backpropagation. *Front. Neurosci.* 10:508. doi: 10.3389/fnins.2016.00508

Li, J., and Tian, Y. (2021). Recent advances in neuromorphic vision sensors: a survey. *Chin. J. Comp.* 44, 1258–1286.

Li, J., Zhang, H., Wu, R., Zhu, Z., Wang, B., Meng, C., et al. (2024). "A graph is worth 1-bit spikes: When graph contrastive learning meets spiking neural networks," in *International Conference on Learning Representations (ICLR)*.

Li, W., Zhao, X.-L., Ma, Z., Wang, X., Fan, X., and Tian, Y. (2023b). "Motion-decoupled spiking transformer for audio-visual zero-shot learning," in *Proceedings of the International Conference on Multimedia (MM)*, 3994–4002.

Li, W., Ma, Z., Deng, L.-J., Man, H., and Fan, X. (2023a). "Modality-fusion spiking transformer network for audio-visual zero-shot learning," in *International Conference on Multimedia and Expo (ICME)* (Brisbane, QLD), 426–431.

Li, X., Zhang, X., Yi, X., Liu, D., Wang, H., Zhang, B., et al. (2023). Review of medical data analysis based on spiking neural networks. *Proc. Comput. Sci.* 221, 1527–1538. doi: 10.1016/j.procs.2023.08.138

Li, Y., Guo, Y., Zhang, S., Deng, S., Hai, Y., and Gu, S. (2021). Differentiable spike: rethinking gradient-descent for training spiking neural networks. *Adv. Neural Inf. Process. Syst.* 34, 23426–23439.

Li, Y., He, X., Dong, Y., Kong, Q., and Zeng, Y. (2022). Spike calibration: fast and accurate conversion of spiking neural network for object detection and segmentation. *arXiv [Preprint]*. doi: 10.24963/ijcai.2022/345

Lian, S., Shen, J., Liu, Q., Wang, Z., Yan, R., and Tang, H. (2023). "Learnable surrogate gradient for direct training spiking neural networks," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)* (Macao), 3002–3010.

Liang, J., Li, R., Wang, C., Zhang, R., Yue, K., Li, W., et al. (2022). A spiking neural network based on retinal ganglion cells for automatic burn image segmentation. *Entropy* 24:1526. doi: 10.3390/e24111526

Liang, L., Qu, Z., Chen, Z., Tu, F., Wu, Y., Deng, L., et al. (2021). H2learn: high-efficiency learning accelerator for high-accuracy spiking neural networks. *IEEE Transac. Comp. Aided Des. Integr. Circ. Syst.* 41, 4782–4796. doi: 10.1109/TCAD.2021.3138347

Liu, J.-H., Zhang, M., Wang, Q., Wu, D.-Y., Jie, W., Hu, N.-Y., et al. (2022). Distinct roles of astroglia and neurons in synaptic plasticity and memory. *Mol. Psychiatry* 27, 873–885. doi: 10.1038/s41380-021-01332-6

Liu, M., Wen, R., and Chen, H. (2023). Spiking-diffusion: vector quantized discrete diffusion model with spiking neural networks. *arXiv*. doi: 10.48550/arXiv.2308.10187

Liu, Z., Shen, Z., Savvides, M., and Cheng, K.-T. (2020). ReActNet: towards precise binary neural network with generalized activation functions. *Proc. Eur. Conf. Comp. Vis.* 12359, 143–159. doi: 10.1007/978-3-030-58568-6_9

Liu, Z., Wu, B., Luo, W., Yang, X., Liu, W., and Cheng, K.-T. (2018). Bi-Real Net: enhancing the performance of 1-bit CNNS with improved representational capability and advanced training algorithm. *Proc. Eur. Conf. Comp. Vis.* 11219, 747–763. doi: 10.1007/978-3-030-01267-0_44

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 10012–10022.

Luo, Y., Shen, H., Cao, X., Wang, T., Feng, Q., and Tan, Z. (2022). Conversion of siamese networks to spiking neural networks for energy-efficient object tracking. *Neur. Comp. Appl.* 34, 9967–9982. doi: 10.1007/s00521-022-06984-1

Lv, C., Li, T., Xu, J., Gu, C., Ling, Z., Zhang, C., et al. (2023). SpikeBERT: a language spikformer trained with two-stage knowledge distillation from bert. *arXiv [Preprint]*. doi: 10.48550/arXiv.2308.15122

Ma, D., Jin, X., Sun, S., Li, Y., Wu, X., Hu, Y., et al. (2024). Darwin3: a large-scale neuromorphic chip with a novel isa and on-chip learning. *Natl. Sci. Rev.* 11:nwae102. doi: 10.1093/nsr/nwae102

Ma, D., Shen, J., Gu, Z., Zhang, M., Zhu, X., Xu, X., et al. (2017). Darwin: a neuromorphic hardware co-processor based on spiking neural networks. *J. Syst. Arch.* 77, 43–51. doi: 10.1016/j.sysarc.2017.01.003

Ma, G., Jiang, R., Yan, R., and Tang, H. (2023). "Temporal conditioning spiking latent variable models of the neural response to natural visual scenes," in *NIPS '23: Proceedings of the 37th International Conference on Neural Information Processing Systems, Vol. 36* (New Orleans, LA).

Maass, W. (1997). Networks of spiking neurons: the third generation of neural network models. *Neural Netw.* 10, 1659–1671. doi: 10.1016/S0893-6080(97)00011-7

Meng, Q., Xiao, M., Yan, S., Wang, Y., Lin, Z., and Luo, Z.-Q. (2022). "Training high-performance low-latency spiking neural networks by differentiation on spike representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (New Orleans, LA), 12444–12453.

Meng, Q., Xiao, M., Yan, S., Wang, Y., Lin, Z., and Luo, Z.-Q. (2023). "Towards memory- and time-efficient backpropagation for training spiking neural networks," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (Paris), 6143–6153.

Modha, D. S., Akopyan, F., Andreopoulos, A., Appuswamy, R., Arthur, J. V., Cassidy, A. S., et al. (2023). "IBM northpole neural inference machine," in *2023 IEEE Hot Chips 35 Symposium (HCS)* (California, MA: IEEE Computer Society), 1–58.

Moradi, S., Qiao, N., Stefanini, F., and Indiveri, G. (2017). A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (dynaps). *IEEE Trans. Biomed. Circuits Syst.* 12, 106–122. doi: 10.1109/TBCAS.2017.2759700

Mozafari, M., Ganjtabesh, M., Nowzari-Dalini, A., and Masquelier, T. (2019). SpykeTorch: efficient simulation of convolutional spiking neural networks with at most one spike per neuron. *Front. Neurosci.* 13:625. doi: 10.3389/fnins.2019.00625

Neftci, E. O., Mostafa, H., and Zenke, F. (2019). Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Process. Mag.* 36, 51–63. doi: 10.1109/MSP.2019.2931595

Painkras, E., Plana, L. A., Garside, J., Temple, S., Galluppi, F., Patterson, C., et al. (2013). SpiNNaker: a 1-W 18-core system-on-chip for massively-parallel neural network simulation. *IEEE J. Solid State Circ.* 48, 1943–1953. doi: 10.1109/JSSC.2013.2259038

Panda, P., and Srinivasa, N. (2018). Learning to recognize actions from limited training examples using a recurrent spiking neural model. *Front. Neurosci.* 12:126. doi: 10.3389/fnins.2018.00126

Parameshwara, C. M., Li, S., Fermüller, C., Sanket, N. J., Evanusa, M. S., and Aloimonos, Y. (2021). "Spikems: deep spiking neural network for motion segmentation," in *International Conference on Intelligent Robots and Systems (IROS)*, 3414–3420.

Park, S., Kim, S., Na, B., and Yoon, S. (2020). "T2FSNN: deep spiking neural networks with time-to-first-spike coding," in *ACM/IEEE Design Automation Conference (DAC)* (San Francisco, CA), 1–6.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). "PyTorch: an imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems (NeurIPS), Vol. 32* (British Columbia).

Pehle, C., Billaudelle, S., Cramer, B., Kaiser, J., Schreiber, K., Stradmann, Y., et al. (2022). The brainscales-2 accelerated neuromorphic system with hybrid plasticity. *Front. Neurosci.* 16:795876. doi: 10.3389/fnins.2022.795876

Pei, J., Deng, L., Song, S., Zhao, M., Zhang, Y., Wu, S., et al. (2019). Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature* 572, 106–111. doi: 10.1038/s41586-019-1424-8

Qasim Gilani, S., Syed, T., Umair, M., and Marques, O. (2023). Skin cancer classification using deep spiking neural network. *J. Digit. Imaging* 36, 1137–1147. doi: 10.1007/s10278-023-00776-2

Qiao, N., Mostafa, H., Adi, F., Osswald, M., Stefanini, F., Sumislawska, D., et al. (2015). A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses. *Front. Neurosci.* 9:141. doi: 10.3389/fnins.2015.00141

Qin, L., Yan, R., and Tang, H. (2023). "A low latency adaptive coding spike framework for deep reinforcement learning," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)* (Macao), 3049–3057.

Rançon, U., Cuadrado-Anibarro, J., Cottereau, B. R., and Masquelier, T. (2022). Stereospike: depth learning with a spiking neural network. *IEEE Access* 10, 127428–127439. doi: 10.1109/ACCESS.2022.3226484

Rasmussen, D. (2019). NengoDL: combining deep learning and neuromorphic modelling methods. *Neuroinformatics* 17, 611–628. doi: 10.1007/s12021-019-09424-z

Rathi, N., and Roy, K. (2023). DIET-SNN: a low-latency spiking neural network with direct input encoding and leakage and threshold optimization. *IEEE Transact. Neural Netw. Learn. Syst.* 34, 3174–3182. doi: 10.1109/TNNLS.2021.3111897

Rathi, N., Srinivasan, G., Panda, P., and Roy, K. (2020). "Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation," in *International Conference on Learning Representations (ICLR)* (Addis Ababa).

Ren, D., Ma, Z., Chen, Y., Peng, W., Liu, X., Zhang, Y., et al. (2023). "Spiking pointnet: spiking neural networks for point clouds," in *Advances in Neural Information Processing Systems (NeurIPS)*, *Vol. 36* (New Orleans, LA).

Ren, H., Zhou, Y., Huang, Y., Fu, H., Lin, X., Song, J., et al. (2023). SpikePoint: an efficient point-based spiking neural network for event cameras action recognition. *arXiv [Preprint]*. doi: 10.48550/arXiv.2310.07189

Roy, K., Jaiswal, A., and Panda, P. (2019). Towards spike-based machine intelligence with neuromorphic computing. *Nature* 575, 607–617. doi: 10.1038/s41586-019-1677-2

Rueckauer, B., Lungu, I.-A., Hu, Y., Pfeiffer, M., and Liu, S.-C. (2017). Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Front. Neurosci.* 11:682. doi: 10.3389/fnins.2017.00682

Schemmel, J., Brüderle, D., Grübl, A., Hock, M., Meier, K., and Millner, S. (2010). "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *International Symposium on Circuits and Systems (ISCAS)* (Paris), 1947–1950.

Schuman, C., Plank, J., and Rose, G. (2022). "Application-hardware co-design: System-level optimization of neuromorphic computers with neuromorphic devices," in *2022 International Electron Devices Meeting (IEDM)* (San Francisco, CA: IEEE), 2–4.

Shan, L., Hu, B., Chen, L., and Guan, Z.-H. (2022). "Detecting covid-19 on CT images with impulsive-backpropagation neural networks," in *Chinese Control and Decision Conference (CCDC)* (Hefei), 2797–2803.

Shen, G., Zhao, D., and Zeng, Y. (2023). Exploiting high performance spiking neural networks with efficient spiking patterns. *arXiv [Preprint]*. doi: 10.48550/arXiv.2301.12356

Shen, J., Ma, D., Gu, Z., Zhang, M., Zhu, X., Xu, X., et al. (2016). Darwin: a neuromorphic hardware co-processor based on spiking neural networks. *Sci. China Inf. Sci.* 59, 1–5. doi: 10.1007/s11432-015-5511-7

Shi, X., Hao, Z., and Yu, Z. (2024). "SpikingResformer: bridging resnet and vision transformer in spiking neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Seattle, WA), 5610–5619.

Shrestha, S. B., and Orchard, G. (2018). "Slayer: spike layer error reassignment in time," in *Advances in Neural Information Processing Systems (NeurIPS)*, *Vol. 31* (Montréal, QC).

Soures, N., and Kudithipudi, D. (2019). Deep liquid state machines with neural plasticity for video activity recognition. *Front. Neurosci.* 13:457929. doi: 10.3389/fnins.2019.00686

Su, Q., Chou, Y., Hu, Y., Li, J., Mei, S., Zhang, Z., et al. (2023). "Deep directly-trained spiking neural networks for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (Paris), 6555–6565.

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 (NeurIPS)*, *Vol. 30* (Long Beach, CA).

Wang, B., Dong, G., Zhao, Y., Li, R., Yang, H., Yin, W., et al. (2022). "Spiking emotions: dynamic vision emotion recognition using spiking neural networks," in *Proceedings of the International Conference on Algorithms, High Performance Computing and Artificial Intelligence (AHPCAI)* (Guangzhou), 50–58.

Wang, H., and Li, Y.-F. (2023). Bioinspired membrane learnable spiking neural network for autonomous vehicle sensors fault diagnosis under open environments. *Reliabil. Eng. Syst. Saf.* 233:109102. doi: 10.1016/j.ress.2023.109102

Wang, H., Sun, M., and Li, Y.-F. (2023b). A brain-inspired spiking network framework based on multi-time-step self-attention for lithium-ion batteries capacity prediction. *IEEE Transact. Consum. Electron.* 70:3279135. doi: 10.1109/TCE.2023.3279135

Wang, M., Li, J., Ma, M., and Fan, X. (2023). Spiking semantic communication for feature transmission with Harq. *arXiv [Preprint]*. doi: 10.48550/arXiv.2310.08804

Wang, Q., and Li, P. (2016). "D-LSM: deep liquid state machine with unsupervised recurrent reservoir tuning," in *2016 23rd International Conference on Pattern Recognition (ICPR)* (IEEE), 2652–2657.

Wang, R., Thakur, C. S., Cohen, G., Hamilton, T. J., Tapson, J., and van Schaik, A. (2017). Neuromorphic hardware architecture using the neural engineering framework for pattern recognition. *IEEE Trans. Biomed. Circuits Syst.* 11, 574–584. doi: 10.1109/TBCAS.2017.2666883

Wang, S., Cheng, T. H., and Lim, M. H. (2022). LTMD: learning improvement of spiking neural networks with learnable thresholding neurons and moderate dropout. *Adv. Neural Inf. Process. Syst.* 35, 28350–28362.

Wang, W., Hao, S., Wei, Y., Xiao, S., Feng, J., and Sebe, N. (2019). Temporal spiking recurrent neural network for action recognition. *IEEE Access* 7, 117165–117175. doi: 10.1109/ACCESS.2019.2936604

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., et al. (2021). "Pyramid vision transformer: a versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 568–578.

Wang, X., Zhang, Y., and Zhang, Y. (2023). MT-SNN: enhance spiking neural network with multiple thresholds. *arXiv [Preprint]*. doi: 10.48550/arXiv.2303.11127

Wang, H., Li, Y.-F., and Zhang, Y. (2023a). Bioinspired spiking spatiotemporal attention framework for lithium-ion batteries state-of-health estimation. *Renew. Sustain. Energy Rev.* 188:113728. doi: 10.1016/j.rser.2023.113728

Wang, Y., Shi, K., Lu, C., Liu, Y., Zhang, M., and Qu, H. (2023). "Spatial-temporal self-attention for asynchronous spiking neural networks," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)* (Macao SAR), 3085–3093.

Wang, Y., Zhang, M., Chen, Y., and Qu, H. (2022). "Signed neuron with memory: Towards simple, accurate and high-efficient ann-snn conversion," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)* (Vienna), 2501–2508.

Werbos, P. (1990). Backpropagation through time: what it does and how to do it. *Proc. IEEE* 78, 1550–1560. doi: 10.1109/5.58337

Wiener, M. C., and Richmond, B. J. (2003). Decoding spike trains instant by instant using order statistics and the mixture-of-poissons model. *J. Neurosci.* 23, 2394–2406. doi: 10.1523/JNEUROSCI.23-06-02394.2003

Wu, M.-H., Huang, M.-S., Zhu, Z., Liang, F.-X., Hong, M.-C., Deng, J., et al. (2020). "Compact probabilistic poisson neuron based on back-hopping oscillation in stt-mram for all-spin deep spiking neural network," in *Symposium on VLSI Technology*, 1–2.

Wu, X., He, W., Yao, M., Zhang, Z., Wang, Y., and Li, G. (2022). MSS-DepthNet: depth prediction with multi-step spiking neural network. *arXiv [Preprint]*. doi: 10.48550/arXiv.2211.12156

Wu, Y., Deng, L., Li, G., Zhu, J., and Shi, L. (2018). Spatio-temporal backpropagation for training high-performance spiking neural networks. *Front. Neurosci.* 12:331. doi: 10.3389/fnins.2018.00331

Wu, Y., Deng, L., Li, G., Zhu, J., Xie, Y., and Shi, L. (2019). "Direct training for spiking neural networks: Faster, larger, better," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 1311–1318.

Xiang, S., Zhang, T., Jiang, S., Han, Y., Zhang, Y., Du, C., et al. (2022). Spiking SiamFC++: deep spiking neural network for object tracking. *arXiv [Preprint]*. abs/2209.12010.

Xiao, M., Meng, Q., Zhang, Z., He, D., and Lin, Z. (2022). Online training through time for spiking neural networks. *Adv. Neural Inf. Process. Syst.* 35, 20717–20730.

Xing, X., Zhang, Z., Ni, Z., Xiao, S., Ju, Y., Wang, Y., et al. (2024). "SpikeLM: towards general spike-driven language modeling via elastic bi-spiking mechanisms," in *Forty-First International Conference on Machine Learning (ICML)* (Vienna).

Xiong, Y., Chen, Y., Chen, C., Wei, X., Xue, Y., Wan, H., et al. (2021). An odor recognition algorithm of electronic noses based on convolutional spiking neural network for spoiled food identification. *J. Electrochem. Soc.* 168:077519. doi: 10.1149/1945-7111/ac1699

Xu, Z., Ma, Y., Pan, Z., and Zheng, X. (2022). Deep spiking residual shrinkage network for bearing fault diagnosis. *IEEE Trans. Cybern.* 54, 1–6. doi: 10.1109/TCYB.2022.3227363

Yamazaki, K., Vo-Ho, V.-K., Bulsara, D., and Le, N. (2022). Spiking neural networks and their applications: a review. *Brain Sci.* 12:863. doi: 10.3390/brainsci12070863

Yang, Z., Wu, Y., Wang, G., Yang, Y., Li, G., Deng, L., et al. (2019). DashNet: a hybrid artificial and spiking neural network for high-speed object tracking. *arXiv [Preprint]*. doi: 10.48550/arXiv.1909.12942

Yao, M., Hu, J., Hu, T., Xu, Y., Zhou, Z., Tian, Y., et al. (2024). "Spike-driven transformer v2: meta spiking neural network architecture inspiring the design of next-generation neuromorphic chips," in *The Twelfth International Conference on Learning Representations (ICLR)*.

Yao, M., Zhao, G., Zhang, H., Hu, Y., Deng, L., Tian, Y., et al. (2023b). Attention spiking neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 9393–9410. doi: 10.1109/TPAMI.2023.3241201

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

Yao, M., Hu, J., Zhou, Z., Yuan, L., Tian, Y., Xu, B., et al. (2023a). "Spike-driven transformer," in *Advances in Neural Information Processing Systems (NeurIPS), Vol. 36* (New Orleans, LA).

Yarga, S. Y. A., and Wood, S. U. N. (2023). "Accelerating SNN training with stochastic parallelizable spiking neurons," in *International Joint Conference on Neural Networks (IJCNN)* (Gold Coast, QLD), 1–8.

Yin, N., Wang, M., Chen, Z., De Masi, G., Xiong, H., and Gu, B. (2024). "Dynamic spiking graph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)* (Vancouver, BC), 16495–16503.

Yin, R., Moitra, A., Bhattacharjee, A., Kim, Y., and Panda, P. (2022). SATA: sparsity-aware training accelerator for spiking neural networks. *IEEE Transact. Comp. Aided Des. Integr. Circ. Syst.* 42. 1926–1938. doi: 10.1109/TCAD.2022.3213211

Yin, B., Corradi, F., and Bohté, S. M. (2020). Effective and efficient computation with multiple-timescale spiking recurrent neural networks. *Int. Conf. Neuromor. Syst.* 2020, 1–8. doi: 10.1145/3407197.3407225

Yu, L., Huang, L., Zhou, C., Zhang, H., Ma, Z., Zhou, H., et al. (2024). SVFormer: a direct training spiking transformer for efficient video action recognition. *arXiv [Preprint]*. abs/2406.15034. doi: 10.48550/arXiv.2406.15034

Yuan, L., Hou, Q., Jiang, Z., Feng, J., and Yan, S. (2022). VOLO: vision outlooker for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 6575–6586. doi: 10.1109/TPAMI.2022.3206108

Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.-H., et al. (2021). "Tokens-to-token vit: training vision transformers from scratch on imagenet," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (Montreal, QC), 558–567.

Zhang, H., Fan, X., and Zhang, Y. (2023). Energy-efficient spiking segmenter for frame and event-based images. *Biomimetics* 8:356. doi: 10.3390/biomimetics8040356

Zhang, H., Zhou, C., Yu, L., Huang, L., Ma, Z., Fan, X., et al. (2024). SGLFormer: spiking global-local-fusion transformer with high performance. *Front. Neurosci.* 18:1371290. doi: 10.3389/fnins.2024.1371290

Zhang, J., Tang, L., Yu, Z., Lu, J., and Huang, T. (2022b). Spike transformer: monocular depth estimation for spiking camera. *Proc. Eur. Conf. Comp. Vis.* 13667, 34–52. doi: 10.1007/978-3-031-20071-7_3

Zhang, J., Wang, J., Di, X., and Pu, S. (2022c). "High-accuracy and energy-efficient action recognition with deep spiking neural network," in *International Conference on Neural Information Processing (ICONIP)*, 279–292.

Zhang, X., Lu, J., Wang, Z., Wang, R., Wei, J., Shi, T., et al. (2021). Hybrid memristor-cmos neurons for *in-situ* learning in fully hardware memristive spiking neural networks. *Sci. Bull.* 66, 1624–1633. doi: 10.1016/j.scib.2021.04.014

Zhang, Y., Zhang, Z., and Lew, L. (2022). "PokeBNN: a binary pursuit of lightweight accuracy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12475–12485.

Zhang, J., Huang, L., Ma, Z., and Zhou, H. (2023). Predicting the temporal-dynamic trajectories of cortical neuronal responses in non-human primates based on deep spiking neural network. *Cogn. Neurodyn.* 1–12. doi: 10.1007/s11571-023-09989-1

Zhang, J., Dong, B., Zhang, H., Ding, J., Heide, F., Yin, B., et al. (2022a). "Spiking transformers for event-based single object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (New Orleans, LA), 8801–8810.

Zheng, H., Wu, Y., Deng, L., Hu, Y., and Li, G. (2021). "Going deeper with directly-trained larger spiking neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)* (Vancouver, BC), 11062–11070.

Zhou, C., Yu, L., Zhou, Z., Zhang, H., Ma, Z., Zhou, H., et al. (2023a). Spikingformer: spike-driven residual learning for transformer-based spiking neural network. *arXiv [Preprint]*. doi: 10.48550/arXiv.2304.11954

Zhou, C., Zhang, H., Zhou, Z., Yu, L., Huang, L., Fan, X., et al. (2024). QKFormer: hierarchical spiking transformer using qk attention. *arXiv [Preprint]*. doi: 10.48550/arXiv.2403.16552

Zhou, C., Zhang, H., Zhou, Z., Yu, L., Ma, Z., Zhou, H., et al. (2023b). Enhancing the performance of transformer-based spiking neural networks by SNN-optimized downsampling with precise gradient backpropagation. *arXiv [Preprint]*. doi: 10.48550/arXiv.2305.05954

Zhou, S., Chen, Y., Li, X., and Sanyal, A. (2020). Deep SCNN-based real-time object detection for self-driving vehicles using lidar temporal data. *IEEE Access* 8, 76903–76912. doi: 10.1109/ACCESS.2020.2990416

Zhou, Z., Che, K., Fang, W., Tian, K., Zhu, Y., Yan, S., et al. (2024). Spikformer v2: join the high accuracy club on imagenet with an SNN ticket. *arXiv [Preprint]*.

Zhou, Z., Zhu, Y., He, C., Wang, Y., YAN, S., Tian, Y., et al. (2023). "Spikformer: when spiking neural network meets transformer," in *International Conference on Learning Representations (ICLR)* (Kigali).

Zhu, L., Li, J., Wang, X., Huang, T., and Tian, Y. (2021). "Neuspike-net: High speed video reconstruction via bio-inspired neuromorphic cameras," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (Montreal, QC), 2400–2409.

Zhu, L., and Tian, Y. (2023). Review of visual reconstruction methods of retina-like vision sensors. *Sci. Sinica* 53, 417–436. doi: 10.1360/SSI-2021-0397

Zhu, L., Wang, X., Chang, Y., Li, J., Huang, T., and Tian, Y. (2022). "Event-based video reconstruction via potential-assisted spiking neural network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (New Orleans, LA), 3594–3604.

Zhu, R., Zhao, Q., and Eshraghian, J. K. (2023). SpikeGPT: generative pre-trained language model with spiking neural networks. *arXiv [Preprint]*. doi: 10.48550/arXiv.2302.13939

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. (2021). "Deformable DETR: deformable transformers for end-to-end object detection," in *International Conference on Learning Representations (ICLR)* (Vienna).

Zhu, Z., Peng, J., Li, J., Chen, L., Yu, Q., and Luo, S. (2022). Spiking graph convolutional networks. *arXiv [Preprint]*. doi: 10.24963/ijcai.2022/338

Zou, S., Mu, Y., Zuo, X., Wang, S., and Li, C. (2023). Event-based human pose tracking by spiking spatiotemporal transformer. *arXiv [Preprint]*. doi: 10.48550/arXiv.2303.09681