Exploiting High Performance Spiking Neural Networks With Efficient Spiking Patterns

Guobin Shen, *Student Member, IEEE*, Dongcheng Zhao, and Yi Zeng

***Abstract*—Spiking Neural Networks (SNNs) use discrete spike sequences to transmit information, which significantly mimics the information transmission of the brain. Although this binarized form of representation dramatically enhances the energy efficiency and robustness of SNNs, it also leaves a large gap between the performance of SNNs and Artificial Neural Networks based on real values. There are many different spike patterns in the brain, and the dynamic synergy of these spike patterns greatly enriches the representation capability. Inspired by spike patterns in biological neurons, this paper introduces the dynamic Burst pattern and designs the Leaky Integrate and Fire or Burst (IF&B) neuron that can make a trade-off between short-time performance and dynamic temporal performance from the perspective of network information capacity. IF&B neuron exhibits three modes, resting, Regular spike, and Burst spike. The burst density of the neuron can be adaptively adjusted, which significantly enriches the characterization capability. We also propose a decoupling method that can losslessly decouple IF&B neurons into equivalent LIF neurons, which demonstrates that IF&B neurons can be efficiently implemented on neuromorphic hardware. We conducted experiments on the static datasets CIFAR10, CIFAR100, and ImageNet, which showed that we greatly improved the performance of the SNNs while significantly reducing the network latency. We also conducted experiments on neuromorphic datasets DVS-CIFAR10 and NCALTECH101 and showed that we achieved state-of-the-art with a small network structure.**

***Index Terms*—Adaptive burst neuron, network capacity, spiking neural networks.**

I. Introduction

THE spiking neural networks (SNNs) use discrete spike sequences to convey information, which is more consistent with how the brain processes information. Although the binarized sequences bring high energy efficiency [1] and robustness [2], they also reduce the representation ability of the spiking neural networks. The non-differential nature of the spikes also makes it challenging to apply the backpropagation algorithm directly to the training of SNNs. Therefore, training high-performance SNNs has been a pressing problem for researchers.

In addition to the conversion-based method [3], [4], which converts the well-trained deep neural networks into SNNs, the proposal of surrogate gradient makes it possible to train a high-performance SNNs [5], [6]. Researchers have tried to close the performance gap in several ways. Some researchers have borrowed mature techniques from deep learning and applied techniques such as normalization [7], [8], [9], [10] and attention [11], [12], [13], etc. to the training of SNNs. This greatly improved the performance of SNNs but ignored the characteristics of SNNs. Some researchers have tried to improve and enhance the learning ability of SNNs structurally by borrowing more complex connections in the brain. BackEISNN [14] took inspiration from the autapses in the brain and introduced the self-feedback connection to regulate the precision of the spikes. LISNN [15] modeled the lateral interactions between the neurons and greatly improved the performance and robustness. However, these methods have improved the learning ability of SNNs to some extent, but they are still far from artificial neural networks (ANNs).

Spiking neurons have rich spatio-temporal dynamics and are highly capable of information processing. Beniaguev et al. [16] found that it takes a multilayer neural network to simulate the complexity of a single biological neuron. Aware of the computational power of spiking neurons, researchers have explored several approaches to building more adaptive and efficient neurons. For instance, [17] introduced neural oscillation and spike-phase information to construct resonant spiking neurons, enhancing temporal dynamics. Additionally, hybrid neural coding approaches have been explored, such as integrating rate and temporal coding for pattern recognition tasks [18] and for scalable neural speech coding [19], aiming to balance efficiency and representation capacity. Synaptic delays, as explored by Yu et al. [20], have been utilized to enhance multispike learning, while axonal delays have been proposed as a mechanism for short-term memory in feedforward deep SNNs [21], improving temporal information retention. Furthermore, [22], [23] introduced learnable time constants to boost the performance of SNNs across various tasks. Shen et al. [24] investigated nonlinear dendritic adaptive computation to further extend the

Received 9 October 2024; accepted 31 December 2024. Date of publication 21 February 2025; date of current version 27 March 2025. This work was supported by the National Natural Science Foundation of China under Grant 62406325. *(Guobin Shen and Dongcheng Zhao contributed equally to this work.) (Corresponding author: Yi Zeng.)*

Guobin Shen is with the Brain-Inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Future Technology, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: shenguobin2021@ia.ac.cn).

Dongcheng Zhao is with the Brain-inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: zhaodongcheng2016@ia.ac.cn).

Yi Zeng is with the Brain-inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, also with the University of Chinese Academy of Sciences, Beijing 100049, China, and also with the Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai 200031, China (e-mail: yi.zeng@ia.ac.cn).

Recommended for acceptance by M. Zhang.

Digital Object Identifier 10.1109/TETCI.2025.3540408

dynamical range of spiking neurons. Other researchers have proposed adaptive mechanisms like adjustable thresholds [25], [26], which control the firing of spiking neurons. These advancements greatly enrich the dynamical properties of spiking neurons, but the inherent binary representation of SNNs continues to create a performance gap compared to float-based ANNs.

Other researchers tried to design a better surrogate gradient function to reduce the information mismatch caused by inaccurate gradients in backpropagation. [27] proposed a gradual surrogate gradient learning algorithm to ensure the precision as well as the effectiveness of the gradient during backpropagation. [25] proposed activity-regularizing surrogate gradients, which exceeded the state-of-the-art performance for SNNs on the challenging temporal benchmarks. [28] introduced the adaptively evolved Differentiable Spike functions to find the optimal shape and smoothness for gradient estimation based on their finite difference gradients. However, the binarized information transfer method still limits the representation ability of SNN.

As a result, some researchers try to enrich the representation ability of SNNs. [29], [30] introduced the negative spikes to cooperate with the regular positive spikes. However, the behavior of releasing negative spikes below the threshold is not consistent with the human brain. [31] proposed the leaky integrate and analog fire neuron model to transmit the analog values among neurons, bringing performance improvements and significantly increasing energy consumption. The brain does not maintain a single spiking pattern for the same input. The coupling of different spiking patterns greatly enriches the representation ability of the spiking neurons and will adaptively cooperate to complete different cognitive functions. As the most commonly observed pattern in different brain regions, bursts might improve the selective communication between neurons [32], the number of spikes of the high-frequency bursts is highly robust to noise [33]. Although there exist some works with the burst spikes [34], [35], their burst intensities are fixed and do not change dynamically according to the input.

In this paper, we introduce the modeling of Leaky Integrate and Fire or Burst (IF&B) neurons with three spiking patterns: resting, regular spike, and burst spike. Experiments show that our algorithm not only dramatically improves the performance of the current SNNs, but also significantly reduces the latency and energy consumption. Our contributions are summarized as follows:

- We introduce the information capacity of SNNs and discuss the relationship between the information capacity and the simulation length and spike patterns. We experimentally verify that the neurons with triple-valued spike representation have the best performance under the same upper bound of information capacity.
- We improved the Leaky Integrate and Fire or Burst neuron (IF&B) so that the bursting strength of neurons can be adaptively adjusted while maintaining the optimal triple-valued spike representation while further improving the performance of SNNs, as shown in Fig. 1. We also exhibit that an IF&B neuron can be decoupled into two LIF neurons, which allows the IF&B neuron to be easily implemented on neuromorphic hardware.

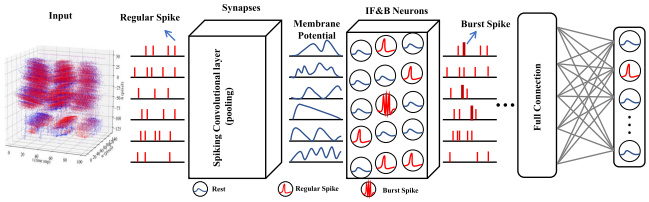

Fig. 1. Illustration of the spiking neural network with our Leaky Integrate and Fire or Burst Neuron. After the network receives the input, the IF&B neurons show three spiking states: regular spike, burst spike, and resting, which significantly improve the representation ability of SNNs.

- We conduct experiments on the static image datasets CIFAR10, CIFAR100, and ImageNet and the neuromorphic datasets DVS-CIFAR10 and NCALTECH101 to verify the superiority of our model. We achieve state-of-the-art performance on these datasets and achieve excellent performance using only minor simulation steps.

II. Method

A. Leaky Integrate and Fire Model

The spiking neuron serves as the fundamental computational unit in SNNs. Neuroscientists have established various mathematical models such as the Hodgkin-Huxley spiking neuron (H-H) [36], the Izhikevich spiking neuron [37], Leaky Integrate and Fire spiking neuron (LIF) [38] to describe the dynamic characteristics of biological neurons. More complex mathematical models can also better describe the computational process of biological neurons. However, they also require more computational resources, while the overly complex properties are challenging to apply to the modeling of large-scale SNNs. As the most common spiking neuron model, the LIF neuron model is widely used in deep SNNs.

$$\tau \frac{\mathrm{d}v}{\mathrm{d}t} = -v + I,$$

$$\text{if } v > v_{th} \text{ , then } v \leftarrow v_{rst} \tag{1}$$

$$s = H(v - v_{th}) \tag{2}$$

In the (1), v is the membrane potential, I is the input current, and v_{th} is the threshold. When the neuron reaches the threshold, it will deliver a spike, and the membrane potential is reset to the resting potential v_{rst}. τ is the membrane time constant, which controls the rate of decay of the membrane potential over time. s denotes the neuronal spikes, $H(\cdot)$ denotes the heaviside step function.

The LIF model can be regarded as an integrator, capable of firing regular spikes at a constant rate and adjusting the firing rate according to the input current. To facilitate the calculation, we obtain the discrete form of (1):

$$v(t+1) = v(t) + \frac{1}{\tau}(-v(t) + I(t)) \tag{3}$$

Although there are many improvements for spiking neurons, they are limited to LIF neurons. The over-simplified computational characteristics of LIF neurons make it only possible to

characterize regular spikes and cannot describe complex spiking patterns. There is a big gap between the LIF model and real biological neurons.

B. Information Capacity for SNNs

Spiking neurons convert continuous membrane potentials into discrete spikes $s \in S = \{0, 1\}$, using spike trains over T time steps to transmit information. This differs significantly from the continuous, real-valued outputs used in ANNs, leading to differences in performance and computational requirements. Here, we analyze how simulation length and other factors influence the performance of SNNs and explore the relationship between SNNs and ANNs from the perspective of information capacity.

Consider a spiking neuron whose output is represented as a t-dimensional Boolean vector $S = \{0, 1\}^t$, where each element in the vector corresponds to a spike (1) or no spike (0) at a particular time step. The behavior of this neuron can be described using a (linear) threshold function $f : S \to \{0, 1\}$, defined as:

$$f(s) = H(\langle a, s\rangle + b), \quad s \in S, \tag{4}$$

where $\langle a, s\rangle$ denotes the dot product between a weight vector $a \in \mathbb{R}^t$ and the spike train s, and b is a bias term. The Heaviside step function $H(\cdot)$ outputs 1 if its argument is positive and 0 otherwise. This function models the decision-making process of a spiking neuron, determining whether it fires based on the input spike pattern s.

The set of all possible threshold functions on S is denoted by $T(S)$. The size of this set, $|T(S)|$, represents the number of distinct ways the neuron can respond to different spike trains. We define the information capacity $C(S)$ of a spiking neuron in terms of the logarithm of the number of threshold functions:

$$C(S) = \log_2 |T(S)|. \tag{5}$$

This equation quantifies how many different input-output mappings (threshold functions) the neuron can implement. The more threshold functions a neuron can realize, the higher its capacity to differentiate between various input spike patterns.

To estimate $|T(S)|$, we can use a geometric interpretation of threshold functions. The set S can be viewed as a t-dimensional space, where each dimension corresponds to a time step in the spike train. Each threshold function partitions this space using hyperplanes, which separate different regions where the neuron outputs 0 or 1.

The number of regions $L(m, n)$ that m hyperplanes can create in an n-dimensional space (with $n = t$) is given by:

$$L(m, n) \leq 2 \sum_{k=0}^{n-1} \binom{m-1}{k}, \tag{6}$$

where $\binom{m-1}{k}$ is a binomial coefficient. This inequality describes the maximum number of connected regions that can be formed by m hyperplanes passing through the origin in $\mathbb{R}^n$. Each region corresponds to a different output behavior of the threshold function. Here, $m = |S| = 2^t$ represents the number of different spike patterns, and $n = t$ is the dimensionality of the space.

Using (6), we can approximate the upper bound of $|T(S)|$:

$$|T(S)| \leq 2 \sum_{k=0}^{t-1} \binom{2^t - 1}{k}. \tag{7}$$

To estimate the sum of binomial coefficients, we use a basic inequality for binomials:

$$\binom{n}{k} \leq \left(\frac{en}{k}\right)^k, \tag{8}$$

which holds when $k \leq n$. Applying this to our binomial terms, we get:

$$\binom{2^t - 1}{k} \leq \left(\frac{e \cdot 2^t}{k}\right)^k \leq \left(\frac{e \cdot 2^t}{t}\right)^t. \tag{9}$$

This simplification leads to:

$$\sum_{k=0}^{t-1} \binom{2^t - 1}{k} \approx \left(\frac{e \cdot 2^t}{t}\right)^t. \tag{10}$$

Substituting back into (7), we find:

$$|T(S)| \leq 2 \left(\frac{e \cdot 2^t}{t}\right)^t. \tag{11}$$

Taking the logarithm yields the information capacity:

$$C(S) \leq \log_2 \left[2 \left(\frac{e \cdot 2^t}{t}\right)^t\right]. \tag{12}$$

Expanding this further gives:

$$C(S) \leq 1 + t \log_2 \left(\frac{e \cdot 2^t}{t}\right) \tag{13}$$

$$= 1 + t \log_2(e) + t^2 - t \log_2(t). \tag{14}$$

For a spiking neuron with a simulation length t and $|S| = 2^t$, this can be approximated as:

$$C(S) \leq 1 + t^2 - t \log_2 \left(\frac{t}{e}\right). \tag{15}$$

This result shows that the information capacity of a spiking neuron increases with the square of the simulation length t, modulated by a term $t \log_2 t$.

In contrast, ANNs use floating-point numbers to represent neuron activations, allowing them to convey information more compactly than SNNs. An ANN neuron's activation value can be considered as a compressed representation of a spike train. Thus, theoretically, ANNs and SNNs with the same simulation length have similar information capacities, but the way they utilize that capacity differs. ANNs condense the information into a single value, while SNNs distribute it across a sequence of discrete spikes.

By compressing spike trains into floating-point values, ANNs achieve higher training and inference efficiency, particularly with gradient descent optimization, yielding better performance. SNNs, however, establish temporal relationships among spikes, making them better suited for processing temporal data, particularly on neuromorphic hardware optimized for accumulate operations (ACs). Nevertheless, this temporal nature complicates

SNN optimization, leading to performance gaps compared to ANNs.

Efficiently organizing spike sequences to maximize performance given a limited information capacity remains a key challenge. Inspired by neuroscience, we recognize that biological neurons exhibit complex spiking patterns that cannot be easily captured by short binary spike sequences. By incorporating more biologically plausible neuron models and optimizing the organization of spike sequences, we can significantly enhance the performance of SNNs.

C. Leaky Integrate and Fire or Burst Model

Burst is a vital pattern in neurons, which contributes to gamma frequency oscillations in the brain, helps reduce neuronal noise [39], and facilitates selective communication between neurons [32]. The Leaky Integrate and Fire or Burst (IF&B) model [40] is an extension of the LIF model, which retains the function of membrane potential accumulation with input current while introducing a calcium T-current parameter, bringing new properties of the burst pattern.

The original IF&B model can better describe the spiking pattern of neurons, but it also brings about three times the computational costs of the LIF model [41]. Meanwhile, since the current directly trained SNNs often have a small simulation step (4 or even shorter), it is not easy to show the difference between regular spikes and burst spikes, which also limits the application of the IF&B model to large-scale SNNs.

Previous studies have made significant strides in the field of large-scale SNNs by successfully compressing various spike patterns into a single simulation step [34], [35]. This technique has enabled the implementation of neurons with burst patterns in large-scale networks, resulting in improved performance. However, this approach also poses a significant challenge in terms of computational complexity. In this work, we delve deeper into the impact of neurons with burst patterns on SNNs, specifically from the perspective of network information capacity. Through rigorous experimentation, we demonstrate that the use of neurons with burst patterns not only improves performance, but can do so while maintaining the same upper bound on information transmission.

$$\tau \frac{\mathrm{d}v}{\mathrm{d}t} = -v + I,$$

$$\text{if } v > v_{th} \text{ , then } v \leftarrow v_{rst} \tag{16}$$

$$s = H(v - v_{th}) + (\kappa - 1)H(v - v_h) \tag{17}$$

In order to capture the impact of burst spikes on neurons in rough time simulations, we take a macroscopic view of their effect over a shorter time frame. This approach involves superimposing the influence of T-currents on the output of the neuron. To ensure efficient implementation of burst spiking, we employ the use of a trainable parameter κ to represent the burst strength. This allows for the burst strength to be adaptively adjusted during the training process and also ensures a more efficient hardware implementation of the simplified IF&B mechanism.

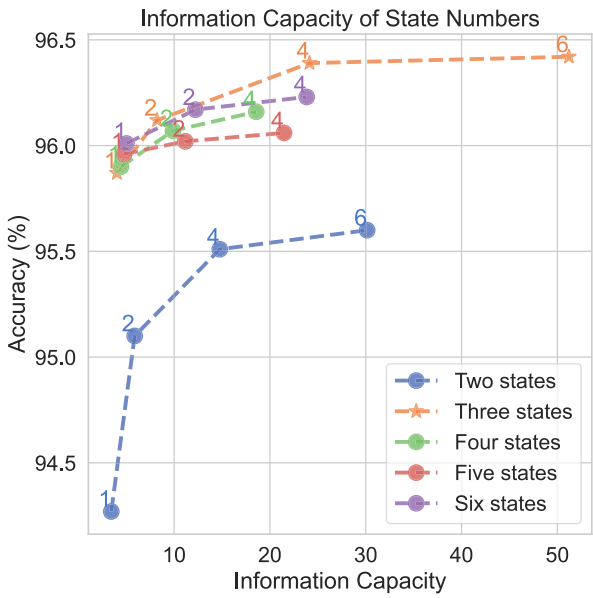

Fig. 2. The relationship between information capacity and performance for different numbers of states on CIFAR10 using SEW-ResNet-18. The colored lines represent the number of states each neuron can take at each time step, while the numbers next to each point indicate the corresponding simulation time.

D. How to Represent the Burst State

Biological neurons can transmit a large amount of information in a short period by burst spikes. The introduction of the burst spike mechanism has dramatically expanded the representation ability of LIF neurons but also brings additional computational overhead. Therefore, it is an attractive problem to define the trade-off between computational overhead and the performance of IF&B neurons.

We consider the problem of representing burst spikes from the perspective of the information capacity of neural networks. A sequence of spikes with bursts of length t, can be represented as $s = \{0, 1, \kappa_1, \kappa_2 \ldots, \kappa_{n-2}\}^t$. where n is the possible states at the moment. According to (14), for such a sequence of spikes with burst states, the upper bound of its information capacity is:

$$C(S) \leq 1 + t^2 \log_2(n) - t \log_2 \left(\frac{1}{e} t\right) \tag{18}$$

To select an efficient representation of burst spikes, we tested the relationship between the upper bound of information capacity and the performance of spike sequences with a different number of states.

As shown in Fig. 2, the neuron with only two states (LIF neuron) has the lowest performance. The introduction of the burst mechanism has tremendously improved the neuron's performance. It performs better if the neuron can represent multiple states at one moment. However, as the simulation time increases, too many spike states will affect the overall performance of the neuron, while $n = 3$ is a trade-off between the short-time performance of the neuron and the overall performance with temporal information. Therefore, we will use $S = \{0, 1, \kappa\}$ to define the output states of the IF&B neuron.

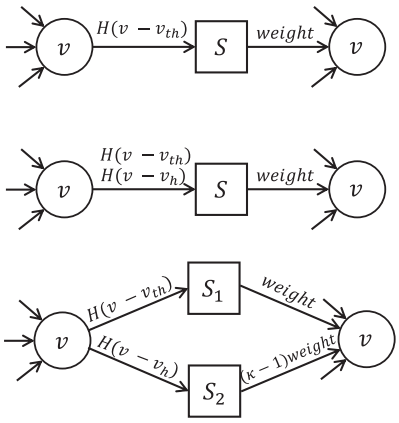

Fig. 3. A IF&B neuron can be decoupled into two LIF neurons. From top to bottom: LIF neuron, IF&B neuron, and LIF neuron decoupled by IF&B neuron.

It is worth mentioning that to introduce very few parameters and increase the support of neurons for burst spiking, we consider a learnable, channel-sharing burst intensity of κ. All neurons of the same channel use the same κ, which is negligible compared to the number of parameters of the network. To allow the burst intensity to be optimized, κ is adjusted during the training process using gradient descent along with other parameters. We use the momentum method to ensure the stability of the burst intensity:

$$\Delta\kappa := \mu\Delta\kappa + \epsilon\frac{\partial\mathcal{E}}{\partial\kappa} \tag{19}$$

In (19), $\frac{\partial\mathcal{E}}{\partial\kappa}$ denotes the gradient propagated from the deep layer. μ is the momentum, and ϵ is the learning rate. We do not restrict the range of κ, and use $\kappa = 1$ as the initial value.

1) Decoupling of IF&B Neurons: We present a method that can efficiently implement IF&B neurons on neuromorphic hardware that is fully compatible with existing hardware without any modification. The IF&B neuron exhibit two different spiking patterns, so it can also be decoupled into two neurons with the same input current, as shown in Fig. 3. This makes it easy to deploy IF&B neurons on hardware designed for LIF neurons while achieving better performance.

III. Experiment

In this section, we evaluate the performance of the proposed IF&B Neuron on the image datasets CIFAR10 [42], CIFAR100 [43] and ImageNet [44] and the neuromorphic datasets DVS-CIFAR10 [45] and NCALTECH101 [46] with BrainCog [47]. The model structures used in this paper include VGG16 [48], ResNet20 [49], ResNet19 [8], ResNet18-sew [50], and SNN6 (64C3-128C3-AP2- 256C3-AP2- 512C3-AP2-512C3-AP2-FC).

In the IF&B Neuron, v_{th} is set to 0.5, v_{rst} is set to 0, and v_h, representing the threshold for burst firing, is set to twice v_{th}, i.e., 1.0. τ is the membrane time constant and is set to 2.0. κ represents the burst intensity and is initialized to 1.0. We use the Adam optimizer for training, with a learning rate set to 5×10^{-3} and momentum set to 0.9. The batch size is set to 64. For all datasets, the training is set for 300 epochs. Since the spiking process is non-differentiable, a surrogate gradient [51] is used to enable backpropagation [52]. We use the same surrogate gradient function as Wu et al. [5], as shown below:

$$\frac{\partial s_t}{\partial u_t} = (\alpha - \alpha^2 |u_t|)\,\mathrm{sign}\left(\frac{1}{\alpha} - |u_t|\right) \tag{20}$$

In (20), α controls the width of the surrogate gradient function, which is set to 2 in our experiments.

TABLE I
Comparison With Existing Works on CIFAR10 and CIFAR100

Model	Architecture	Simulation Length	CIFAR10	CIFAR100
Bu et al. [3]	ResNet-18	4	90.43	75.67
Rathi et al. [53]	ResNet-20	250	92.22	67.87
Rathi & Roy [54]	ResNet-20	10	92.54	64.07
Wu et al. [5]	CIFARNet	12	89.83	-
Wu et al. [9]	CIFARNet	12	90.53	-
Zhang & Li [6]	CIFARNet	5	91.41	-
Shen et al. [1]	7-layer-CNN	8	92.15	69.32
Kim et al. [55]	NAS	5	92.73	73.04
Na et al. [56]	NAS	16	93.15	69.16
Zheng et al. [8]	ResNet-19	6	93.16	-
Deng et al. [57]	ResNet-19	6	94.50	74.72
Guo et al. [58]	ResNet-19	6	95.55	74.10
Guo et al. [59]	ResNet-19	2	95.80	80.20
Zhou et al. [60]	Spikformer	4	95.19	77.86
Zhou et al. [61]	Spikingformer	4	95.61	79.09
Our Method	ResNet-19	1	95.94±0.09	77.86±0.43
	ResNet-19	2	96.01±0.07	78.04±0.37
	ResNet-19	4	96.21±0.10	78.12±0.51
	ResNet-19	6	**96.32**±0.06	78.31±0.58
	Spikformer	1	95.56±0.07	**79.84**±0.46
	Spikformer	4	96.03±0.04	**80.72**±0.42

A. Comparison With Other Methods

To verify the effectiveness of our algorithm, we compare it with several current best SNNs, including conversion-based and backpropagation-based. The results for the static image data and the neuromorphic data classification task are listed in Tables I, II, and III. For the static image datasets, we report the performance at simulation steps 1, 2, 4, and 6 to highlight the accuracy at different temporal scales.

For CIFAR10 and CIFAR100, IF&B achieves higher accuracy than previous work at a simulation length of 1. In particular, using the same network structure and simulation step length, our IF&B also has a significant advantage, improving 0.59% and 3.31% on CIFAR10 and CIFAR100 compared with RecDis [58].

For the more challenging ImageNet dataset, we achieve 65.60% accuracy using only the lightweight ResNet18 structure. Moreover, we achieve better performance than SEW-ResNet152. [22] when using the SEW-ResNet34 structure only at a simulation length of 1. Our IF&B achieves 2.02% improvements using the same structure and simulation length as previous work.

When $t = 1$, our IF&B neurons may appear similar to quantized artificial neural networks (ANNs) with multiple states.

TABLE II
COMPARISON WITH EXISTING WORKS ON IMAGENET

Model	Methods	Architecture	Simulation Length	Params (M)	OPs (G)	Power (mJ)	Accuracy
Rathi et al. [53]	Hybrid Training	ResNet-34	250	21.79	-	-	61.48
Guo et al. [59]	Ternary Spike	ResNet-34	4	21.79	-	-	70.74
Zheng et al. [8]	STBP-tdBN	Spiking-ResNet-34	6	21.79	6.50	6.39	63.72
Fang et al. [22]	SEW ResNet	SEW-ResNet-34	4	21.79	3.88	4.04	67.04
		SEW-ResNet-50	4	25.56	4.83	4.89	67.78
		SEW-ResNet-101	4	44.55	9.30	8.91	68.76
		SEW-ResNet-152	4	60.19	13.72	12.89	69.76
Zhou et al. [60]	Spikformer	Spikformer-8-384	4	16.81	6.82	12.43	70.24
		Spikformer-8-512	4	29.68	11.09	18.82	73.38
		Spikformer-8-768	4	66.34	22.09	32.07	74.81
Zhou et al. [61]	Spikingformer	Spikingformer-8-384	4	16.81	3.88	4.69	72.45
		Spikingformer-8-512	4	29.68	6.52	7.46	74.79
		Spikingformer-8-768	4	66.34	12.54	13.68	75.85
Our Method	SEW ResNet	SEW-ResNet-18	1	11.68	0.54	0.53	65.60
		SEW-ResNet-34	1	21.79	1.07	1.04	69.34
		SEW-ResNet-34	4	21.79	4.27	4.21	70.02
	Spikformer	Spikformer-8-512	1	29.68	2.78	5.07	72.47
		Spikformer-8-512	4	29.68	11.09	21.03	**76.64**

TABLE III
COMPARISON WITH EXISTING WORKS ON NEUROMORPHIC DATASETS

Dataset	Model	Architecture	Simulation Length	Accuracy
DVS-CIFAR10	Zheng et al. [8]	ResNet-19	10	67.8
	Kugele et al. [62]	DenseNet	10	66.8
	Wu et al. [31]	LIAF-Net	10	71.7
	Wu et al. [31]	LIAF-Net	10	70.4
	Na et al. [56]	NAS	16	72.5
	Shen et al. [1]	5-layer-CNN	16	78.9
	Guo et al. [58]	ResNet-19	10	72.4
	Deng et al. [57]	VGGSNN	10	83.2
	Zhou et al. [60]	Spikformer	10	78.6
	Zhou et al. [61]	Spikingformer	10	80.6
	Our Method	SNN7	10	**83.8**±0.70
N-Caltech101	Kugele et al. [62]	VGG11	20	55.0
	Ramesh et al. [63]	N/A	N/A	66.8
	Our Method	SNN7	10	**81.7**±0.81

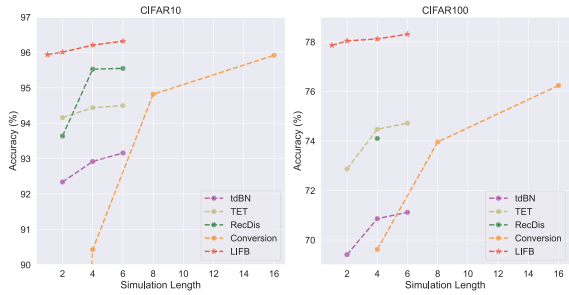

Fig. 4. Relationship between simulation length and accuracy on the CIFAR10/100 dataset.

However, quantized ANNs handle different states through multiply/add operations with carry, managing discrete levels of quantization. In contrast, our approach introduces a burst mechanism, allowing neurons to dynamically adjust their spiking strength in response to varying inputs, enriching temporal representation. The burst mechanism enables IF&B neurons to achieve a more efficient and biologically plausible representation, capturing richer temporal dynamics compared to the arithmetic manipulation of discrete values in quantized ANNs.

The energy efficiency of our method is notable. As shown in Table II, for the SEW-ResNet-34 architecture, our IF&B method achieves 2.3% higher accuracy than the original SEW ResNet while requiring only a single simulation step. Moreover, the energy consumption of IF&B is approximately one-quarter of that of the original method, consuming just 1.04 mJ. This demonstrates that although our method involves fewer bursts, these bursts are highly effective, leading to both accuracy improvements and lower energy requirements.

A similar advantage is observed with the Spikformer architecture. At a simulation length of 1, IF&B with the Spikformer-8-512 structure achieves competitive accuracy while consuming significantly less power, using only 2.78 mJ compared to the 5.07 mJ required by the original Spikformer. In terms of energy consumption calculations, we have demonstrated that a single IF&B neuron can be decoupled into two LIF neurons, allowing us to use the same calculation method as Spikformer. Additionally, we treat the energy consumed by spikes and bursts as equivalent, summing them for the final calculation. This approach ensures a fair comparison while highlighting the efficiency of our method in real-world applications where energy constraints are crucial.

For the neuromorphic dataset DVS-CIFAR10, our IF&B achieves state-of-the-art performance by using only the SNN7 structure with less than half the parameters of VGGSNN [57]. For the N-Caltech101 dataset, we achieved 83.44% top-1 accuracy, achieving a performance far beyond previous work.

To further illustrate the advantages of our IF&B, we show the comparison with previous methods at different simulation lengths. As shown in Fig. 4, we compared IF&B with directly trained SNNs and converted SNNs. Our IF&B shows a significant advantage at shorter simulation lengths due to its more vital representation ability.

B. Ablation Studies

Compared with other advanced methods, IF&B allows models to exhibit better performance and achieve better top-1 accuracy

TABLE IV
ABLATION STUDIES ON DIFFERENT NETWORK STRUCTURES AND SIMULATION LENGTHS

Architecture	Neuron	CIFAR10				CIFAR100			
		T=1	T=2	T=4	T=6	T=1	T=2	T=4	T=6
VGG16	LIF	92.02	93.41	94.13	94.08	66.40	69.18	71.15	71.99
	IF&B	94.53	95.02	95.28	95.36	73.00	73.90	74.34	75.02
ResNet19	LIF	93.76	94.44	95.07	95.51	73.70	74.34	75.01	75.62
	IF&B	95.94	96.01	96.21	96.32	77.86	78.04	78.12	78.31
ResNet20	LIF	83.65	86.47	88.09	89.16	49.93	53.90	57.40	58.17
	IF&B	89.72	90.88	91.30	91.65	60.63	62.95	63.28	64.33
SEW-ResNet18	LIF	94.27	95.10	95.51	95.60	72.59	74.16	75.68	76.52
	IF&B	95.87	96.12	96.39	96.42	75.88	77.38	78.41	78.67

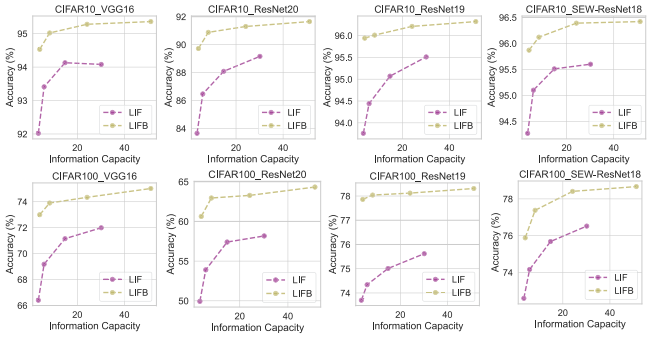

Fig. 5. Effect of neuron type on information capacity and performance.

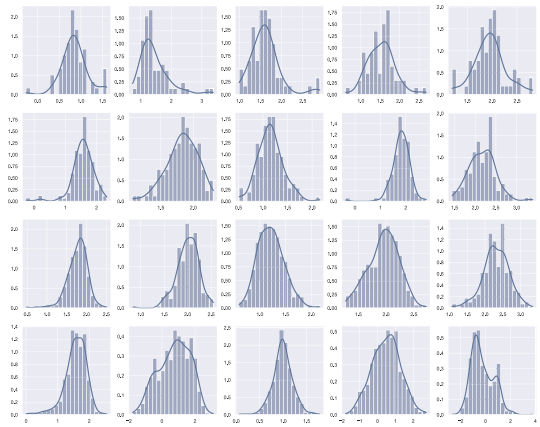

Fig. 6. Distribution of burst intensity of well-trained SEW-ResNet18 on CIFAR10.

on classification tasks. We conducted ablation studies to further verify the contribution of IF&B for different network structures and simulation lengths.

As shown in Table IV, IF&B maintained its advantage over LIF neurons for all models and all simulation lengths. IF&B also has higher accuracy than LIF neurons with longer simulation time only at the simulation length $t = 1$. Fig. 5 shows a comparison of the impact of neuron type in terms of the information capacity of neurons. It can be seen that our IF&B still maintains a higher accuracy than LIF with the same information capacity.

In the original IF&B model, the switching between different neuronal spiking patterns, such as regular spikes and burst spikes, is achieved by adjusting the conductance of the calcium T-current. This biologically inspired mechanism involves slow adjustments that are difficult to apply effectively in short simulation settings, especially in large-scale spiking neural networks (SNNs) due to their computational cost. To address this, we propose a simplified version of the IF&B neuron, which directly integrates the effect of T-current conductance into the neuron output through a learnable parameter κ. This parameter allows for the dynamic adjustment of burst intensity, thus enhancing the flexibility of neuronal behavior while maintaining computational efficiency.

Fig. 6 illustrates the distribution of burst intensity across different layers of SEW-ResNet18 trained on the CIFAR10 dataset. This figure shows how burst intensity varies between layers, highlighting that the adaptive burst mechanism is not uniformly distributed but instead tailored across the network based on each layer's specific role in feature extraction and information processing. The variation observed indicates that the deeper layers tend to exhibit more dynamic burst activities, while the earlier layers maintain a more stable pattern. This suggests that deeper layers may benefit more from dynamic adjustments, leveraging the learnable κ to better capture and process complex, high-level features.

TABLE V
COMPARISON OF FIXED/LEARNABLE BURST INTENSITY OF VGG16 ON CIFAR10

κ	**LIF**	**0.5**	**1**	**1.5**	**2**	**learnable**
	92.02	92.13	94.17	94.11	93.52	**94.53**

Additionally, we conducted an ablation study to further assess the impact of learnable burst intensity. As shown in Table V, we compared neurons with fixed burst intensity to those with learnable burst intensity on the CIFAR10 dataset, observing significant improvements in top-1 accuracy for the latter. This highlights the importance of adaptive burst mechanisms in boosting the performance of SNNs. Fig. 6 effectively visualizes the diverse burst intensities learned across the network, emphasizing how our proposed approach enables efficient and flexible spiking behaviors that enhance the model's overall representational power.

While both IF&B neurons and PosNeg neurons aim to enhance the representational capability of spiking neurons, there is a fundamental difference in their approach. IF&B neurons leverage burst mechanisms, allowing different channels to exhibit

TABLE VI
COMPARISON OF IF&B NEURON WITH POSNEG NEURON ON CIFAR10

Neuron / Step	T=1	T=2	T=4	T=6
PosNeg	93.87	94.16	94.33	94.43
IF&B	94.53	94.91	95.17	95.15

varying burst intensities, which provides rich representational capabilities and aligns more closely with biological plausibility. In contrast, PosNeg neurons utilize both positive and negative spikes, which may enhance representation in certain contexts but diverges from the typical spiking behavior observed in biological neurons. IF&B neurons maintain a more natural spiking pattern while still achieving strong performance.

The learnable channel-sharing burst intensity enables neurons to learn the appropriate burst from the data, and neurons can achieve bursts of arbitrary strength compared to regular spikes, which is more biologically plausible and enhances the performance of SNNs.

C. Comparison With Other Triple-Value Neurons

By considering the different spiking patterns of neurons, we design IF&B neurons that can exhibit triple neuronal states: rest, regular spike, and burst spike. Our IF&B neuron expands the representation ability of neurons, is more biologically plausible, and exhibits better energy efficiency and performance than other binary-value neurons.

In addition to our IF&B neuron, there are many works that enable neurons to fire positive and negative spikes (PosNeg) to achieve triple-value representation [29], [30]. Here we consider the same approach.

We compare our IF&B neuron with the PosNeg at different simulation lengths on CIFAR10 dataset as shown in Table VI. Our IF&B neuron shows better performance at different simulation lengths.

D. Loss Landscape Around Local Minima

We further show the 2D landscapes of SNNs with different types of neurons around their local minima [64] to verify the effect of IF&B neurons on the generalization ability. As shown in Fig. 7, we show the local 2D landscape of the VGG16 model on CIFAR10/100, using different neurons. It can be seen that IF&B Neuron finds flatter local minima and more minor losses. This further demonstrates the ability of the IF&B neuron to enhance the representation and generalization of the model.

E. Comparison of IF&B With Decoupled LIF

IF&B neuron achieves much better performance than LIF Neuron by better modeling biological neurons, but this also entails additional computational overhead. Although the above experiments have demonstrated that IF&B neurons perform better than LIF neurons with longer simulation times, this cannot be achieved by increasing computational resources. To further illustrate that the performance gain of IF&B neurons comes from more reasonable modeling rather than more computational resources, we performed a fairer comparison.

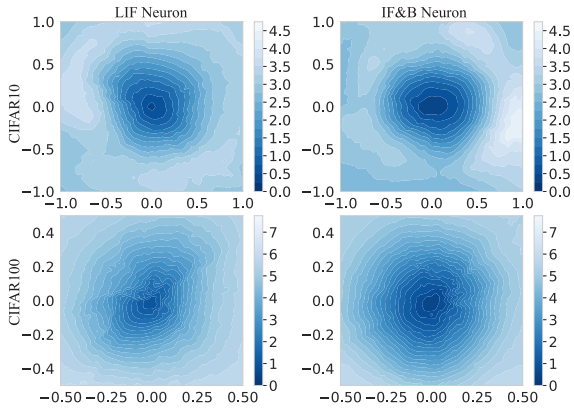

Fig. 7. Comparison of loss landscapes of different neurons on VGG16.

TABLE VII
COMPARISON OF IF&B NEURONS AND EQUIVALENT DECOUPLED LIF NEURONS TRAINED FROM SCRATCH ON VGG16

Neuron / Step	T=1	T=2	T=4	T=6
LIF	92.02	93.41	94.13	94.08
Scratch	93.78	94.36	94.83	95.02
IF&B	94.53	95.02	95.28	95.36

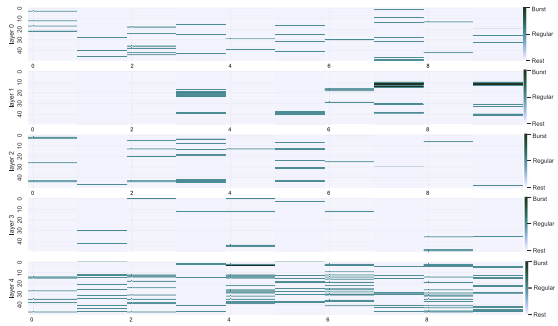

Fig. 8. Visualization of neural activity of IF&B neurons on DVS-CIFAR10 dataset of VGG7.

As discussed in Section II-D1, a IF&B neuron can be decoupled into two LIF neurons with the same input current and different threshold voltages. We, therefore, compared the IF&B Neuron with its equivalent decoupled LIF neuron trained from scratch, as shown in Table VII. The results directly indicate that most of the performance gains from IF&B neurons come from well-formulated spiking pattern design rather than from the higher computational costs.

F. Visualization of Neural Activity

The neural activity of VGG7 on CIFAR10 dataset with IF&B neurons at different layers is shown in Fig. 8. We randomly selected 50 neurons in each layer, with cyan color indicating that the neuron is at a regular spike state and dark cyan indicating that the neuron is at a burst spike state. Although neurons are rarely in burst mode, this biologically plausible neuron model is essential for the performance of SNNs.

IV. Conclusion

Inspired by the multi-spike delivery form of the brain, we design an efficient Leaky Integrate and Fire or Burst neuron model with triple-valued output from the perspective of network information capacity, while the burst density in IF&B can be adaptively adjusted. This multi-spike issuing form of synergistic neurons greatly enriches the characterization capability of the SNNs. Experimental results on static datasets CIFAR10, CIFAR100, and ImageNet show that we only need one simulation step to achieve a very high accuracy, which significantly reduces the latency of the SNNs. Also, we achieve state-of-the-art performance on the neuromorphic datasets DVS-CIFAR10 and NCALTECH101.

The proposed IF&B neuron model has the potential for real-world applications that require low-power, efficient, and robust neural computation. For example, neuromorphic hardware incorporating IF&B neurons could be considered for edge computing tasks such as real-time image recognition, where both low latency and energy efficiency are important. The adaptive burst mechanism may also offer benefits in scenarios needing biologically plausible signal processing, such as medical devices or robotics that involve sensory processing and decision-making. By achieving high accuracy with minimal simulation steps, our approach could contribute to the development of energy-efficient AI systems with practical applicability.

References

[1] G. Shen, D. Zhao, and Y. Zeng, "Backpropagation with biologically plausible spatiotemporal adjustment for training deep spiking neural networks," *Patterns*, vol. 3, no. 6, 2022, Art. no. 100522.

[2] D. Zhao, Y. Li, Y. Zeng, J. Wang, and Q. Zhang, "Spiking CapsNet: A spiking neural network with a biologically plausible routing rule between capsules," *Inf. Sci.*, vol. 610, pp. 1–13, 2022.

[3] T. Bu, W. Fang, J. Ding, P. Dai, Z. Yu, and T. Huang, "Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks," in *Proc. Int. Conf. Learn. Representations*, 2021.

[4] Y. Li, Y. Zeng, and D. Zhao, "BSNN: Towards faster and better conversion of artificial neural networks to spiking neural networks with bistable neurons," *Front. Neurosci.*, vol. 16, 2022, Art. no. 991851.

[5] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *Front. Neurosci.*, vol. 12, 2018, Art. no. 331.

[6] W. Zhang and P. Li, "Temporal spike sequence learning via backpropagation for deep spiking neural networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 12022–12033.

[7] Y. Kim and P. Panda, "Revisiting batch normalization for training low-latency deep spiking neural networks from scratch," *Front. Neurosci.*, vol. 15, 2021, Art. no. 773954.

[8] H. Zheng, Y. Wu, L. Deng, Y. Hu, and G. Li, "Going deeper with directly-trained larger spiking neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 12, pp. 11062–11070.

[9] Y. Wu, L. Deng, G. Li, J. Zhu, Y. Xie, and L. Shi, "Direct training for spiking neural networks: Faster, larger, better," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 01, pp. 1311–1318.

[10] M. Xu, Y. Wu, L. Deng, F. Liu, G. Li, and J. Pei, "Exploiting spiking dynamics with spatial-temporal feature normalization in graph learning," in *Proc. 13th Int. Joint Conf. Artif. Intell. (IJCAI-21)*, Z.-H. Zhou, Ed., Aug. 2021, pp. 3207–3213, doi: 10.24963/ijcai.2021/441.

[11] R.-J. Zhu et al., "TCJA-SNN: Temporal-channel joint attention for spiking neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–14, 2024, doi: 10.1109/TNNLS.2024.3377717.

[12] M. Yao et al., "Temporal-wise attention spiking neural networks for event streams classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10221–10230.

[13] M. Yao et al., "Attention spiking neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 9393–9410, 2023, doi: 10.1109/TPAMI.2023.3241201.

[14] D. Zhao, Y. Zeng, and Y. Li, "BackEISNN: A deep spiking neural network with adaptive self-feedback and balanced excitatory–inhibitory neurons," *Neural Netw.*, vol. 154, pp. 68–77, 2022.

[15] X. Cheng, Y. Hao, J. Xu, and B. Xu, "LISNN: Improving spiking neural networks with lateral interactions for robust object recognition," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, 2020, pp. 1519–1525.

[16] D. Beniaguev, I. Segev, and M. London, "Single cortical neurons as deep artificial neural networks," *Neuron*, vol. 109, no. 17, pp. 2727–2739, 2021.

[17] Y. Chen, H. Qu, M. Zhang, and Y. Wang, "Deep spiking neural network with neural oscillation and spike-phase information," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 8, pp. 7073–7080.

[18] X. Chen, Q. Yang, J. Wu, H. Li, and K. C. Tan, "A hybrid neural coding approach for pattern recognition with spiking neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 5, pp. 3064–3078, May 2024.

[19] K. Zhen, J. Sung, M. S. Lee, S. Beack, and M. Kim, "Scalable and efficient neural speech coding: A hybrid design," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 12–25, 2021.

[20] Q. Yu, J. Gao, J. Wei, J. Li, K. C. Tan, and T. Huang, "Improving multispike learning with plastic synaptic delays," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 10254–10265, Dec. 2023.

[21] P. Sun, L. Zhu, and D. Botteldooren, "Axonal delay as a short-term memory for feed forward deep spiking neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 8932–8936.

[22] W. Fang, Z. Yu, Y. Chen, T. Masquelier, T. Huang, and Y. Tian, "Incorporating learnable membrane time constant to enhance learning of spiking neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2661–2671.

[23] W. Zhang and P. Li, "Skip-connected self-recurrent spiking neural networks with joint intrinsic parameter and synaptic weight training," *Neural Comput.*, vol. 33, no. 7, pp. 1886–1913, 2021.

[24] G. Shen, D. Zhao, and Y. Zeng, "Exploiting nonlinear dendritic adaptive computation in training deep spiking neural networks," *Neural Netw.*, vol. 170, pp. 190–201, 2024.

[25] B. Yin, F. Corradi, and S. M. Bohté, "Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks," *Nature Mach. Intell.*, vol. 3, no. 10, pp. 905–913, 2021.

[26] N. Rathi and K. Roy, "DIET-SNN: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 6, pp. 3174–3182, Jun. 2023.

[27] Y. Chen, S. Zhang, S. Ren, and H. Qu, "Gradual surrogate gradient learning in deep spiking neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 8927–8931.

[28] Y. Li, Y. Guo, S. Zhang, S. Deng, Y. Hai, and S. Gu, "Differentiable spike: Rethinking gradient-descent for training spiking neural networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 23426–23439.

[29] Q. Yu, C. Ma, S. Song, G. Zhang, J. Dang, and K. C. Tan, "Constructing accurate and efficient deep spiking neural networks with double-threshold and augmented schemes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 4, pp. 1714–1726, Apr. 2022.

[30] J. C. Thiele, O. Bichler, and A. Dupret, "SpikeGrad: An ann-equivalent computation model for implementing backpropagation with spikes," in *Proc. Int. Conf. Learn. Representations*, 2020.

[31] Z. Wu, H. Zhang, Y. Lin, G. Li, M. Wang, and Y. Tang, "LIAF-Net: Leaky integrate and analog fire network for lightweight and efficient spatiotemporal information processing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6249–6262, Nov. 2022.

[32] E. M. Izhikevich, N. S. Desai, E. C. Walcott, and F. C. Hoppensteadt, "Bursts as a unit of neural information: Selective communication via resonance," *Trends Neurosc.*, vol. 26, no. 3, pp. 161–167, 2003.

[33] A. Kepecs and J. Lisman, "Information encoding and computation with spikes and bursts," *Netw.: Comput. Neural Syst.*, vol. 14, no. 1, 2003, Art. no. 103.

[34] S. Park, S. Kim, H. Choe, and S. Yoon, "Fast and efficient information transmission with burst spikes in deep spiking neural networks," in *Proc. 56th ACM/IEEE Des. Automat. Conf.*, 2019, pp. 1–6.

[35] Y. Li and Y. Zeng, "Efficient and accurate conversion of spiking neural network with burst spikes," in *Proc.31st Int. Joint Conf. Artif. Intell. (IJCAI-22)*, L. De Raedt, Ed., Jul. 2022, pp. 2485–2491, doi: 10.24963/ijcai.2022/345.

[36] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol.*, vol. 117, no. 4, 1952, Art. no. 500.
[37] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569–1572, Nov. 2003.
[38] P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge, MA, USA: MIT Press, 2005.
[39] J. E. Lisman, "Bursts as a unit of neural information: Making unreliable synapses reliable," *Trends Neurosc.*, vol. 20, no. 1, pp. 38–43, 1997.
[40] G. D. Smith, C. L. Cox, S. M. Sherman, and J. Rinzel, "Fourier analysis of sinusoidally driven thalamocortical relay neurons and a minimal integrate-and-fire-or-burst model," *J. Neuriophysiol.*, vol. 83, no. 1, pp. 588–610, 2000.
[41] E. M. Izhikevich, "Which model to use for cortical spiking neurons?," *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1063–1070, Sep. 2004.
[42] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," 2009 .
[43] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*.
[44] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
[45] H. Li, H. Liu, X. Ji, G. Li, and L. Shi, "CIFAR10-DVS: An event-stream dataset for object classification," *Front. Neurosci.*, vol. 11, 2017, Art. no. 309.
[46] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Front. Neurosci.*, vol. 9, 2015, Art. no. 437.
[47] Y. Zeng et al., "Braincog: A spiking neural network based brain-inspired cognitive intelligence engine for brain-inspired AI and brain simulation," *Patterns*, vol. 4, no. 8, Aug. 2023, Art. no. 100789.
[48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[50] W. Fang, Z. Yu, Y. Chen, T. Huang, T. Masquelier, and Y. Tian, "Deep residual learning in spiking neural networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 21056–21069.
[51] S. M. Bohte, "Error-backpropagation in networks of fractionally predictive spiking neurons," in *Proc. 21st Int. Conf. Artif. Neural Netw., Mach. Learn.*, Espoo, Finland, Jun. 2011, pp. 60–68.
[52] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
[53] N. Rathi, G. Srinivasan, P. Panda, and K. Roy, "Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation," in *Proc. Int. Conf. Learn. Representations*, 2019.
[54] N. Rathi and K. Roy, "DIET-SNN: Direct input encoding with leakage and threshold optimization in deep spiking neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 6, pp. 3174–3182, 2023, doi: 10.1109/TNNLS.2021.3111897.
[55] Y. Kim, Y. Li, H. Park, Y. Venkatesha, and P. Panda, "Neural architecture search for spiking neural networks," in *Proc. Eu. Conf. Comput. Vis. (ECCV)* 2022, pp. 36–56.
[56] B. Na, J. Mok, S. Park, D. Lee, H. Choe, and S. Yoon, "AutoSNN: Towards energy-efficient spiking neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2022, pp. 16253–16269.
[57] S. Deng, Y. Li, S. Zhang, and S. Gu, "Temporal efficient training of spiking neural network via gradient re-weighting," in *Proc. Int. Conf. Learn. Representations*, 2021.
[58] Y. Guo et al., "Recdis-SNN: Rectifying membrane potential distribution for directly training spiking neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 326–335.
[59] Y. Guo et al., "Ternary spike: Learning ternary spikes for spiking neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2024, vol. 38, no. 11, pp. 12244–12252.
[60] Z. Zhou et al., "SpikFormer: When spiking neural network meets transformer," in *Proc. 11th Int. Conf. Learn. Representations*, 2023.
[61] C. Zhou et al., "SpikingFormer: Spike-driven residual learning for transformer-based spiking neural network," 2023, *arXiv:2304.11954*.
[62] A. Kugele, T. Pfeil, M. Pfeiffer, and E. Chicca, "Efficient processing of spatio-temporal data streams with spiking neural networks," *Front. Neurosci.*, vol. 14, 2020, Art. no. 439.
[63] B. Ramesh, H. Yang, G. Orchard, N. A. Le Thi, S. Zhang, and C. Xiang, "DART: Distribution aware retinal transform for event-based cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 11, pp. 2767–2780, Nov. 2020.
[64] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31, pp. 6391–6401.

Guobin Shen (Student Member, IEEE) received the B.Eng. degree from Sun Yat-sen University, Guangzhou, Guangdong, China. He is currently working toward the Ph.D. degree with Brain-inspired Cognitive Intelligence Laboratory, Institute of Automation, Chinese Academy of Sciences, Beijing, China, under the supervision of Prof. Yi Zeng. His research interests include biologically-inspired learning algorithms and neural network architecture design and training strategies.

Dongcheng Zhao received the bachelor's degree from XiDian University, Xi'an, Shaanxi, China, in 2016, and the Ph.D. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2021. He is currently an Assistant Professor with the Brain-inspired Cognitive Intelligence Laboratory, Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing. His research interests include learning algorithms in spiking neural networks, thalamus-cortex interaction, and visual object tracking.

Yi Zeng is currently a Professor and the Director with Brain-inspired Cognitive Intelligence Laboratory, Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China. He is currently a Principal Investigator with the Key Laboratory of Brain Cognition and Brain-inspired Intelligence Technology, Chinese Academy of Sciences, China, and a Professor with the School of Artificial Intelligence, School of Future Technology, and School of Humanities, University of Chinese Academy of Sciences, China. He is also the Founding Director with the Center for Long-term AI and the Beijing Institute of AI Safety and Governance, China. His research interests include brain-inspired Artificial Intelligence, brain-inspired cognitive robotics, and ethics and governance of Artificial Intelligence.