Brain-Inspired Evolutionary Architectures for Spiking Neural Networks

Wenxuan Pan, Feifei Zhao, Zhuoya Zhao, and Yi Zeng

***Abstract*—The intricate and distinctive evolutionary topology of the human brain enables it to execute multiple cognitive tasks simultaneously, and this automated evolutionary process of biological networks motivates our investigation into efficient architecture optimization for spiking neural networks (SNNs). Diverging from traditional manual-designed and hierarchical network architecture search (NAS), we advance the evolution of SNN architecture by integrating local, brain region-inspired modular structures with global cross-module connectivity. Locally, the brain region-inspired module consists of multiple neural motifs with excitatory and inhibitory connections; globally, free connections among modules, including long-term cross-module feedforward and feedback connections are evolved. We introduce an efficient multiobjective evolutionary algorithm that leverages a few-shot predictor, endowing SNNs with high performance and low energy consumption. Extensive experiments across both static (CIFAR10, CIFAR100) and neuromorphic (CIFAR10-DVS, DVS128-Gesture) datasets reveal that the proposed model significantly exhibits robustness while maintaining consistent and exceptional performance. This study pioneers in searching for optimal neural architectures for SNNs by integrating the human brain's advanced connectivity and modular organization into SNN optimization, thereby contributing valuable perspectives to the development of brain-inspired artificial intelligence.**

***Impact Statement*—NAS aims to automate the search for optimal architectures, offering alternatives to handcrafted designs. However, existing works of NAS for SNN lack deep reference to the brain topology properties, leading to suboptimal performance. Here, we propose an innovative evolutionary brain-inspired neural architecture search (EB-NAS) to optimize SNN architectures. Different from traditional manually designed hierarchical networks, EB-NAS integrates multiple evolvable self-organizing modules that reflect brain regions, consisting of neural motifs with both excitatory and inhibitory connections. Supported by an efficient few-shot predictor, EB-NAS demonstrates consistently outstanding performance across various datasets, setting a new benchmark against existing SNN architecture optimization.**

***Index Terms*—Biological module and long-term connection, brain-inspired spiking neural networks, evolutionary neural architecture search, neural circuit motifs.**

I. Introduction

AFTER millions of years of evolution, neurons with complex information processing capabilities and microcircuits with specific functions have emerged in the human brain, empowering it to be a powerful computational device. As the third generation of neural networks, spiking neural network (SNN) [1] realizes a low energy consumption and high-efficiency computing paradigm by simulating the characteristics of neuron communication in the brain. However, current SNN research often relies on fixed architectures that do not adequately incorporate brain-inspired topological features, which restricts their multitasking capabilities.

Neural architecture search (NAS) attempts to automate the design process traditionally dependent on human experience, with some approaches surpassing human-designed architectures in specific tasks [2], [3]. Evolutionary neural architecture search (ENAS), as an optimization strategy within NAS, stands out for its robustness against local optima and its reduced computational resource requirements compared to reinforcement learning (RL)-based methods [4]. Leveraging the present understanding of the brain's evolutionary architecture, this article endeavours to investigate the development of brain-inspired local and global neural circuits to refine SNN architectures and functions.

In ENAS, due to the large population and numerous iterations, training SNN models individually to evaluate each individual's performance becomes highly time-consuming. Therefore, efficient evaluation methods are particularly important, and research ideas are usually inseparable from weight-sharing [5], performance predictor [6], [7], or zero-shot methods [8]. The weight-sharing approach, known as the one-shot method, requires training only a single supernet throughout the algorithmic process. However, training to

Manuscript received 10 March 2024; revised 11 May 2024; accepted 26 May 2024. Date of publication 31 May 2024; date of current version 12 November 2024. This work was supported by the Key Research Program of Frontier Sciences, Chinese Academy of Sciences under Grant ZDBS-LY-JSC013, in part by the Postdoctoral Fellowship Program of CPSF under Grant GZC20232994, and in part by the National Natural Science Foundation of China under Grant 62106261. This article was recommended for publication by Associate Editor Junhua Li upon evaluation of the reviewers' comments. *(Wenxuan Pan and Feifei Zhao contributed are co-first authors.) (Corresponding author: Yi Zeng.)*

Wenxuan Pan is with the Brain-inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China.

Feifei Zhao is with the Brain-inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.

Zhuoya Zhao 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 101408, China.

Yi Zeng is with the Brain-inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, also with the Center for Long-term Artificial Intelligence, Beijing 100190, China, also with the University of Chinese Academy of Sciences, Beijing 100083, China, and also with the Key Laboratory of Brain Cognition and Brain-inspired Intelligence Technology, Chinese Academy of Sciences, Shanghai 200031, China (e-mail: yi.zeng@ia.ac.cn).

Digital Object Identifier 10.1109/TAI.2024.3407033

convergence takes a long time, and it may encounter the problem of performance collapse, which could not reflect real network fitness. Despite this efficiency, reaching convergence can be time-intensive and take a risk of performance collapse, potentially failing to accurately reflect the true network fitness. Zero-cost evaluation methods, which forego training, offer significant time savings. However, their reliance on merely approximating attributes that correlate with architectural performance could result in inaccurate fitness rankings. As a few-shot method, performance predictor trains a regression model for directly predicting the performance of unknown architectures based on the information of some trained architectures [6], [7], [9], [10] and can be divided into online and offline according to whether the predictor is updated during the NAS process. Offline predictors sample a limited set of architectures only once and do not incorporate new information after construction, limiting their adaptability. Therefore, the online predictor is more practical and flexible, capable of continually enhancing evaluation accuracy by integrating and updating with newly acquired knowledge [11].

A large amount of ENAS research is based on deep neural networks (DNNs), and existing works on evolving SNN architecture are still very limited. Na et al. [12] propose AutoSNN to evolve a more energy-efficient SNN architecture, employing an artificially determined weight factor to optimize both the number of spikes and classification accuracy, essentially reducing it to a single-objective optimization. This approach fails to fully address the coordination between these two contradictory evolutionary goals. Kim et al. [13] apply a zero-shot approach to evaluate the classification performance of SNNs, greatly reducing the time and computational cost. In addition, from the perspective of search space, Na et al. follow the topological characteristics of VGG-Net and ResNet, searching for the stacking method of spiking convolution and residual blocks. Kim et al. consider the cell-based search space with only one global backward connection. Configuring the search space based on traditional DNN practices fails to leverage the potential of brain-inspired structures or modules, subsequently constraining the evolutionary algorithm's search efficacy. Originating from an exploration of the computing nature of the brain, SNNs prompt us to explore evolutionary SNN architectures from a standpoint that offers greater biological interpretability.

After millions of years of evolution, the synaptic connection patterns between nearly 100 billion different excitatory and inhibitory neurons in the human brain have formed diverse circuit motifs and modular brain regions [14], [15]. Neuroscientific findings inspire the design of a search space tailored for SNNs, drawing from a brain-inspired perspective. Locally, each module incorporates five common types of circuit motifs found in different brain regions. Globally, nonhierarchical long-term connections including cross-module or feedback connections are evolved between modules, resembling the complex connections between different brain regions. To expedite the evolutionary process, we utilize a few-shot multiobjective evolutionary algorithm, facilitating the population's evolution toward low energy consumption and high efficiency without the need to sample and train a large number of candidates.

Overall, the proposed EB-NAS is designed to expedite multiobjective evolutionary algorithms using an online performance predictor, facilitating the automatic designing of low-energy, high-efficiency SNN models that are combined with brain-inspired topological properties. All contributions can be divided into the following points.

1) Our approach to evolving SNN architectures draws inspiration from brain-region modules composed of neural motifs, coupled with the flexibility of long-term connections between these modules (such as cross-module and feedback connections), allowing SNNs to autonomously evolve into architectures that are more closely to the biological intricacies of the brain.
2) We deploy a multiobjective evolutionary algorithm focused on minimizing fired spikes while maximizing performance. To alleviate the time-consuming evolution process, a time-efficient few-shot evaluation method (an online performance predictor) is developed.
3) On a variety of datasets, the proposed EB-NAS achieves superior performance with significantly lower energy consumption. The thorough analysis also proves the effectiveness of drawing on the naturally evolved neural topology of the human brain for evolving SNN architectures.

II. Related Work

NAS is a method for automatically designing architectures [16], [17], [18] based on human experts' experience, which is more efficient than manually designed fixed-architectures. Existing NAS research can be roughly divided into three categories: RL-based methods, gradient-based methods, and evolution-based methods. RL-based methods consume a lot of computing resources [4], [19] and require an additional controller to help search. Gradient-based methods require the prior construction of a supernet, limiting their automation and frequently resulting in ill-conditioned architectures. ENAS is a computing paradigm that simulates species evolution or population behavior. Candidate architectures will be initialized and encoded, then optimized and iterated through evolutionary computing (EC) methods such as genetic algorithms (GAs) [20], particle swarm optimization (PSO) [21], evolutionary strategy (ES) [22], and differential evolution (DE) [23] until the fitness of the population is gradually evolved to obtain a set of high-quality solutions. ENAS is more computationally resource-efficient than RL-based NAS, requiring less prior knowledge than gradient-based NAS, and is well-suited for solving complex multiobjective optimization problems [11], [24], and [25].

A. Multiobjective ENAS

Many ENAS algorithms aim to optimize both performance and cost [25], [26], [27], [28], presenting challenges due to the conflicting objectives. Common optimization methods for dealing with multiobjectives can be divided into scalarization and population-based methods [11]. Scalarization methods [12],

[29], [30], [31], [32] combine multiple objectives into a single function with fixed weight factors, but may not accurately reflect the tradeoffs between different goals. Na et al. [12] consider both the number of spikes (evaluating SNN energy consumption) and classification accuracy with scalarized fitness function. The problem with such methods is that the predetermined weighting of different objectives, may not accurately reflect their relative importance and fail to consider the potential conflict and necessary tradeoffs between these evolving objectives. On the other hand, population-based approaches [27], [33], [34] employ heuristic search strategies, offering a more comprehensive consideration of each objective and a better approach to the Pareto optimal front for multiobjective optimization.

B. Efficient Evaluation

During evolution, evaluating the current individual's performance is crucial for guiding the optimization of the next generation. However, training a large neural network from scratch to convergence can consume many graphic processing unit (GPU) days [2], [18], [35], making evaluation the most time-intensive phase of evolution. Consequently, reducing evaluation time has become a focal point of numerous studies. Efficient evaluation methods, depending on the number of architectures requiring training, can be categorized into three types: few-shot, one-shot, and zero-shot approaches.

One-shot methods train a supernet encompassing all candidate architectures, while less time-consuming than training each model individually, still requires substantial time to converge and can encounter multimodel forgetting problems [36]. Even with reduced computation time compared to training each candidate from scratch, the accuracy of the supernet's predictive capabilities is not necessarily reliable. Without training any architectures, zero-shot methods propose many architecture-related indicators from the perspective of neural network theory, such as learnability [8], generalization [37], and expressiveness [38]. However, these abstract metrics might not accurately capture the candidates' true performance, raising concerns about their prediction [9].

Performance predictor [6], [7], [9], [10], a few-shot method, samples and trains a limited subset of candidates to gather architecture data and their corresponding performances, forming a regression model capable of forecasting the performance of untested architectures. An online predictor undergoes automatic updates throughout its construction and training phases. Liu et al. [39] train the predictor starting with the simplest shallow models, progressing to complex structures, and culling bad structures. The process incorporates promising architectures as new inputs, ensuring the predictor's state is constantly refined and updated. Lu et al. [11] take a similar approach to train adaptive switching surrogate predictors. Differ from offline predictors, these research enhance predictive accuracy over time by integrating new samples of architecture and performance data.

C. Search Space

Besides increasing the speed of evaluation, researchers also streamline the search space to improve efficiency. Block is proposed as a new topology, where the layers inside it have specific connections mode and functions, such as ResBlock [40], DenseBlock [41], ConvBlock [42], and InceptionBlock [43]. The cell-based search space can be viewed as a specific instance of a block-based architecture, where each block is identical, yet the connections among layers within each block are allowed to vary freely [3], [44], [45]. Existing works of searching the SNN architecture have not made more innovations in the search space. Na et al. [12] draw on the experience of ConvBlock and ResBlock to find the best combination of them. Kim et al. [13] follow the cell-based search space, but only focus on one backward connection, making the search capability limited. However, there is a notable lack of in-depth consideration of brain topology characteristics, leaving room for capability enhancements.

Addressing the limitations of NAS in SNNs highlighted above, this article adopts a brain-inspired approach to refine both the search space and the evaluation methodology. We introduce a more efficient and biologically interpretable search space, capable of evolving modules that encompass five prevalent circuit motifs and their long-term global interconnections. Furthermore, a few-shot online performance predictor is constructed to speed up the population-based multiobjective evolution process, ensuring efficiency without compromising accuracy.

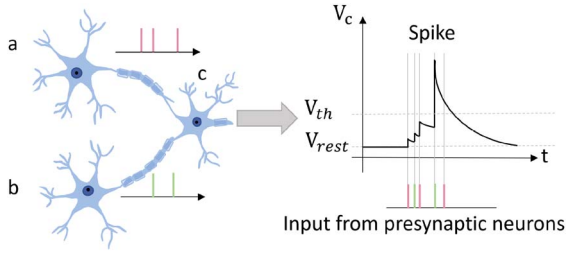

Fig. 1. Computation of LIF neurons. Neuron c receives spikes from presynaptic neurons a and b, and the potential V_c accumulates continuously from the resting potential V_{rest} until it reaches the threshold V_{th}, fires a spike and returns to the resting potential.

III. Methods

A. SNN Foundation

In SNNs, spikes carrying information are transmitted from presynaptic to postsynaptic neurons. A postsynaptic neuron fires when the accumulated potential, triggered by the presynaptic neuron's firing, reaches a certain threshold. Here, we adopt a commonly used leaky integrate-and-fire (LIF) neuron model [46]. The dynamic process of neuron i membrane potential $V_i(t)$ changes with time t can be described as (1) and Fig. 1.

$$\tau_m \frac{\mathrm{d}V_i(t)}{\mathrm{d}t} = -\left[V_i(t) - V_{\text{rest}}\right] + I_i(t) \tag{1}$$

$$h(V_i(t)) = \begin{cases} 1, V_i(t) \geq V_{\text{th}} \\ 0, V_i(t) < V_{\text{th}} \end{cases} \tag{2}$$

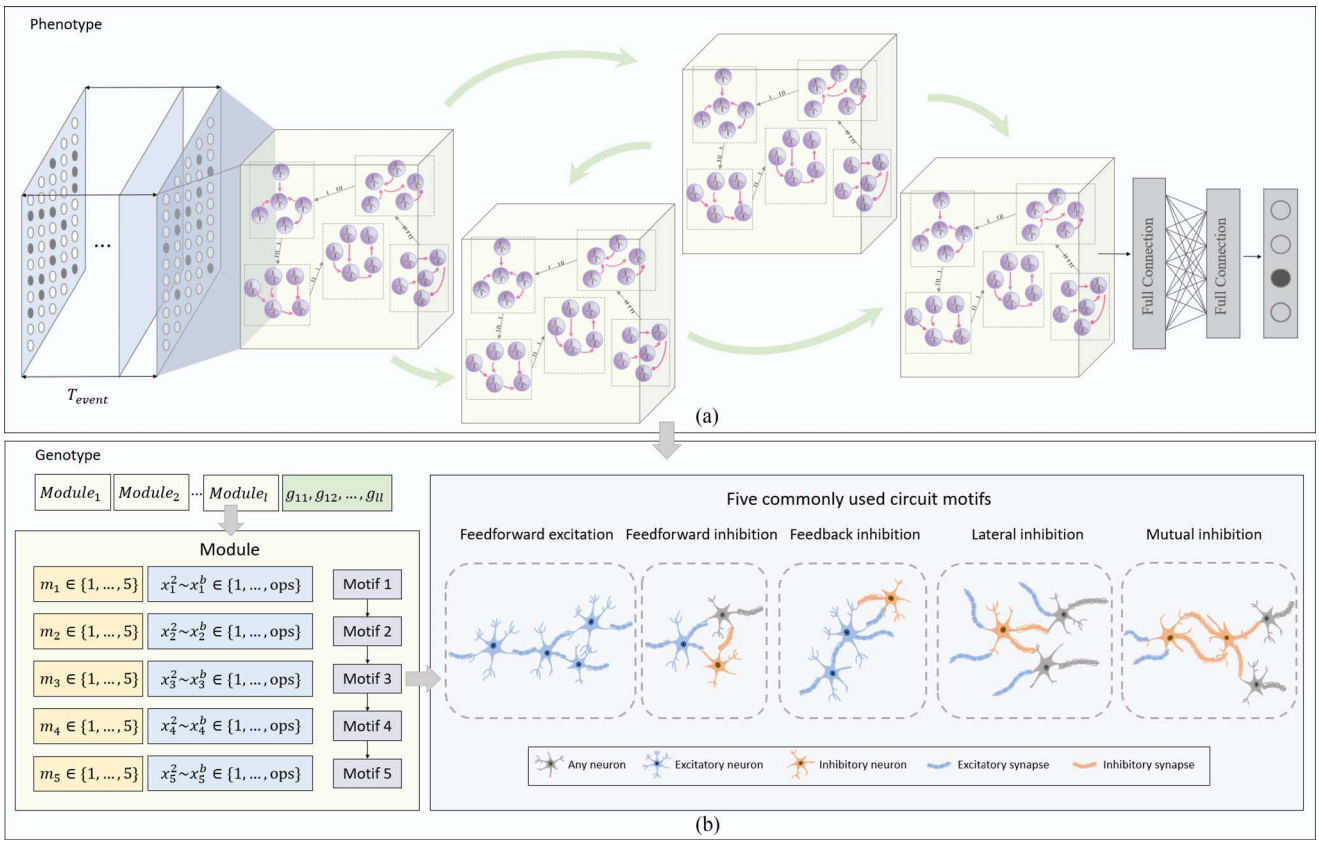

Fig. 2. Phenotype-to-genotype mapping and encoding schemes. (a) Entire network is composed of l modules, each of which contains five neural circuits. Globally, the modules are freely connected without self-loop. The number of input features of the last fully connected layer is the number of features of the previous layer, and the number of output features is 10*the number of classes in the classification problem. (b) Encoding scheme for the model.

where $\tau_m = \mathrm{RC}$ is the membrane potential time constant (R is the membrane resistance and C is the membrane capacitance). The input current $I_i(t)$ leads to the accumulation of membrane potential $V_i(t)$. In (2), $h(V_i(t))$ is a function indicating the firing of neuron i. When the threshold V_{th} is reached, the neuron fires and the potential returns to V_{rest}. This calculation process is iterated over a specific duration, during which the cumulative number of spikes gathered by the final layer of neurons directly influences the model's output.

Specifically, take a convolutional neural network as an example. The first convolutional layer takes the image X as input and applies a set of filters (or kernels) to extract features, generating a feature map Y. Y is then fed into the spiking mechanism of the SNN, where each feature value is considered as the membrane potential of a neuron. The spike outputs from the first layer, denoted as H, where $H = h(Y)$ as calculated in (2), can be input to a second convolutional layer.

The spike transmission process described in (2) is nondifferentiable, which limits the optimization of network weights by backpropagation. Thus, we train SNNs with a surrogate gradient [47] as follows:

$$\frac{\partial h(V_i(t))}{\partial V_i(t)} = \begin{cases} 1 & \text{if } -\frac{1}{\alpha} \le V_i(t) - V_{\text{th}} \le \frac{1}{\alpha} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where α (set to 2) is a parameter that determines the steepness of the curve. The spiking neuron model and training algorithm used in this work are implemented based on the brain-inspired cognitive intelligence engine (BrainCog) framework [48].

B. Encoding Brian-Inspired Neural Circuit

The proposed brain-inspired search space for evolving SNN architectures includes local modular structures consisting of five neural motifs, along with global free long-term connectivity between modules. Common neural circuit motifs found in the human brain are feedforward excitation (FE), feedforward inhibition (FI), feedback inhibition (FbI), lateral inhibition (LI), and mutual inhibition (MI). For example, FE and FbI are related to multisensory fusion [15] and enhancing the signal-to-noise ratio [14]. FI helps the network maintain a balance of excitatory and inhibitory [49]. LI plays a role in amplifying the perception of various sensory stimuli [50] and MI is linked to the rhythmic activity of neural networks [51], [52]. We implement them with spiking neurons as shown in Fig. 2(b). As shown in Fig. 2(a), the combinations of motifs within different modules vary, reflecting the diverse structures found in diverse brain regions. Global connections imitate the way brain regions are connected, evolving for a nonhierarchical, freely connected approach that does not limit connection direction or module crossing. The

TABLE I
DETAILED DESCRIPTION OF THE MEANINGS OF DIFFERENT GENE VALUES IN A GENOME

Gene	Value	Description
g_{ij}	0	No connection from K_i to K_j
	1	One connection from K_i to K_j
m_i	1	Feedforward excitation
	2	Feedforward inhibition
	3	Feedback inhibition
	4	Lateral inhibition
	5	Mutual inhibition
$x_i^2 \sim x_i^b$	1	3×3 conv
	2	5×5 conv

encoded SNN architecture can be expressed as

$$[K_1, K_2, K_3, ..., K_l, g_{11}, g_{12}, ..., g_{ll}] \tag{4}$$

where a single module K_i consisting of five motifs is encoded as

$$[m_1, x_1^2, ..., x_1^b, m_2, x_2^2, ..., x_2^b, ..., m_5, x_5^2, ..., x_5^b] \tag{5}$$

where each motif is coded as $(m_i, x_i^2, ..., x_i^b)$. m_i indicates which kind of circuit motif it is, while x_i^2 to x_i^b encode the computation inside the motif. x_i^2 to x_i^b have ops possible values, and each represents one kind of operation. b is the maximum number of genes computed internally within a motif and set to 20. Two types of convolution operations: 3 x 3 convolution and 5 x 5 convolution are chosen to evolve. $[g_{11}, g_{12}, ..., g_{ll}]$ are global parameters indicating the connection mode between l modules. For example, g_{ij} indicates whether there is a connection from K_i to K_j. To prevent self-loop, we limit g_{ii} to 0. See Table I for a detailed description of different gene values in a genome.

C. ENAS for SNN

1) Multiobjective Optimization: In SNNs, improving accuracy frequently leads to increased energy loss, manifested by a higher number of spikes. To identify an SNN architecture that balances lower energy consumption with greater efficiency, this article considers both validation error and the number of spikes as optimization objectives. The multiobjective optimization problem (MOP) can be articulated as follows:

$$\underset{M \in \Omega}{\operatorname{argmin}} \boldsymbol{F}(M) = \{f_1(M), f_2(M)\}. \tag{6}$$

The individual M must be within the specified search space Ω consisting of all genomes encoded based on (4) and Table I. The first objective is to minimize the verification error of the model M on a given classification dataset

$$f_1(M) = \text{minL}\left(M, \mathcal{D}_{\text{tst}}, \mathcal{D}_{\text{trn}}\right) \tag{7}$$

where $\text{L}(*, \mathcal{D}_{\text{tst}}, \mathcal{D}_{\text{trn}})$ represents the verification error of the model M trained on $\mathcal{D}_{\text{trn}}$ on the validation dataset $\mathcal{D}_{\text{tst}}$.

Suppose the model M consists of l modules, then the second evolutionary goal can be expressed as

$$f_2(M) = \sum_{i=0}^{l} S(K_i) \tag{8}$$

where $S(K_i)$ indicates the number of spikes fired by all spiking neurons in K_i.

2) Efficient Evaluation: Our goal is to identify nondominated architectures of the entire search space Ω. However, evaluating all architectures during the evolution process is impractical due to their large number and the significant time required [4]. Therefore, we build an online performance predictor Predictor based on Set_1: Predictor is a classification and regression trees (CART) [6] surrogate model that can quickly predict the validation error of unknown architectures based on the existing knowledge set Set_1, thereby circumventing the need to train all explored architectures and enhancing the efficiency of the evolution.

At initialization, an initial population P_0 of size n_0 is constructed. Each individual in P_0 can be trained for e_{eval} epochs to obtain a corresponding verification error through f_1, which is stored in Set_1. To make the search of the evolutionary algorithm more directional, we take the following approach: in t_{th} iteration, a new population of size n_{new} in Ω is sampled, which can be evaluated by Predictor. According to Predictor and f_2, the fitness of f_1 and f_2 of all individuals can be obtained, respectively, as the input of the Nondominated Sorting Genetic Algorithms (NSGA-II [53]). The introduction of elite preservation ensures the best solutions that can generate the next generation based on the calculation of crowding distance and nondominated sorting. Two-point crossover and polynomial mutation operators are employed to evolve the selected individuals. After gens generations, the nondominated sorted population (denoted as O) is obtained, which can be expressed as

$$O = \text{NSGA-II}(\text{Predictor}, f_2, \text{gens}). \tag{9}$$

Each of these n_s outstanding candidates will be trained for e_{eval} epochs, and the validation error is added to Set_1, enriching the knowledge for training Predictor.

As the iterations proceed and more and more information in Set_1, Predictor is trained in online mode to make it accurate. Therefore, the found O will get closer and closer to true nondominated solutions in ω. The process demonstrated above is repeated until $t =$ iters. The nondominated individual $\text{Net}_{\text{evolved}}$ with the lowest loss in O is selected and trained e_{trn} epoch to get the top-1 accuracy, as shown in Algorithm 1 and Fig. 3. All parameter values involved in our model are shown in Table II.

IV. EXPERIMENTS AND ANALYSIS

In this section, we conduct the experiments on static CIFAR10 [54], CIFAR100 [55] and neuromorphic DVS128-Gesture [56], and CIFAR10-DVS [57] datasets to illustrate the superiority of our proposed model. The neuromorphic dataset CIFAR10-DVS is an event stream-based dataset converted from the frame-based static dataset CIFAR10 by a dynamic vision sensor (DVS). DVS128-Gesture is a hand gesture dataset, comprising a combination of various lights and gestures, captured by a DVS over about 6 s. We compare the classification accuracy and energy consumption of EB-NAS against several other methods.

Algorithm 1: The procedure of EB-NAS algorithm.

input : initial population size n_0, number of iterations to search $iters$, number of generations to evolve in NSGA-II $gens$, training dataset $\mathcal{D}_{\text{trn}}$, validation dataset $\mathcal{D}_{\text{tst}}$, search space Ω

output: the evolved SNN architecture $Net_{evolved}$

$P_0 \leftarrow$ Initialize a population of size n_0 in Ω;
$Set_1 \leftarrow \emptyset$;
while $i < n_0$ **do**
  $loss \leftarrow f_1(P_0[i], \mathcal{D}_{\text{trn}}, \mathcal{D}_{\text{tst}})$
$Set_1 \cup (P_0[i], loss)$;
while $t < iters$ **do**
  $Predictor \leftarrow$ UpdatePredictor(Set_1);
  $O \leftarrow$ NSGA-II($Predictor, f_2, gens$);
  for $j \leftarrow 0$ **to** n_s **do**
    $loss \leftarrow f_1(O[j], \mathcal{D}_{\text{trn}}, \mathcal{D}_{\text{tst}})$
  $Set_1 \cup (O[j], loss)$;
Select the non-dominated SNN architecture $Net_{evolved}$ with the lowest loss in O ;
return $Net_{evolved}$

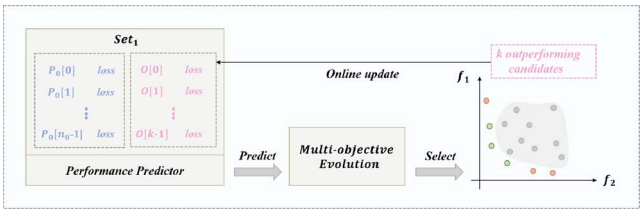

Fig. 3. Multiobjective evolutionary process with the help of the performance predictor. Blue represents the initial population and pink indicates the selected n_s outperforming candidates after nondominated sorted, which will be trained to obtain real loss and used to further enrich the knowledge of the predictor.

TABLE II
VALUE OF ALL PARAMETERS IN OUR MODEL

Parameter	Description	Value
τ	Membrane potential time constant	2.0
V_{rest}	Resting potential of neurons	0
V_{th}	Firing threshold of neurons	0.5
l	Maximum number of modules	4
n_0	Initial population size	300
iters	Number of iterations	50
n_{new}	Size of newly generated population in NSGA-II	60
gens	Number of generations evolved in NSGA-II	40
e_{eval}	Number of epochs required for evaluation	10
e_{trn}	Number of epochs required for training	600
n_s	Number of candidates selected in each iteration	10

A. Comparative Results

Table III shows the comparison results between EB-NAS and other SNN models on static datasets CIFAR10 and CIFAR100. On CIFAR10, compared with fixed-architecture SNN models, the classification accuracy of EB-NAS exceeds the best-performing ResNet-18 architecture [69] by 1.06%. The performance of EB-NAS is also significantly better than models based on ResNet, VGG-Net, and other convolutional structures, ranging from 1.22% to 5.83%. Compared to existing SNN-based NAS models [12], [13] trained with STBP, the classification accuracy of EB-NAS is 2.51% and 2.93% higher, respectively. On CIFAR100, EB-NAS outperforms the best fixed-architecture SNN model [63] by at least 6.88%. Compared with other SNN-based NAS models [12], [13], our method improves the classification accuracy by 10.05% and 6.17%, respectively. Among all the models listed in Table III, EB-NAS has the shortest simulation time (4), which significantly improves the computational efficiency of the model.

Table IV shows the comparison results between our model and other SNN models on neuromorphic datasets CIFAR10-DVS and DVS128-Gesture. On CIFAR10-DVS, the classification accuracy of our model exceeds that of fixed-structure models including DensNet and VGG-Net, ranging from 0.31% to 16.68%. Meanwhile, EB-NAS significantly outperforms the SNN-based NAS model [12] by 10.98%. Similar conclusions can also be observed on DVS128-Gesture: whether it is a fixed

TABLE III
COMPARISON OF CLASSIFICATION PERFORMANCE ON STATIC DATASETS CIFAR10 AND CIFAR100

Dataset	Model	Training Methods	Architecture	Timesteps	Accuracy (%)
CIFAR10	Wu et al. [58]	STBP NeuNorm	CIFARNet	12	90.53
	Zhang and Li [59]	TSSL-BP	CIFARNet	5	91.41
	Rathi et al. [60]	Hybrid	ResNet-20	250	92.22
	Zhu et al. [61]	Event-driven BP	ResNet-14	-	92.45
	Rathi and Roy [62]	Diet-SNN	ResNet-20	10	92.54
	Li et al. [63]	ANN-SNN conversion	VGG16	32	93.0
	Zheng et al. [64]	STBP-tdBN	ResNet-19	6	93.16
	Fang et al. [65]	STBP	6Conv, 2Linear	8	93.5
	Han et al. [66]	ANN-SNN conversion	VGG16	2048	93.63
	Xiao et al. [67]	OTTT	VGG	6	93.73
	Deng et al. [68]	TET	ResNet-19	4	94.44
	Yang and Chen [69]	Spike representation	ResNet-18	-	94.6
	Yang et al. [70]	SIBoLS	ResNet-18	-	94.65
	Kim et al. [13]	STBP	NAS	5	92.73
	Na et al. [12]	STBP	NAS	16	93.15
	Our Method	STBP	EB-NAS	**4**	**95.66**
CIFAR100	Zhu et al. [61]	Event-driven BP	VGG-11	-	63.97
	Rathi and Roy [62]	Diet-SNN	ResNet-20	5	64.07
	Rathi et al. [60]	Hybrid	VGG11	125	67.87
	Garg et al. [71]	DCT	VGG9	48	68.3
	Park et al. [72]	TTFS	VGG15	680	68.8
	Han et al. [66]	ANN-SNN conversion	VGG16	2048	70.9
	Xiao et al. [67]	OTTT	VGG	6	71.11
	Guo et al. [73]	Real spike	VGG16	10	71.17
	Li et al. [63]	ANN-SNN conversion	ResNet-20	16	72.33
	Li et al. [74]	Dspike	ResNet18	6	74.24
	Yang et al. [70]	SIBoLS	ResNet-18	-	76.31
	Yang and Chen [69]	Spike representation	ResNet-18	-	77.4
	Na et al. [12]	STBP	NAS	16	69.16
	Kim et al. [13]	STBP	NAS	5	73.04
	Our Method	STBP	EB-NAS	**4**	**79.21**

Note: The bold part is to highlight the superiority of our method.

TABLE IV
COMPARISON OF CLASSIFICATION PERFORMANCE ON NEUROMORPHIC DATASETS CIFAR10-DVS AND DVS128-GESTURE

Dataset	Model	Training Methods	Architecture	Timesteps	Accuracy (%)
CIFAR10-DVS	Kugele et al. [75]	Streaming rollout	DenseNet	10	66.80
	Zheng et al. [64]	STBP-tdBN	ResNet-19	10	67.8
	Wu et al. [76]	BPTT	LIAF-Net	10	70.4
	Xiao et al. [67]	OTTT	VGG	10	77.1
	Guo et al. [73]	Real spike	ResNet-20	10	78.0
	Yang and Chen [69]	Spike representation	VGG11	-	78.2
	Deng et al. [68]	TET	VGGSNN	10	83.17
	Na et al. [12]	STBP	NAS	16	72.5
	Our Method	STBP	EB-NAS	**10**	**83.48**
DVS128-Gesture	Xing et al. [77]	SLAYER	5-Layer-CNN	20	92.01
	Shrestha and Orchard [78]	SLAYER	16-layer-CNN	300	93.64
	Fang et al. [65]	STBP	5Conv, 2Linear	20	97.57
	He et al. [40]	STBP	7B-Net	16	97.92
	Na et al. [12]	STBP	NAS	16	96.53
	Our Method	STBP	EB-NAS	**10**	**98.08**

Note: The bold part is to highlight the superiority of our method.

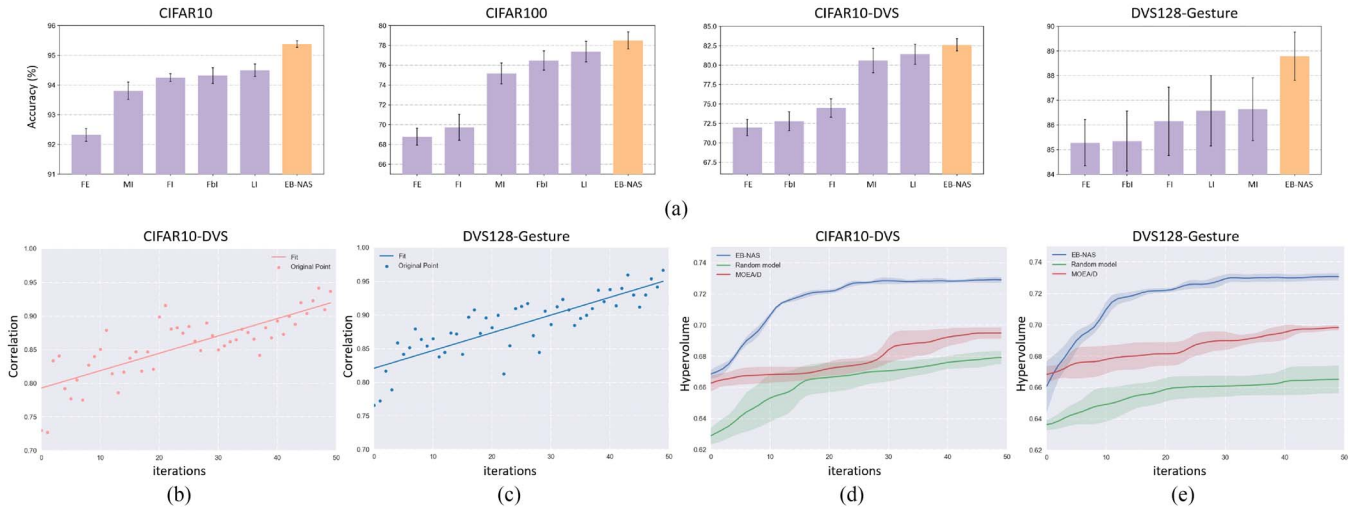

Fig. 4. Ablation results. (a) Evolutionary performance comparison of single-motif models and EB-NAS. (b) Accuracy of the predictor on CIFAR10-DVS in iteration. (c) Accuracy of the predictor on DVS128-Gesture in iteration. (d) and (e) Changes of hypervolume of two evolutionary models on CIFAR10-DVS and DVS128-Gesture.

structure or a structure searched by NAS, EB-NAS achieves better classification accuracy with the shortest timesteps (10). By contrasting EB-NAS with both fixed-structure SNNs and NAS-based SNNs across various datasets, we observe that current NAS-based SNNs often struggle to outperform architectures with fixed designs. EB-NAS overcomes this challenge, not only surpassing existing NAS-based SNN models in terms of classification performance but also surpassing those that are manually designed. This achievement not only demonstrates the superior performance of EB-NAS, but also marks a breakthrough in SNN architecture design that transcends the limitations of human expertise, emphasizing its importance to the field of NAS for SNN.

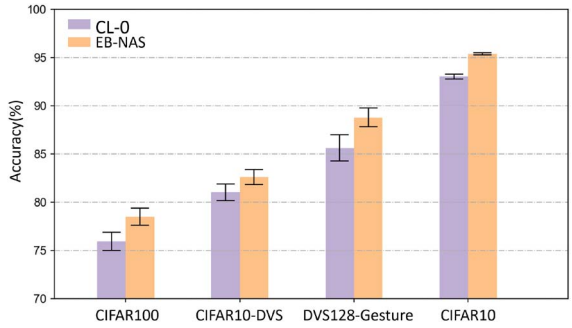

Fig. 5. Evolutionary performance comparison of CL-0 and EB-NAS. All candidates denoted by CL-0 are fixed to the hierarchical connection structure.

B. Ablation Study

The brain-inspired search space tailored for SNN, as proposed in this article, introduces two innovative aspects: 1) the use of a combination of five functional microcircuits, identified in the brain, as the foundational module; and 2) the allowance for global and unrestricted connectivity among modules, moving beyond mere hierarchical connections. Ablation studies will be conducted to verify these two aspects. Aside from the variables explored in the ablation experiments, the parameters and methodology of the evolutionary algorithm will remain unchanged. We conduct the ablation experiments on four datasets CIFAR10, CIFAR100, CIFAR10-DVS, and DVS128-Gesture, and the initial channels are set to 48, 48, 32, and 32, respectively.

1) Combination of Motifs: To verify the effectiveness of the motif combinations, we compare EB-NAS with models of fixed motif types as shown in Fig. 4(a). In each single-motif model, the motifs in all modules are specified as one of the five categories. In this way, five comparison models can be obtained, denoted as FE, FI, FbI, LI, and MI in Fig. 4(a). As depicted in Fig. 4(a), across the datasets CIFAR10, CIFAR100, CIFAR10-DVS, and DVS128-Gesture, FE exhibits the worst classification ability, likely a consequence of excessive excitatory signaling. Conversely, LI shows comparatively better performance, possibly attributed to its internal balance between excitatory and inhibitory characteristics. Fig. 4(a) also highlights EB-NAS's superior classification accuracy compared to single-motif models. Across all datasets, EB-NAS's classification accuracy surpasses the best single-motif model by 0.88%, 1.13%, 1.18%, and 2.15% (averaged across multiple trials), respectively. Moreover, EB-NAS exhibits the least variance on CIFAR10 and CIFAR10-DVS datasets.

2) Global Connections: We validate the role of evolvable global free long-term connections on four datasets through two models: 1) the proposed method marked as EB-NAS; and 2) the hierarchical connected model marked as CL-0. The ablation results are shown in Fig. 5. It can be seen that the model evolved by the free global connection EB-NAS is 2.35%, 2.54%, 1.58%, and 3.17% higher, respectively than the traditional forward connection CL-0 model on CIFAR10, CIFAR100, CIFAR10-DVS, and DVS128-Gesture datasets. The incorporation of free global connections broadens the search space beyond the traditional hierarchically connected models, offering a wider array

of possibilities and diversifying the types of individuals that the evolutionary algorithm can explore. By emulating the connectivity found among brain regions, this mode of connection introduces more features to each module, thereby enhancing the computational capabilities of the entire model.

C. Effects of Performance Predictor

The performance predictor trains a regression model capable of predicting the accuracy of unknown individuals based on the information of known individuals and the associated classification accuracy, which greatly reduces the time required for NSGA-II: in each iteration, all $n_{\text{new}} * \text{gens} = 2400$ individuals originally needed to be trained for e_{eval} epochs. With the help of the predictor, only iters $* k = 500$ nondominated models need to be trained, and the algorithm speed is 4.8x faster than the model without a predictor. To showcase the effectiveness of the predictor, we plot the correlation between the loss predicted by the predictor and the actual loss during iterations on the CIFAR10-DVS and DVS128-Gesture datasets, as Fig. 4(b) and 4(c) shown. In Fig. 4(b) and 4(c), the y-axis is the Spearman correlation coefficient [79] between the prediction result of the predictor and the true performance of individuals. The larger the coefficient, the more accurate the prediction. The linear fit to the scatter points suggests that the predictor's accuracy on both datasets improves as the iterations progress.

D. Multiobjective Evolution Efficiency

Hypervolume [80] represents the volume of the hypercube enclosed by the nondominated solution set and the reference point in the target space [81], which can be used to evaluate the efficiency of multiobjective evolution. A higher value indicates that the evolutionary process introduces more optimal solutions to the population, signifying the attainment of a superior Pareto front. We construct two baseline model for comparing evolution efficiency: random model and multi-objective evolutionary algorithm based on decomposition (MOEA/D). In random model, the O to be input into Set_1 is randomly sampled in the entire search space, rather than the nondominated solutions generated by NSGA-II. In MOEA/D, NSGA-II is replaced by MOEA/D algorithm [82], while genetic operators remain unchanged. Penalty-based boundary intersection (PBI) is used to decompose the multiobjective optimization problem. The changing trend of hypervolume along with the evolution process on CIFAR10-DVS and DVS128-Gesture is drawn in Fig. 4(d) and 4(e). The blue, red, and green lines depict the performance of EB-NAS, MOEA/D, and random models, respectively. It can be observed that the hypervolume of all models continues to expand as the evolution proceeds, while EB-NAS achieves larger hypervolumes faster than other models on both datasets.

In SNNs, information is conveyed through spikes, with fewer spikes indicating more energy-efficient models. By varying the initial channel count of the model, we achieve diverse classification accuracy results evolved on CIFAR10 and then compare them with those of other SNN models, as detailed in Table V. Compared with fixed-structure models, our model has obvious advantages in energy consumption. For example, we use no more than one-third of the number of spikes to achieve about 1% higher performance than [65]. EB-NAS requires only about half the number of spikes (178 000) compared with the same NAS-based method [12] (310 000) to achieve a 2.39% improvement in performance. Furthermore, with similar energy consumption levels (178 000 and 176 000, respectively), EB-NAS exhibits a performance enhancement of 4.22%.

TABLE V
COMPARISON RESULT OF SPIKE NUMBERS OF EB-NAS WITH DIFFERENT ACCURACY ON CIFAR10

SNN Models	Accuracy (%)	Spikes(K)
CIFARNet-Wu [58]	84.36 87.80	361 1298
CIFARNet-Fang [65]	80.82 93.15	104 507
ResNet11-Lee [83]	84.43 90.24	140 1530
ResNet19-Zheng [64]	83.95 93.07	341 1246
AutoSNN [12]	88.67 93.15	108 310
EB-NAS	**89.80** **94.11** **95.54**	**86** **129** **178**

Note: The bold part is to highlight the superiority of our method.

TABLE VI
ROBUSTNESS OF EB-NAS ON CIFAR10 AND CIFAR100

Dataset	Method	Clean Acc (%)	Gaussian Noise (%)
CIFAR10	TIC	92.04	69.01(−23.03)
	EB-NAS	**95.66**	**87.5**(**–8.16**)
CIFAR100	TIC	68.17	37.60(−30.57)
	TIC	68.47	36.98(−31.49)
	EB-NAS	**79.21**	**58.71**(**–20.5**)

Note: The top-1 accuracy of the model without noise is denoted as Clean Acc, while the results with noise are denoted as Gaussian Noise. The bold part is to highlight the superiority of our method.

By introducing Gaussian noise into the model, we verify the robustness of EB-NAS compared with TIC [84]. Specifically, independent Gaussian noise is imposed with a norm equal to 50% of the input norm for each image. As shown in Table VI, On CIFAR10, noise reduces the classification accuracy of EB-NAS by 8.16%, which is far better than TIC's 23.03%. On CIFAR100, the classification accuracy of EB-NAS dropped by 20.5%, which is also better than TIC (more than 30%).

V. CONCLUSION AND FUTURE WORK

Current research on SNNs was predominantly confined to fixed structures inspired by DNNs, and the performance of the few existing NAS-based SNN models does not surpass these, indicating significant potential for enhancement. To address this challenge, we automatically evolve the architecture of SNNs from a brain-inspired perspective based on the combination of common neural motifs and free long-term connections between them. A time-saving online performance predictor was

employed to find high-performance and low-energy SNN architectures and accelerate the multiobjective evolution process. Extensive ablation and comparison experiments show that the proposed biologically interpretable brain-inspired search space and online performance predictor could effectively improve the classification accuracy of candidate SNN architectures through neuroevolution and achieve superior performance with significantly lower energy consumption compared to both fixed-architecture SNNs and NAS-based SNNs. This not only showcases the powerful performance of EB-NAS but also highlights the potential of auto-evolving SNN architectures to outperform human expertise, underscoring the efficacy of natural evolutionary mechanisms in the practical design of SNN architectures from a computational modeling perspective.

However, verification in this article was only limited to classification tasks. As we continue to unravel the mysteries of the human brain, we expect to identify other brain-inspired topological properties that could enhance the cognitive functions of SNNs. These insights will offer fresh perspectives for tackling complex tasks such as transfer learning, continuous learning, and lifelong learning, paving the way for the development of general artificial intelligence characterized by adaptive evolution across multiple brain regions.

REFERENCES

[1] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Netw.*, vol. 10, no. 9, pp. 1659–1671, 1997.

[2] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 4780–4789.

[3] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8697–8710.

[4] Y. Liu, Y. Sun, B. Xue, M. Zhang, G. G. Yen, and K. C. Tan, "A survey on evolutionary neural architecture search," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 2, pp. 550–570, Feb. 2023.

[5] G. Bender, P.-J. Kindermans, B. Zoph, V. Vasudevan, and Q. Le, "Understanding and simplifying one-shot architecture search," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2018, pp. 550–559.

[6] Y. Sun, H. Wang, B. Xue, Y. Jin, G. G. Yen, and M. Zhang, "Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 350–364, Apr. 2020.

[7] W. Wen, H. Liu, Y. Chen, H. Li, G. Bender, and P.-J. Kindermans, "Neural predictor for neural architecture search," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, U.K. (Part XXIX). New York, NY, USA: Springer-Verlag, 2020, pp. 660–676.

[8] J. Mellor, J. Turner, A. Storkey, and E. J. Crowley, "Neural architecture search without training," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 7588–7598.

[9] X. Xie, X. Song, Z. Lv, G. G. Yen, W. Ding, and Y. Sun, "Efficient evaluation methods for neural architecture search: A survey," 2023, *arXiv:2301.05919*.

[10] B. Deng, J. Yan, and D. Lin, "Peephole: Predicting network performance before training," 2017, *arXiv:1712.03351*.

[11] Z. Lu, K. Deb, E. Goodman, W. Banzhaf, and V. N. Boddeti, "NSGANetV2: Evolutionary multi-objective surrogate-assisted neural architecture search," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, U.K. (Part I 16). New York, NY, USA: Springer-Verlag, 2020, pp. 35–51.

[12] 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.*, PMLR, 2022, pp. 16253–16269.

[13] Y. Kim, Y. Li, H. Park, Y. Venkatesha, and P. Panda, "Neural architecture search for spiking neural networks," in *Proc. 17th Eur. Conf. Comput. Vis. (ECCV)*, Tel Aviv, Israel (Part XXIV), New York, NY, USA: Springer-Verlag, 2022, pp. 36–56.

[14] L. Luo, "Architectures of neuronal circuits," *Science*, vol. 373, no. 6559, 2021, Art. no. eabg7285.

[15] M. A. Meredith, "On the neuronal basis for multisensory convergence: A brief overview," *Cogn. Brain Res.*, vol. 14, no. 1, pp. 31–40, 2002.

[16] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 1997–2017, 2019.

[17] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7036–7045.

[18] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," 2016, *arXiv:1611.01578*.

[19] P. Ren et al., "A comprehensive survey of neural architecture search: Challenges and solutions," *ACM Comput. Surv. (CSUR)*, vol. 54, no. 4, pp. 1–34, 2021.

[20] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.

[21] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN'95)*, vol. 4, Piscataway, NJ, USA: IEEE Press, 1995, pp. 1942–1948.

[22] T. Bäck, D. B. Fogel, and Z. Michalewicz, "Handbook of Evolutionary Computation," *Release*, vol. 97, no. 1, 1997, Art. no. B1.

[23] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. New York, NY, USA: Springer-Verlag, 2006.

[24] Y. Sun, G. G. Yen, and Z. Yi, "IGD indicator-based evolutionary algorithm for many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 173–187, Apr. 2019.

[25] Z. Lu et al., "NSGA-Net: Neural architecture search using multi-objective genetic algorithm," in *Proc. Genetic Evol. Comput. Conf.*, 2019, pp. 419–427.

[26] Z. Lu et al., "Multiobjective evolutionary design of deep convolutional neural networks for image classification," *IEEE Trans. Evol. Comput.*, vol. 25, no. 2, pp. 277–291, 2020.

[27] T. Elsken, J. H. Metzen, and F. Hutter, "Efficient multi-objective neural architecture search via Lamarckian evolution," 2018, *arXiv:1804.09081*.

[28] "CARS: Continuous evolution for efficient neural architecture search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1829–1838.

[29] H. Cai, L. Zhu, and S. Han, "ProxylessNAS: Direct neural architecture search on target task and hardware," 2018, *arXiv:1812.00332*.

[30] A. Howard et al., "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1314–1324.

[31] M. Tan et al., "MNASNet: Platform-aware neural architecture search for mobile," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2820–2828.

[32] B. Wu et al., "FBNet: Hardware-aware efficient convnet design via differentiable neural architecture search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10734–10742.

[33] X. Chu, B. Zhang, and R. Xu, "FairNAS: Rethinking evaluation fairness of weight sharing neural architecture search," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 12239–12248.

[34] Z. Lu, K. Deb, and V. N. Boddeti, "MUXConv: Information multiplexing in convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12044–12053.

[35] E. Real et al., "Large-scale evolution of image classifiers," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2017, pp. 2902–2911.

[36] Y. Benyahia et al., "Overcoming multi-model forgetting," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 594–603.

[37] J. Xu, L. Zhao, J. Lin, R. Gao, X. Sun, and H. Yang, "KNAS: Green neural architecture search," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 11613–11625.

[38] M. Lin et al., "Zen-NAS: A zero-shot NAS for high-performance image recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 347–356.

[39] C. Liu et al., "Progressive neural architecture search," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 19–34.

[40] 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.

[41] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[42] T. Elsken, J.-H. Metzen, and F. Hutter, "Simple and efficient architecture search for convolutional neural networks," 2017, *arXiv:1711.04528*.

[43] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
[44] C. Ying, A. Klein, E. Christiansen, E. Real, K. Murphy, and F. Hutter, "NAS-Bench-101: Towards reproducible neural architecture search," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 7105–7114.
[45] X. Dong and Y. Yang, "NAS-Bench-201: Extending the scope of reproducible neural architecture search," 2020, *arXiv:2001.00326*.
[46] B. Nicolas, C. Mark, and W. van Rossum, "Quantitative investigations of electrical nerve excitation treated as polarization," *Biological Cybernetics*, vol. 97, no. 5–6, p. 341, 2007.
[47] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *Frontiers Neurosci.*, vol. 12, 2018, Art. no. 331.
[48] 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, Jul. 2023, Art. no. 100789, doi: 10.1016/j.patter.2023.100789.
[49] J. S. Isaacson and M. Scanziani, "How inhibition shapes cortical activity," *Neuron*, vol. 72, no. 2, pp. 231–243, 2011.
[50] G. Laurent, M. Stopfer, R. W. Friedrich, M. I. Rabinovich, A. Volkovskii, and H. D. Abarbanel, "Odor encoding as an active, dynamical process: experiments, computation, and theory," *Annu. Rev. Neurosci.*, vol. 24, no. 1, pp. 263–297, 2001.
[51] S. Grillner, "Biological pattern generation: The cellular and computational logic of networks in motion," *Neuron*, vol. 52, no. 5, pp. 751–766, 2006.
[52] C. B. Saper, P. M. Fuller, N. P. Pedersen, J. Lu, and T. E. Scammell, "Sleep state switching," *Neuron*, vol. 68, no. 6, pp. 1023–1042, 2010.
[53] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Proc. Parallel Problem Solving from Nature PPSN VI: 6th Int. Conf.,* Paris, France. New York, NY, USA: Springer-Verlag, 2000, pp. 849–858.
[54] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," May 2012.
[55] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*.
[56] A. Amir et al., "A low power, fully event-based gesture recognition system," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7243–7252.
[57] H. Li, H. Liu, X. Ji, G. Li, and L. Shi, "CIFAR10-DVS: An event-stream dataset for object classification," *Frontiers Neurosci.*, vol. 11, 2017, Art. no. 309.
[58] 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. 1, pp. 1311–1318.
[59] W. Zhang and P. Li, "Temporal spike sequence learning via backpropagation for deep spiking neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 12022–12033.
[60] N. Rathi, G. Srinivasan, P. Panda, and K. Roy, "Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation," 2020, *arXiv:2005.01807*.
[61] Y. Zhu, Z. Yu, W. Fang, X. Xie, T. Huang, and T. Masquelier, "Training spiking neural networks with event-driven backpropagation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 30528–30541.
[62] N. Rathi and K. Roy, "Diet-SNN: Direct input encoding with leakage and threshold optimization in deep spiking neural networks," 2020, *arXiv:2008.03658*.
[63] Y. Li, S. Deng, X. Dong, R. Gong, and S. Gu, "A free lunch from ANN: Towards efficient, accurate spiking neural networks calibration," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 6316–6325.
[64] 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.
[65] 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.
[66] B. Han, G. Srinivasan, and K. Roy, "RMP-SNN: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13558–13567.
[67] M. Xiao, Q. Meng, Z. Zhang, D. He, and Z. Lin, "Online training through time for spiking neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 20717–20730.
[68] S. Deng, Y. Li, S. Zhang, and S. Gu, "Temporal efficient training of spiking neural network via gradient re-weighting," 2022, *arXiv:2202.11946*.
[69] S. Yang and B. Chen, "Effective surrogate gradient learning with high-order information bottleneck for spike-based machine intelligence," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 22, 2023.
[70] S. Yang, H. Wang, and B. Chen, "SIBoLS: robust and energy-efficient learning for spike-based machine intelligence in information bottleneck framework," *IEEE Trans. Cogn. Devel. Syst.*, early access, Nov. 6, 2023.
[71] I. Garg, S. S. Chowdhury, and K. Roy, "DCT-SNN: Using DCT to distribute spatial information over time for low-latency spiking neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 4671–4680.
[72] S. Park, S. Kim, B. Na, and S. Yoon, "T2FSNN: Deep spiking neural networks with time-to-first-spike coding," in *Proc. 57th ACM/IEEE Des. Automat. Conf. (DAC)*, Piscataway, NJ, USA: IEEE Press, 2020, pp. 1–6.
[73] Y. Guo et al., "Real spike: Learning real-valued spikes for spiking neural networks," in *Proc. Eur. Conf. Comput. Vis.*, New York, NY, USA: Springer-Verlag, 2022, pp. 52–68.
[74] Y. Li, Y. Guo, S. Zhang, S. Deng, Y. Hai, and S. Gu, "Differentiable spike: Rethinking gradient-descent for training spiking neural networks," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 23426–23439.
[75] A. Kugele, T. Pfeil, M. Pfeiffer, and E. Chicca, "Efficient processing of spatio-temporal data streams with spiking neural networks," *Frontiers Neurosci.*, vol. 14, 2020, Art. no. 439.
[76] 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.
[77] Y. Xing, G. Di Caterina, and J. Soraghan, "A new spiking convolutional recurrent neural network (SCRNN) with applications to event-based hand gesture recognition," *Frontiers Neurosci.*, vol. 14, 2020, Art. no. 590164.
[78] S. B. Shrestha and G. Orchard, "Slayer: Spike layer error reassignment in time," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1419–1428.
[79] C. Spearman, "The proof and measurement of association between two things." *The American Journal of Psychology*, vol. 100, no. 3–4, pp. 441–471, 1987.
[80] E. Zitzler, D. Brockhoff, and L. Thiele, "The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration," in *Proc. 4th Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, Matsushima, Japan. New York, NY, USA: Springer-Verlag, 2007, pp. 862–876.
[81] X. Lin, Z. Lin, and S. Wei, "Multi-objective optimized driving strategy of dual-motor EVs using NSGA-II as a case study and comparison of various intelligent algorithms," *Appl. Soft Comput.*, vol. 111, 2021, Art. no. 107684.
[82] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
[83] C. Lee, S. S. Sarwar, P. Panda, G. Srinivasan, and K. Roy, "Enabling spike-based backpropagation for training deep neural network architectures," *Frontiers Neurosci.*, vol. 14, 2020, Art. no. 119.
[84] Y. Kim, Y. Li, H. Park, Y. Venkatesha, A. Hambitzer, and P. Panda, "Exploring temporal information dynamics in spiking neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 7, pp. 8308–8316.

Wenxuan Pan received the B.Eng. degree in software engineering from the University of Electronic Science and Technology of China, Chengdu, Sichuan, China. She is currently working toward the Ph.D. degree with the Brain-inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences, Beijing, China, supervised by Prof. Yi Zeng.

Her research interests focus on brain-inspired evolutionary spiking neural networks.

Feifei Zhao received the B.S. degree in digital media technology from the Northeastern University, Shenyang, Liaoning, China, in 2014, and the Ph.D. degree in pattern recognition and intelligent systems from the University of Chinese Academy of Sciences, Beijing, China, in 2019.

She is currently an Associate Professor with the Brain-inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences, Beijing, China. Her research interests include multibrain areas coordinated learning and decision-making spiking neural network, brain-inspired developmental, and evolutionary architecture search for SNNs.

Zhuoya Zhao received the B.Eng. degree in measurement and control technology and instruments from Harbin Engineering University, Harbin, Heilongjiang, China, in 2019. She is currently working toward the Ph.D. degree with the Brain-inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences, Beijing, China, supervised by Prof. Yi Zeng.

Her research interests include brain-inspired theory of mind spiking neural networks.

Yi Zeng received his Ph.D degree in 2010 from Beijing University of Technology, Beijing, China. He is currently a Professor and the Director with the Brain-inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China. He is a Principal Investigator in the Key Laboratory of Brain Cognition and Brain-inspired Intelligence Technology, Chinese Academy of Sciences, and a Professor with the School of Future Technology, and the School of Humanities, University of Chinese Academy of Sciences, and a Founding Director of Center for Long-term Artificial Intelligence, Beijing, China. His research interests include brain-inspired Artificial Intelligence, brain-inspired cognitive robotics, and ethics and governance of artificial intelligence.