

Developmental Plasticity-inspired Adaptive Pruning for Deep Spiking and Artificial Neural Networks

Bing Han, Feifei Zhao, Yi Zeng, Guobin Shen

arXiv:2211.12714v3 [cs.NE] 28 Oct 2024

Abstract—Developmental plasticity plays a prominent role in shaping the brain’s structure during ongoing learning in response to dynamically changing environments. However, the existing network compression methods for deep artificial neural networks (ANNs) and spiking neural networks (SNNs) draw little inspiration from brain’s developmental plasticity mechanisms, thus limiting their ability to learn efficiently, rapidly, and accurately. This paper proposed a developmental plasticity-inspired adaptive pruning (DPAP) method, with inspiration from the adaptive developmental pruning of dendritic spines, synapses, and neurons according to the “use it or lose it, gradually decay” principle. The proposed DPAP model considers multiple biologically realistic mechanisms (such as dendritic spine dynamic plasticity, activity-dependent neural spiking trace, and local synaptic plasticity), with additional adaptive pruning strategy, so that the network structure can be dynamically optimized during learning without any pre-training and retraining. Extensive comparative experiments show consistent and remarkable performance and speed boost with the extremely compressed networks on a diverse set of benchmark tasks for deep ANNs and SNNs, especially the spatio-temporal joint pruning of SNNs in neuromorphic datasets. This work explores how developmental plasticity enables complex deep networks to gradually evolve into brain-like efficient and compact structures, eventually achieving state-of-the-art (SOTA) performance for biologically realistic SNNs.

Index Terms—Integrated Development Mechanisms, Dendritic Spine Dynamic Plasticity, Activity-dependent Synaptic plasticity, Brain-inspired Deep ANNs and SNNs Compression

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

Yi Zeng is with the Brain-inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and Center for Long-term Artificial Intelligence, Beijing 100190, China, and University of Chinese Academy of Sciences, Beijing 100049, China, and Key Laboratory of Brain Cognition and Brain-inspired Intelligence Technology, Chinese Academy of Sciences, Shanghai, 200031, China.

Guobin Shen is with the Brain-inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and School of Future Technology, University of Chinese Academy of Sciences, Beijing 100049, China.

The first and the second authors contributed equally to this work, and serve as co-first authors.

The corresponding author is Yi Zeng (e-mail: yi.zeng@ia.ac.cn).

This paper was produced by the IEEE Publication Technology Group. They are in Piscataway, NJ.

Manuscript received April 19, 2021; revised August 16, 2021.

I. INTRODUCTION

THE human brain with highly plastic is the product of hundreds of millions of years of evolution, thereby allowing infants to emerge with high-level intelligence as they grow and develop. Neural circuits and network topology of the brain are also the products of development over an individual’s life span. Since the birth of the baby, synapses first undergo explosive growth, peaking by age two or three. Then those surplus synapses are gradually eliminated throughout childhood and adolescence according to adaptive pruning mechanisms [1], [2]. This developmental process dynamically shapes the network structure as a result of continuous interaction with the environment and neural changes induced by learning [3], [4]. The developmental plasticity of the brain enables it to show remarkable plasticity in response to changing environments and to perform multiple complex cognitive functions with extremely low energy consumption. However, current deep neural networks (DNNs) and deep spiking neural networks (DSNNs) employ complex networks with a large number of parameters to solve a single task, which leads to prohibitively expensive computational costs and storage overhead. Besides, DNNs and DSNNs without developmental structural plasticity lack sufficient adaptability and flexibility in learning different tasks. This is a significant gap between baby-like highly-efficient learning and adaptive development.

Taking inspiration from the multi-scale developmental plasticity in the brain, whereby dendritic spines, synapses, and neurons adaptive formation and elimination according to the “use it or lose it” principle, we proposed a generalized developmental plasticity-inspired adaptive pruning (DPAP) method for DNNs and SNNs. Incorporating DPAP into the ongoing learning and optimization of neural networks enables dynamically pruning redundant synapses and neurons according to their activity levels. Especially for event-driven SNNs, we use neuronal temporal spiking sequences to represent activity levels and incorporate temporal dimension pruning. Different from the existing network compression models, DPAP is remarkable at multiple levels, more biologically plausible with ongoing developmental plasticity, more adaptive pruning for efficient structure shaping, and naturally brings superior performance and learning speed.

The existing model compression methods are intended to reduce memory and operations consumption while minimizing accuracy drop. The DNNs compression methods include pruning [5], quantization [6], [7]and knowledge distillation [8]. Here, we focus on the pruning methods. DNNs pruning methods considered weight magnitude [9], weight gradient

[10], [11], weight similarity [12], Batch Normalization (BN) factor [13] as evaluation criteria, pruning fine-grained individual parameters [14] or coarse-grained overall structures [15], [16]. Deep networks are very sensitive to such pruning strategies, thus pre-training and retraining are required to guarantee performance, which is not biologically plausible. Some developmental plasticity-inspired pruning methods prune neurons or synapses adaptively through a biologically reasonable dynamic strategy, helping to effectively prevent overfitting and underfitting [17]–[19]. Such methods are only suitable for shallow artificial neural networks (ANNs), and the pure biological brain development mechanism has not been well understood and referenced.

Spiking Neural Networks (SNNs) are considered to be the third generation neural networks [20], with spike event-driven computation, spatio-temporal joint information processing and high biological plausibility [21], which is more in line with the processing mechanism of the brain nervous network. Therefore, learning from the adaptive pruning mechanism of brain development is an effective way to prune SNNs, which is also lacking in current studies. Many existing methods simply apply pruning methods in ANNs to SNNs [22]–[26], which ignore the unique information processing with binary spikes of SNNs, thus limiting the performance of pruned SNNs. Some more biological SNNs pruning methods use spike-timing dependent plasticity (STDP) as an evaluation criterion, dynamically prune synapses with smaller weights or decayed weights in shallow SNNs [27]–[30]. Essentially, STDP as a local unsupervised plasticity mechanism is hard to be applied to deep SNN learning and far from the brain’s pruning mechanism. Besides, these methods are limited to spatial pruning, ignoring the SNN’s unique temporal dimension.

Although these attempts have become a feasible way of compressing deep networks, they draw little inspiration from the brain’s development. Substantial efforts are still needed toward studying developmental plasticity-inspired deep networks, as only such biologically interpretable and plausible methods have the potential to approach the highly efficient brain nervous system. This paper aims to incorporate multi-scale spatio-temporal developmental plasticity mechanisms into both DNNs and DSNNs and answer how much the brain’s adaptive pruning mechanism helps to better shape the network’s structure.

As the brain ongoing learns, neurons dynamically stretch out multiple dendrites for receiving information, and some dendritic spines form synapses through specific connections [31]. Synaptic plasticity between pre-synaptic and post-synaptic neurons contributes to the connectivity and efficiency of neural circuits that support learning, memory, and other cognitive abilities [32], [33]. During developmental pruning of the brain, dendritic spines, synapses, and neurons are continuously strengthened or decayed or even death according to the “use it or lose it, gradually decay” principle [34]. Dendritic spines formation (or enlargement) and elimination (or shrinkage) depend on the temporal spiking activity of the post-synaptic neuron: repeated inductions of long-term potentiation (LTP) lead to dendritic spines formation (or enlargement), whereas long-term depression (LTD) coupled with the elimination

(or shrinkage) of spines [35], [36]. Repeated activation and frequent use after LTP also strengthen neuronal activity levels and synaptic efficacy. All these temporal event-driven developmental plasticity mechanisms induce adaptive pruning according to synaptic and neuronal efficacy. Specifically, brain pruning principles include: 1) Synapses and neurons that are rarely used are more likely to be eliminated during the pruning process [37]. 2) Unimportant and redundant synapses and neurons are first gradually decayed and eventually pruned away [38]. 3) Dendritic spine elimination precedes synaptic pruning, and synaptic pruning precedes neural death [39], [40].

In this paper, we propose a generalized adaptive pruning algorithm for SNNs and DNNs inspired by the developmental plasticity of the brain. The proposed algorithm integrates multi-scale spatio-temporal developmental plasticity as the importance measure, and combines the pruning strategy of “use it or lose it, gradually decay” to dynamically eliminate redundancy progressively evolving brain-inspired compact neural circuits and network architectures. The main contributions of this paper are summarized as follows:

- Synthesizing dendritic spine dynamic plasticity, local Bienenstock-Cooper-Munros (BCM) [41] synaptic plasticity, and activity-dependent neural spiking trace, we propose a generalized adaptive pruning algorithm to dynamically prune inactive synapses and neurons in both SNNs and DNNs.
- For event-driven SNNs, we spatially prune neurons and synapses based on temporal spiking traces. Inspired by the temporal attention mechanism of the brain, we further implement temporal dimension pruning for SNNs to reduce energy consumption even more.
- We introduce the proposed algorithm into both DNNs and SNNs improving convergence speed and accuracy while reducing energy consumption for various temporal and spatial datasets. Specifically, the proposed spatio-temporal pruning method brings state-of-the-art (SOTA) performance with only 6.00% energy consumption for SNNs on temporal datasets.

The remainder of this paper is organized as follows. In Section II, we present the proposed DPAP framework in detail. In Sections III, we evaluate the performance of DPAP on SNNs and DNNs. We conduct a series of discussions and analyses in Section IV. Finally, we conclude our findings in Section V.

II. DEVELOPMENTAL PLASTICITY-INSPIRED ADAPTIVE PRUNING ALGORITHM

In this section, we present the proposed developmental plasticity-inspired adaptive pruning algorithm generalized for SNNs and DNNs, as shown in Fig. 1. We describe the overall framework of the proposed learning-while-pruning algorithm. Then, we provide computational details of the bioplasticity-based importance. Finally, we introduce the proposed adaptive pruning strategy in SNNs and DNNs respectively.

A. The overall learning-while-pruning framework

DPAP adaptively prunes irrelevant synapses and neurons during the ongoing learning process without pre-training and

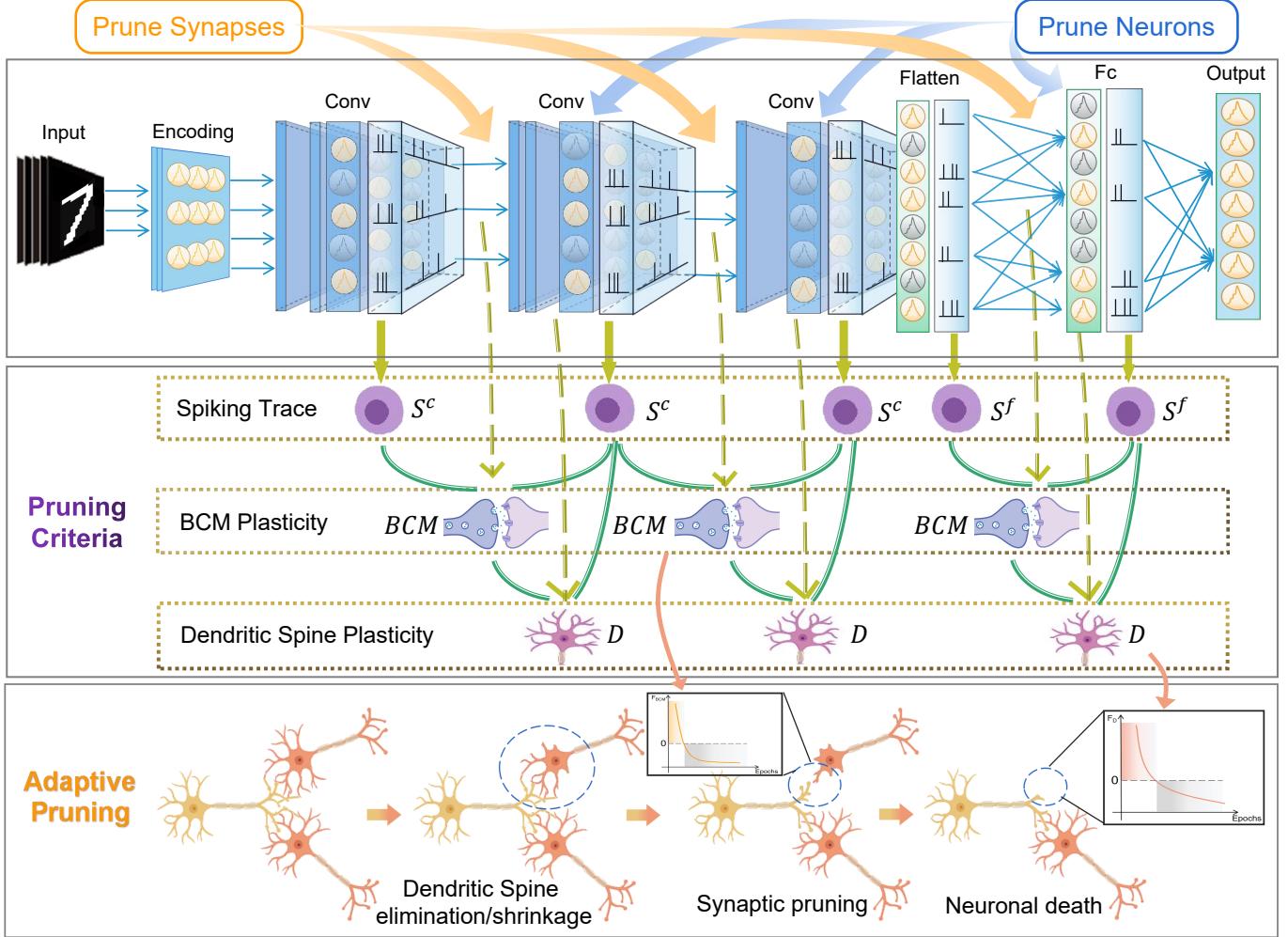


Fig. 1. **The procedure of DPAP method.** The SNN structure (**top block**) consists of convolutional layers and fully connected layers. Pruning criteria (**middle block**) contains trace-based BCM plasticity for synapses and dendritic spine plasticity for neurons. Adaptive pruning (**bottom block**) gradually prunes decayed synapses and neurons according to survival function. The orange graph represents the survival function of the pruned synapse, and the red graph represents the survival function of pruned neurons.

re-training. According to the “use it or lose it” principle, the DPAP takes inspiration from multiscale brain pruning mechanisms, including local synaptic plasticity, activity-dependent neural spiking trace, and dendritic spine dynamic plasticity:

1) *Trace-based BCM synaptic plasticity:* We employed trace-based BCM synaptic plasticity [41] to measure the importance (or efficacy) of synapses, as BCM can induce LTP or LTD based on the activity of pre-synaptic and post-synaptic neurons, which is consistent with the criterion of brain synaptic pruning. Furthermore, the introduction of spiking trace makes the later timestep the neural firing, the stronger the correlation, which temporal information distinguishes the magnitude of synaptic or neuronal activity. Trace-based BCM is not only determined by the current spiking traces of pre-synaptic and post-synaptic neurons, but also takes into account the average of the historical activity of the post-synaptic neuron. Thus, the synaptic importance increases when the trace of the pre-synaptic neuron is sufficient to activate the post-synaptic neurons’ trace above the historical threshold. Otherwise, the synaptic importance declines.

2) *Activity-dependent Neural Spiking Trace:* For the unique spatio-temporal information fusion of SNNs, the activity level of the neuron is measured by the spiking trace, which takes into account the spiking sequence in the previous period and the firing state at the current moment. Pre- and post-synaptic spiking traces model the effects of N-methyl-D-aspartate (NMDA) and Ca^{2+} on biological neuronal activity, respectively [42], [43]. At any time, the neural spiking trace S is accumulated by 1 when the neuron emits a spike, otherwise, the trace gradually decays with a time constant τ . In DNNs, spiking traces are represented by neuron activation outputs.

3) *Dendritic Spine Dynamic Plasticity:* Neurons extend a large number of dendritic spines to receive pre-synaptic information, and the density and volume of dendritic spines could reflect the activity level of neurons. For dendritic spines, some spines form synapses with pre-synaptic axon terminals to receive pre-synaptic spike signal transmission, and other spines are also expanding and shrinking in preparation for the formation of synaptic connections. Thus, the dynamic plasticity of dendritic spines incorporates the BCM synaptic plasticity

with the neural spiking traces. Dendritic spines provide a comprehensive measure of the efficacy and importance of the neurons.

Next, the proposed adaptive pruning strategy is fed with synaptic importance measured by BCM synaptic plasticity and neuronal importance measured by dendritic spine plasticity to dynamically prune neurons and synapses. Drawing on the “gradually decay or even die” mechanism in brain developmental pruning, we define a survival function to decide whether synapses or neurons are removed. The survival function considers continuous changes in the importance of neurons and synapses, and only neurons and synapses with negative values of the survival function are permanently deleted. Thus, DPAP could ensure that the pruned synapses and neurons are redundant and unimportant, and naturally more biologically plausible.

For an epoch training of SNN and DNN, we follow the common direct training algorithm for SNNs and the back-propagation algorithm for DNNs to update weight gradient in the learning process of each batch. Only after all batch is completed, we calculate the importance of neurons and synapses and prune the network redundancy according to the proposed adaptive pruning strategy. For convolutional layers, we adopt structured pruning that treats each channel as an overall neuron population to prune. The DPAP method computational details are in the following sections.

B. Biologically Plausible Pruning Criteria for SNNs

In SNN, we used the common leaky integrate-and-fire (LIF) spiking neurons [44] as the basic unit and trained the network directly with the surrogate gradient algorithm [45], [46]. Event-driven temporal characterization is the unique advantage making SNN bio-interpretable and energy-efficient. Hence, both our synaptic and neuronal pruning criteria in SNN are calculated based on the temporal spiking sequences.

1) Activity-dependent neural spiking trace: To synthesize the spike firing of the neuron at all timesteps, we introduced the spiking trace to represent the neuronal activity level. At the t -th timestep, if the neuron fires, its spiking trace S will be added by 1. Otherwise, its spiking trace decayed with time constant $\tau = 0.5$. For the fully connected layer, the spiking trace of the neuron i in layer f at timestep $t+1$ is calculated:

$$S_i^{t+1,f} = \tau S_i^{t,f} + o_i^{t+1,f} \quad (1)$$

Considering the structural characteristics of the convolutional layer, we regard each channel as an entire population of neurons. The neural spiking trace for the convolution layer c which has $C \times N \times N$ neurons is calculated as follow:

$$S_i^{t+1,c} = \tau S_i^{t,c} + \sum_{k=1}^{N,N} o_i^{t+1,c} \quad (2)$$

Therefore, the more spikes a neuron fires in its given timesteps, the larger the spiking trace.

2) Trace-based BCM synaptic plasticity: The trace-based BCM not only calculates the spiking traces in the current epoch, but also considers the historical neuronal traces. The synaptic importance is jointly measured by the pre-synaptic and post-synaptic spiking traces. In particular, post-synaptic neurons are considered more important, the difference between their spiking trace and the sliding threshold θ determines the direction of the importance update. The trace-based BCM of batch b is calculated by:

$$BCM_{pre-post}^b = S_{pre}^T \cdot S_{post}^T \cdot (S_{post}^T - \theta) \quad (3)$$

where S_{pre}^T and S_{post}^T are the pre- and post-synaptic spiking trace over T timesteps, respectively. The sliding threshold θ is the historical average of post-synaptic neuronal activity, as shown in Eq 4. It determines the direction of LTP and LTD synaptic plasticity.

$$\theta = \frac{\theta * (Num - 1) + S_{post}}{Num} \quad (4)$$

where Num is the number of all the batches experienced from the beginning of learning. At the batch b in the epoch e , Num is the following value:

$$Num = e * N_{batch} + b \quad (5)$$

where N_{batch} is the number of the batch in an epoch. For each epoch, the synaptic importance pruning criteria BCM^e is calculated as the sum of all batches:

$$BCM^e = \sum_{b=1}^{N_{batch}} BCM_{pre-post}^b \quad (6)$$

3) Dendritic spine dynamic plasticity: Dendritic spine plasticity is jointly determined by the neural spiking traces and the trace-based BCM synaptic plasticity. Therefore, for each epoch, dendritic spine dynamic plasticity D^e as the neuronal importance pruning criteria is calculated as follow:

$$D^e = \sum_{b=1}^{N_{batch}} S_{post}^T * \sum_{j=1}^{N_{pre}} BCM^e \\ = \sum_{b=1}^{N_{batch}} (S_{post}^T \sum_{j=1}^{N_{pre}} S_{pre}^T \cdot S_{post}^T \cdot (S_{post}^T - \theta)) \quad (7)$$

where N_{pre} is the number of pre-synaptic channels or neurons. In summary, both synaptic and neuronal importance are calculated relying on the neuronal spiking traces, that is, the spiking information in the temporal dimension unique to the SNN.

4) Timestep importance criteria: When processing temporal sequential data, the brain adapts to accurately capture valid events in key times [47], [48]. Spatio-temporal integrated SNN excels at processing event-driven neuromorphic temporal datasets, but the cumulative computation of multiple timesteps greatly increases the energy consumption of SNN. Therefore, inspired by the temporal attention mechanism of the brain, we design SNN-unique temporal dimension pruning to eliminate redundant timesteps.

Specifically, the importance of the current timestep depends on the neuronal activity at that timestep. It is determined by the acceptable spikes already fired by the pre-synaptic neurons before timestep t and spike from this neuron at the current timestep. We follow the proposed spike-based importance, and the timestep t importance of neuron i is calculated by the spiking trace of the pre-synaptic neuron during the $0 \sim t$ timesteps and the spike of the post-synaptic neuron at the timestep t , as follow:

$$Time^t = \sum_{j=1}^{N_{pre}} S_{pre}^{0 \sim t} \cdot O_{post}^t \quad (8)$$

C. Developmental Plasticity-inspired Adaptive Pruning

1) DPAP in SNNs: The DPAP method prunes unimportant synapses, neurons and timesteps according to their importance BCM^e , D^e and $Time^t$. Inspired by the pruning mechanism of brain development, we designed the survival function for neuron F_D , synapse F_{BCM} and timestep F_{Time} . At the beginning of learning, we initialized the survival functions as the constant β . Take synapses for example, the trace-based BCM synaptic plasticity linearly normalized in each epoch as follow:

$$\delta_{BCM}^e = 2 * Normalized(BCM^e) - \epsilon \quad (9)$$

where ϵ is the decay value. Then, we calculated the update value of synapse survival function ΔF_{BCM}^e :

$$\Delta F_{BCM}^e = \begin{cases} \delta_{BCM}^e + C, & \delta_{BCM}^e \geq 0 \\ \delta_{BCM}^e, & \text{otherwise} \end{cases} \quad (10)$$

For the reliability of pruning, we protected the synapses with positive δ_{BCM}^e by extra increasing by constant C ($C = 5$ in convolutional layers, $C = 2$ in fully connected layers). During the whole learning process, the survival function F_{BCM} are updated as the decay rate η :

$$F_{BCM} = \gamma F_{BCM} + e^{-\frac{\text{epoch}}{\eta}} \Delta F_{BCM}^e \quad (11)$$

where $\gamma = 0.999$ is the decay constant of the survival function.

The neuron survival function F_{BCM} and the timestep survival function F_{Time} are calculated and updated consistent with the above synapse survival function F_{BCM} . Finally, we prune the synapses whose $F_{BCM} < 0$ by setting their weight $w_{ij} = 0$, prune the neurons whose $F_D < 0$ by setting all their pre-synaptic weight $W_i = 0$, and prune the neuronal timestep whose $F_{Time} < 0$ by forcing the neuron to no longer firing spike at this timestep.

2) DPAP in DNNs: For traditional DNNs, we employed the RELU activation function and the cross-entropy loss function. Since DNNs do not have temporal dimension information which is distinct from SNNs, we only prune synapses and neurons in the spatial dimension. Besides, the neural spiking trace of the neuron in DNN is defined as the output of the neuron after activation function. For the fully connected layer:

$$S_i^{DNN,f} = \text{RELU}(W_f x_j^{f-1} + b_f) \quad (12)$$

Algorithm 1: The DPAP Algorithm

```

Input: Base redundant model; Training set  $Y$ .
Output: The pruned model.
Initialize: randomly initialize network weight  $W$ ,
initialize  $F_{BCM} = \beta$ ,  $F_D = \beta$  for each layer.
for  $e = 0$ ;  $e < Epoch$ ;  $e++$  do
    for  $b = 0$ ;  $b < N_{batch}$ ;  $b++$  do
        //Learning network weights
        Forward propagation getting outputs;
        Backward propagation by surrogate gradient
        for SNN or standard BP for DNN;
        //Collecting variables needed for pruning
        Calculate spiking trace as Eq 1,2 or Eq12,13;
        Calculate  $BCM_{pre-post}^b$  as Eq 3,4;
    end
    //Pruning network redundancy
    Calculate  $BCM^e$  and  $D^e$  as Eq 6 and 7;
    Calculate the  $F_{BCM}$  and  $F_D$  as Eq 9 to Eq 11;
    Prune neurons with  $F_D < 0$  and synapses with
     $F_{BCM} < 0$ ;
end

```

For the convolution layer with $C \times N \times N$ neurons:

$$S_i^{DNN,c} = \sum_{k=1}^{N,N} \text{RELU}(W_c \otimes x_j^{c-1} + b_c) \quad (13)$$

where x_j^{f-1} and x_j^{c-1} is the inputs of the layer.

For each batch, the trace-based BCM synaptic plasticity $BCM_{pre-post}^b$ is calculated same as Eq 3 based on pre- and post-synaptic spiking trace as Eq 12,13. The sliding threshold is update as Eq 4. For each epoch, the BCM^e is the sum of $BCM_{pre-post}^b$ and the dendritic spine D^e is calculated same as Eq 7. Besides, the adaptive pruning strategy in DNNs is the same as in SNNs as Eq 9 to 11.

For a feed-forward SNN or DNN, we provide the generic detailed procedure for the DPAP algorithm as Algorithm 1:

TABLE I
INITIAL STRUCTURES OF DIFFERENT TASKS

	Dataset	Initial structure
SNN	N-MNIST	15C3-AvgPool2-40C3-AvgPool2-300FC-10FC
	DVS-Gesture	15C3-AvgPool2-40C3-AvgPool2-300FC-10FC
	MNIST	15C3-AvgPool2-40C3-AvgPool2-300FC-10FC
	CIFAR-10	128C3-BN-128C3-BN-MaxPool2-256C3-BN-256C3 -BN-MaxPool2-512C3-BN-512C3-BN-512FC-10FC
DNN	ImageNet	ResNet-18
	MNIST	20C5-MaxPool2-50C5-MaxPool2-500FC-10FC
	CIFAR-10	VGG16
	ImageNet	ResNet-50

III. EXPERIMENTS

In this section, we performed extensive experiments and comparisons with others validating that the DPAP method

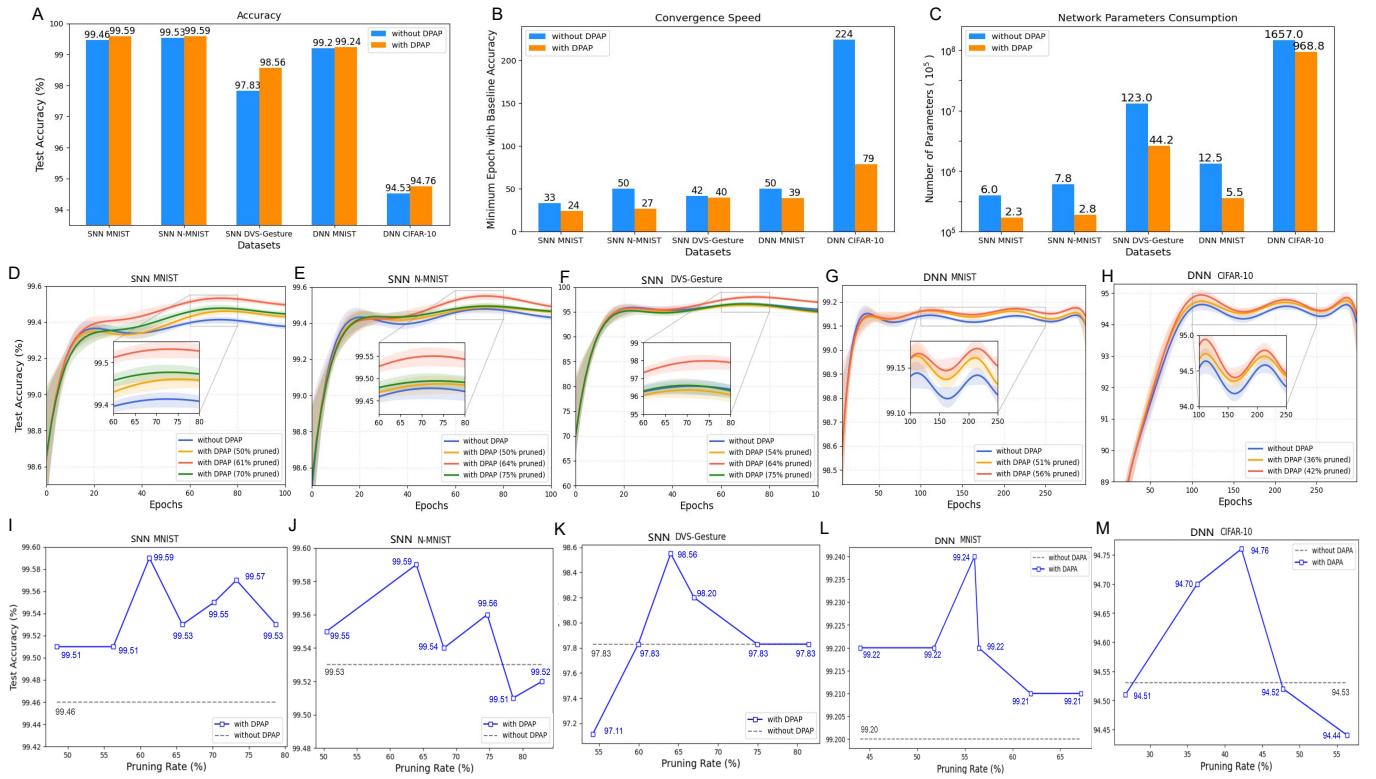


Fig. 2. The effectiveness of introducing DPAP to DSNNs and DNNs. (A) to (C): The test accuracy, convergence speed and energy consumption achieved with and without DPAP, respectively. (D) to (H): Under different pruning rates, the accuracy changes with the iteration process for different datasets. (I) to (M): The test accuracy achieved by DPAP with different pruning rates for different datasets.

could remarkably reduce energy consumption while improving convergence speed and accuracy for both DSNNs and DNNs¹.

A. Experimental settings

We introduce spatial pruning and temporal pruning in the temporal neuromorphic datasets (N-MNIST [49], DVS-Gesture [50]) for SNNs. The N-MNIST Dataset is captured by the neuromorphic vision sensor from original MNIST, and the DVS-Gesture dataset has 11 different gestures of 29 subjects captured by the DVS camera, with 1176 training samples and 280 testing samples. Moreover, we validate spatial pruning in the spatial datasets (MNIST [51], CIFAR-10 [52], ImageNet [53]) for SNNs and DNNs. The initial network structures used for the above tasks are shown in Table I.

The evaluation criteria of convergence speed is defined by the minimum epochs required to reach the highest accuracy of the initial baseline network. As in [54], the energy consumption of SNN is defined as:

$$E_{SNN} = FLOPS_{SNN} * E_{AC} * T \quad (14)$$

where E_{AC} is the energy consumption of accumulate (AC) operations and $FLOPS_{SNN}$ is the floating point operations per second. The energy consumption of SNN after pruning is:

$$E_{SNN}^p = (1 - \rho_w) * FLOPS_{SNN} * E_{AC} * (1 - \rho_t) * T \quad (15)$$

¹Our code is available at: https://github.com/BrainCog-X/BrainCog/tree/main/examples/Structural_Development/DPAP

where ρ_w and ρ_t are the weight sparsity and timestep sparsity respectively. For DNNs, the spatial weights sparsity represents the energy reduction.

B. DPAP reduces energy consumption, improves the performance, and speeds up the convergence rate of DSNNs

We first tested the effects of introducing DPAP on the accuracy (Fig.2 A), convergence speed (Fig.2 B) and energy consumption (Fig.2 C) of SNNs for three benchmark datasets: MNIST, N-MNIST, DVS-Gesture. From Fig.2 A and Fig.2 C, we found that introducing DPAP could slightly improve the accuracy (averaged by $\sim 0.31\%$) compared to the initial network without DPAP, while the networks are extremely compressed (averaged by $\sim 63\%$). Compared to the initial network without DPAP, the pruned network with DPAP helped to elevate the test accuracy from 99.46% to 99.59% with only 38.75% energy consumption for MNIST, and from 99.53% to 99.59% with 36.05% energy consumption for N-MNIST, and from 97.83% to 98.56% with 35.97% energy consumption for DVS-Gesture, respectively. These results also highlighted the superiority of our DPAP method on temporal neuromorphic datasets (such as DVS-Gesture). Besides, the convergence speeds of DSNNs using DPAP are significantly faster (speeds up by $\sim 1.4\times$ as shown in Fig.2 B) than that without DPAP. Especially on N-MNIST dataset, DPAP achieves more outstanding advantages in accelerating convergence (speeds up by $\sim 1.85\times$). In summary, DPAP could elevate the efficiency of DSNNs by extremely compressing the network (up to

TABLE II
PERFORMANCE COMPARISON FOR SNN ON TEMPORAL DATASETS N-MNIST AND DVS-GESTURE.

Dataset	Method	Structure	Pruning Methods	Weight sparsity	Timestep sparsity	Energy Consumption	Accuracy	Accuracy Loss
N-MNIST	ADMM-based [23]	LeNet-5	Spatial pruning	50.00%	0.00%	50.00%	98.34%	-0.61%
				75.00%	0.00%	25.00%	96.83%	-2.12%
	Grad R [22]	2 Conv 2 FC	Spatial pruning	65.00%	0.00%	35.00%	99.37%	0.54%
				75.00%	0.00%	25.00%	98.56%	-0.27%
	Our DPAP	2 Conv 2 FC	Spatial pruning	50.41%	0.00%	49.59%	99.55%	0.02%
				63.95%	0.00%	36.05%	99.59%	0.06%
DVS-Gesture	Our DPAP	2 Conv 2 FC	Spatial pruning	74.66%	0.00%	25.34%	99.56%	0.03%
				62.01%	53.39%	17.70%	99.53%	0.00%
	Deep R [22]	2 Conv 2 FC	Spatial pruning	70.63%	62.25%	11.09%	99.55%	0.02%
				50.00%	0.00%	50.00%	81.59%	-2.53%
	Grad R [22]	2 Conv 2 FC	Spatial pruning	75.00%	0.00%	25.00%	81.23%	-2.89%
				50.00%	0.00%	50.00%	84.12%	0.00%
	Our DPAP	2 Conv 2 FC	Spatial pruning	75.00%	0.00%	25.00%	91.95%	7.83%
				64.03%	0.00%	35.97%	98.56%	0.73%
	Our DPAP	2 Conv 2 FC	Spatial pruning	66.98%	0.00%	33.02%	98.20%	0.37%
				81.45%	0.00%	18.55%	97.83%	0.00%
			Spatial-temporal pruning	57.67%	49.93%	21.19%	98.20%	0.37%
				81.72%	67.20%	6.00%	98.20%	0.37%

64.03%) and speed up learning (up to $1.85\times$) with even relative accuracy improvement (up to 0.73%).

Furthermore, we compared the test accuracy of different datasets during learning by DPAP under different pruning rates and without DPAP (Fig.2 D-F after polynomial fit). Here, the different pruning rates are affected by two parameters (decay value ϵ and decay rate η) of the survival function in the DPAP method, where the faster neurons and synapses decay, the greater the pruning rate of the final network. Similar conclusions can be obtained with different datasets and pruning rates, that is, DPAP starts to make sense between 20-40 epochs, and then gradually widens the performance gap with the network without DPAP, and eventually achieves the highest performance between 60-80 epochs. Moreover, DPAP could achieve comparable or even better performance under different pruning rates, especially for the MNIST and N-MNIST datasets, which show more adaptability and stability to different pruning rates.

Fig.2 I-K illustrates the best performance achieved by DPAP with different pruning rates for different datasets. Unlike the conclusion found in most other works that the higher the pruning rate, the worse the performance, we observed that the accuracy peaks at about 60% pruning ratio. Actually, at age two or three, a child's brain has up to twice as many synapses as it will have in adulthood [2]. Synaptic pruning happens very quickly between ages 2 and 10 [58]. During this time, about 50 percent of the extra synapses are eliminated [59]. Subsequently, synaptic pruning continues through adolescence, but not as fast as before. The total number of synapses begins to stabilize [2]. In addition, “over-pruned” or “under-pruned” during brain development would lead to the occurrence of diseases, such as schizophrenia and autism spectrum disorders, respectively [60]–[62]. All these evidence reveals that our conclusions are more biologically reasonable, and closer to

the pruning mechanism of brain development.

C. Comparison with existing state-of-the-art SNNs compression algorithms on five benchmark datasets

Table. II and Table. III shows the comparison of the performance of different methods on temporal datasets and spatial datasets, respectively. For the temporal N-MNIST dataset, accuracy of spatial pruning drops when compressing the network by Grad R [22] and ADMM-based [23] pruning method, such as the accuracy reduces by 2.12% when the network is compressed by 75.00% for ADMM-based method [23]. Our DPAP method achieves 74.66% compression with even 0.03% relative accuracy improvement, and reaches the maximum accuracy of 99.59% with 63.95% pruning rate. When temporal pruning is added, we still have the accuracy up to 99.55% under 70.63% spatial weight sparsity and 62.25% timestep sparsity. According to Eq. 15, the energy required for the network after joint spatio-temporal pruning is only 11.09% of the basic dense network.

For the DVS-Gesture dataset, the accuracy is very sensitive to Deep R [22] pruning method, with 2.53% accuracy drops at 50.00% pruning rate, and with 2.89% accuracy drops at 75.00% pruning rate. Our DPAP could still improve the accuracy by 0.73% while compressing the network up to 64.03%. Moreover, even with only 18.55% connections, the accuracy of our method is still up to 97.83% (without any accuracy drop). With the similar spatial pruning rate of 81.72% and temporal pruning rate of 67.20%, the accuracy of joint spatio-temporal pruning is improved by 0.37% compared to only spatial pruning reaching 98.20%. Meanwhile, the network energy consumption is only 6.00% of the basic dense network. These experimental results demonstrate that our proposed temporal pruning working jointly with spatial pruning further

TABLE III
PERFORMANCE COMPARISON FOR SNN ON SPATIAL DATASETS.

Dataset	Method	Structure	Sparsity	Accuracy	AccLoss
	Online APTN [†] [55]	2 FC	90.00%	86.53%	-3.87%
	Threshold based [†] [28]	1 FC	70.00%	75.00%	-19.05%
	Threshold based [‡] [27]	2 FC	92.00%	91.50%	-1.70%
	Threshold based [‡] [56]	2 FC	74.00%	95.00%	-0.60%
MNIST	ADMM based [23]	LeNet-5	50.00%	99.10%	0.03%
			75.00%	96.84%	-2.23%
	Deep R [22]	2 FC	62.86%	98.56%	-0.36%
			86.70%	98.36%	-0.56%
	Grad R [22]	2 FC	74.29%	98.59%	-0.33%
			82.06%	98.49%	-0.43%
	DynSNN [30]	3 FC	57.40%	99.23%	-0.02%
	DynSNN [*] [30]	LeNet-5	61.50%	99.15%	-0.35%
	Our DPAP	2 FC	77.36%	98.74%	-0.07%
	Our DPAP	2Conv2FC	61.25%	99.59%	0.13%
	Our DPAP	2Conv2FC	84.22%	99.56%	0.10%
CIFAR-10	ADMM based [23]	7Conv2FC	40.00%	89.75%	0.18%
			60.00%	88.35%	-1.18%
	DynSNN [*] [30]	ResNet-20	37.13%	91.13%	-0.23%
	Grad R [22]	6Conv2FC	71.59%	92.54%	-0.30%
			87.96%	92.50%	-0.34%
	Our DPAP	6Conv2FC	33.46%	94.27%	-0.27%
			50.80%	93.83%	-0.71%
	ADMM based [23]	ResNet18	70.75%	59.48%	-3.74%
			78.96%	55.85%	-7.37%
ImageNet	Grad R [22]	ResNet18	50.94%	60.05%	-3.17%
			53.65%	24.62%	-38.60%
	Unstructured Pruning [57]	ResNet18	64.74%	61.89%	-1.29%
			72.26%	60.00%	-3.18%
	Our DPAP	ResNet18	22.69%	63.74%	-2.00%
			37.76%	63.35%	-2.39%
			51.71%	60.41%	-5.33%

Training methods: [†] STDP [‡] Event-driven CD ^{*} ANN-to-SNN
The rest use surrogate gradient.

reduces the network energy consumption through suppressing the spiking firing without affecting the network performance.

For the spatial MNIST dataset, other pruning methods compress the network at the cost of an accuracy drop. The ADMM-based pruning method [23] loses 2.23% accuracy when the network is compressed by 75.00%. Our method could maintain slight performance improvement at different pruning rates ranging from 48.42% to 84.22%. Especially when the pruning rate reaches 84.22%, our method can still achieve an accuracy improvement of 0.1% (accuracy is up to 99.56%). For the complex CIFAR-10 dataset, our DPAP method achieves 93.83% accuracy at 50.80% pruning rate (with 0.71% accuracy drop). Although there is a slight drop in accuracy, our method still achieves the highest accuracy of 94.27% at the pruning rate of 33.46%, outperforming the highest accuracy of 92.54% with Gard R [22] and the highest accuracy of 89.75% with ADMM-based [23] methods. For large-scale ImageNet dataset, our method achieves the outstanding accuracy of 63.35% with 37.76% pruning rate, which is higher than the next highest Unstructured Pruning [57]

TABLE IV
PERFORMANCE COMPARISON FOR DNN.

Dataset	Pruning Method	Structure	Sparsity	Accuracy	AccLoss
MNIST	BSP [18]	2FC	83.38%	95.94%	0.33%
	NP [12]	LeNet	59.62%	98.98%	-0.08%
	AL [63]	LeNet	90.51%	99.04%	-0.03%
	Our DPAP	2Conv2FC	56.00%	99.24%	0.04%
CIFAR-10	AFP-F [64]	VGG16	81.39%	92.87%	-0.05%
	AAP [65]	VGG16	70.04%	93.37%	-0.27%
	CP [66]	VGG16	50.00%	93.67%	-0.32%
	NS [67]	VGG16	51.00%	93.80%	-0.19%
	ThiNet [68]	VGG16	50.00%	93.85%	-0.14%
	Li-pruned [69]	VGG16	64.00%	94.40%	0.15%
ImageNet	Our DPAP	VGG16	41.54%	94.76%	0.23%
	Entropy [70]	ResNet50	35.00%	70.84%	-2.04%
	ThiNet [68]	ResNet50	56.00%	71.01%	-1.87%
	GAL [71]	ResNet50	16.90%	71.95%	-4.20%
	HRank [72]	ResNet50	62.00%	71.98%	-4.17%
	ABCPruner [73]	ResNet50	68.00%	72.58%	-3.42%
	SM [74]	ResNet50	82.00%	72.65%	-1.36%
	PS [75]	ResNet50	78.60%	72.80%	-4.40%
	Our DPAP	ResNet50	59.63%	73.26%	-1.00%

method of 1.46%. Under the pruning rate of 51.71%, DPAP still achieves a 60.41% accuracy, which improves by 0.36% compared to Grad R [22] with similar pruning rate.

To sum up, compared to other existing state-of-the-art SNNs compression algorithms, our pruning method shows obvious advantages, it can guarantee stable performance improvement under different pruning rates, and achieves SOTA effects on both performance and energy consumption for several benchmark datasets.

D. DPAP reduces energy consumption, improves the performance, and speeds up the convergence rate of DNNs

To verify the generality of our developmental plasticity-inspired pruning model, we also examined the effects of introducing DPAP to DNNs on spatial datasets. Fig.2 A-C also illustrates the accuracy, convergence speed and energy consumption of the network with DPAP and without DPAP for DNNs. Notably, DPAP could greatly compress the network (56.00% compressed for MNIST, 41.54% compressed for CIFAR-10) while relatively improving accuracy (0.04% for MNIST, 0.23% for CIFAR-10). In terms of convergence speed, for the MNIST dataset, the original network needs 50 epochs to reach the maximum accuracy of 99.20%, while the network with DPAP can achieve the same accuracy at the 39th epoch. The learning speed accelerated by 1.28×. Moreover, for the CIFAR-10 dataset, DPAP improves the convergence speed more than that without DPAP by 2.84×. Furthermore, we compared the curve of test accuracy with different pruning rates achieved by DPAP or without DPAP and found that DPAP can consistently improve the performance of the network (Fig.2 G,H after polynomial fit). Especially from 100-250 epochs, the accuracy of DNN with DPAP is significantly higher than the baseline DNN without DPAP.

With the datasets of MNIST and CIFAR-10, we also analyzed the accuracy of DNNs with DPAP under different pruning rates, as shown in Fig.2 L,M. The results obtained a

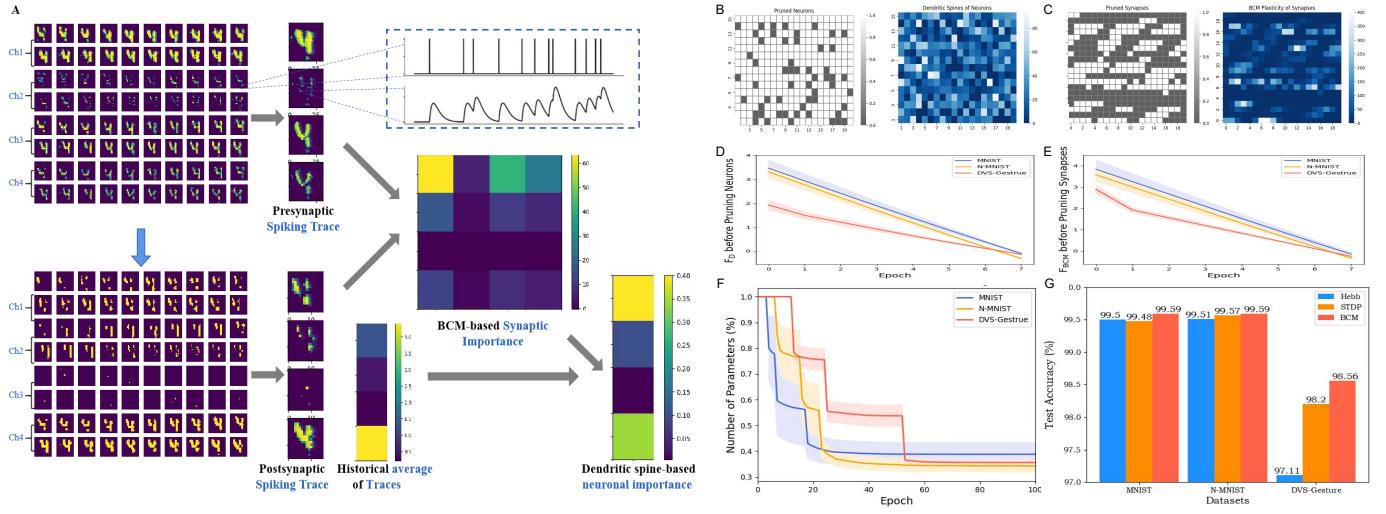


Fig. 3. Analyses and visualization of the biological plausibility of DPAP. (A): Visualization of temporal spikes, spiking traces, and the importance of spatial pruning computed from temporal spiking traces. Take for example the first four channels of the first and second convolutional layers of the N-MNIST network with the timesteps of 20. (B,C): The retained neurons and the average importance of all neurons (or synapses) The white squares represent pruned neurons and synapses. (D,E): During the 8 epochs before the neurons or synapses are pruned, the changing of survival function F_D and F_{BCM} . (F): The changing of network parameters during learning. (G): Test accuracy comparison of different synaptic plasticity used.

similar conclusion to DSNNs with DPAP, that as the pruning rate gradually increased, the accuracy showed a trend of increasing first and then decreasing, and formed a clear peak. Specifically, for the MNIST dataset, DNNs with DPAP reached the optimal balance of accuracy at 56.00% pruning rate. Besides, the accuracy consistently exceeds baseline levels from 44.00% to 67.17% of the pruning rate. Even with 67.17% pruning, DPAP can still achieve 99.21% accuracy (improved by 0.01%). For the CIFAR-10 dataset, DPAP can improve the accuracy of the network in the case of 36.31% and 41.54% compression, and achieves the highest performance of 94.76% at 41.54% pruning rate, while the performance is slightly lower (averaged by $\sim 0.04\%$) than the baseline at 26.71%, 47.78%, and 56.37% pruning rate (maintained an acceptable accuracy). As shown in Fig.2 G,H, the introduction of DPAP to DNNs can achieve optimal effects when the pruning rate is about 48.77%, which is consistent with the biological developmental mechanism. Furthermore, compared to other DNNs pruning algorithms under the same structure, our method achieves comparable performance. Particularly, for the large-scale dataset ImageNet, our DPAP method achieves an excellent 73.26% accuracy (slightly lower than baseline DNN 1.00%) at a biologically reasonable pruning rate of 59.63%. At similar pruning rates, the DPAP accuracy is improved by 2.25% and 1.28% compared to ThiNet [68] and HRank [72], respectively. Compared to the Pruning from scratch (PS) [75], which has the next highest accuracy, our DPAP improves by 0.46%, while reducing the pruning accuracy loss by 3.40%. These results also illustrate the effectiveness of introducing the proposed brain development-inspired pruning approach (DPAP) to DNNs.

In conclusion, introducing DPAP into DSNNs and DNNs could effectively improve the accuracy and learning speeds, and extremely compress the networks, showing its general effectiveness, high efficiency and flexible adaptability in var-

ious learning tasks and different network structure. More importantly, DPAP, as a brain developmental plasticity-inspired pruning approach, could reveal the brain developmental principle to a certain extent and reflect some characteristics during child development.

IV. DISCUSSION

In this study, we introduced biologically plausible developmental pruning mechanisms into DNNs and SNNs, and demonstrated that the proposed model could help optimize and compress efficient network architectures while improving performance and convergence speed for multiple benchmark datasets. To our best knowledge, this is the first work that studies a purely developmental plasticity-inspired pruning model that brings superior performance while also revealing the naturally occurring pruning processes during brain development. The proposed adaptive pruning strategy is consistent with the developmental pruning mechanism of the brain from multiple perspectives:

1) *Temporal characteristics capture for SNN*: We visualized the process of temporal spiking sequences acting in DPAP as shown in Fig.3 A. Comparison of the visualized images of spikes and spiking traces reveals that the neuron with a larger number of fired spikes in 20 timesteps correspond to more significant spiking trace. This demonstrates that the spiking trace can represent the activity level of neuron in the temporal dimension. According to the BCM synaptic plasticity theory, our DPAP method not only calculates the current spiking traces of neurons, but also considers the historical mean of neuronal spikes. As shown in Fig.3 A, both synaptic and neuronal pruning importance are calculated relying on the neuronal spiking traces. Hence, our SNN spatial pruning criterions are determined by the unique temporal spiking characteristics.

2) *Biologically plausible pruning criteria*: DPAP method employs local trace-based BCM synaptic plasticity as the

measure of synaptic importance. Dendritic spine dynamic plasticity incorporating neuronal activity traces and trace-based BCM synaptic plasticity is used to assess the importance of neurons. Such evaluation criteria are in line with the “activity-dependent, use it or lose it” developmental principle. Based on these pruning criteria, rarely used and unimportant synapses and neurons are pruned during learning. Fig.3 B,C shows the average importance of all neurons and synapses throughout the learning process. We found that after pruning, the retained neurons and synapses are more important, while relatively unimportant ones were eliminated.

3) Biologically plausible gradual decay or even death: DPAP prunes synapses or neurons is not an instantaneous impulsive decision, but a continuous and thoughtful evaluation. Pruning occurs after several successive gradual decays, where the survival functions for neurons (Fig.3 D) and synapses (Fig.3 E) gradually decline over a period of time before being pruned, which ensures that the pruned synapses or neurons are redundant. This is also consistent with the biological development that dendritic spine enlargement precedes growth, contraction precedes elimination, and synaptic decay precedes elimination [38].

4) Biologically plausible dynamic pruning: In the brain, pruning is an ongoing process that occurs concurrently with learning [76]. The pruning process is not arbitrary, but first drops sharply, then slowly decreases, and finally tends to be stable [2]. Specifically, dendritic spine elimination precedes synaptic pruning, and synaptic pruning precedes neural death [39], [40]. We examined whether the DPAP model can represent these dynamic phenomena in brain developmental pruning. Results are as expected the total number of connections in the network falls sharply at first and then gradually keeps steady during learning (see Fig.3 F). Furthermore, the average number of synapses contained in pruned neurons is 272, while in retained neurons is 298, which indicates that neurons with more pruned synapses are more likely to be deleted. Moreover, the shrinkage and elimination of dendritic spines result from the reduction of neuronal activity and synaptic plasticity, which also leads to the deletion of synapses. Therefore, we can conclude that the shrinkage and elimination of dendritic spines and synaptic pruning are prerequisites for neuronal pruning.

Effects of BCM synaptic plasticity A small number of brain-inspired SNNs pruning methods dynamically prune synapses with smaller weights or decayed weights in shallow SNNs based on the STDP measure of importance [27]–[29], [77]. Although STDP is a feasible and biologically realistic measure of importance, that depends on the spike timing difference of the pre-synaptic and post-synaptic neurons [78]–[80], it is not entirely consistent with the biological LTP and LTD. Unlike STDP and Hebbian [32] synaptic modification, BCM theory accounts for experience-dependent synaptic plasticity that could undergo both LTP or LTD depending on the level of post-synaptic response [41], [81]. There is substantial evidence both in the hippocampus and visual cortex that active synapses undergo LTD or LTP depending on the level of post-synaptic spiking which is consistent with the BCM theory [82]–[86]. To illustrate the superiority of BCM plasticity, we conducted experiments on SNNs pruning with

BCM, STDP, and Hebb plasticity, respectively. As shown in Fig.3 G, replacing BCM with Hebb and STDP will reduce the test accuracy for different datasets, while the more brain-like BCM used in our method achieves the best results.

V. CONCLUSION

In this paper, we demonstrated that introducing a generalized developmental pruning strategy helped to adaptively compress and optimize the network during the ongoing learning through pruning away redundancy. In contrast to existing pruning methods, our model incorporated multi-scale spatio-temporal developmental plasticity that led to a more compact and efficient network while improving the performance and convergence speed. The proposed method achieves the SOTA results on different datasets for DSNNs. More importantly, our approach shows highly biological plausibility and provides insight into naturally occurring pruning in the developing brain from multiple aspects. In addition, our DPAP algorithm provides a way to achieve the evolution from fixed hierarchical structures to brain-inspired neural circuits, enabling neurons to self-organize to form different neural circuits for performing different tasks. In the future, we expect further studies to combine developmental growth to realize the gradual evolution from small into a complex but efficient network that could adapt to the dynamic changing environment.

ACKNOWLEDGMENT

This work is supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB1010302), the National Natural Science Foundation of China (Grant No. 62106261).

REFERENCES

- [1] P. R. Huttenlocher, *Neural plasticity: The effects of environment on the development of the cerebral cortex*. Harvard University Press, 2009.
- [2] P. R. Huttenlocher *et al.*, “Synaptic density in human frontal cortex—developmental changes and effects of aging,” *Brain Res*, vol. 163, no. 2, pp. 195–205, 1979.
- [3] J. L. Elman, E. A. Bates, and M. H. Johnson, *Rethinking innateness: A connectionist perspective on development*. MIT press, 1996, vol. 10.
- [4] M. H. Johnson, “Functional brain development in humans,” *Nature Reviews Neuroscience*, vol. 2, no. 7, pp. 475–483, 2001.
- [5] S. Han, J. Pool, J. Tran, and W. J. Dally, “Learning both weights and connections for efficient neural networks,” in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. MIT Press, 2015, p. 1135–1143.
- [6] S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding,” *Fiber*, vol. 56, no. 4, pp. 3–7, 2015.
- [7] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, “Quantized neural networks: Training neural networks with low precision weights and activations,” *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6869–6898, 2017.
- [8] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *Computer Science*, vol. 14, no. 7, pp. 38–39, 2015.
- [9] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, “Pruning convolutional neural networks for resource efficient inference,” *arXiv preprint arXiv:1611.06440*, 2016.
- [10] B. Hassibi, D. G. Stork, and G. J. Wolff, “Optimal brain surgeon and general network pruning,” in *IEEE international conference on neural networks*. IEEE, 1993, pp. 293–299.
- [11] Z. Huang and N. Wang, “Data-driven sparse structure selection for deep neural networks,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 304–320.

- [12] S. Srinivas and R. V. Babu, "Data-free parameter pruning for deep neural networks," *arXiv preprint arXiv:1507.06149*, 2015.
- [13] Z. You, K. Yan, J. Ye, M. Ma, and P. Wang, "Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks," *Advances in neural information processing systems*, vol. 32, 2019.
- [14] M. Yang, M. Faraj, A. Hussein, and V. Gaudet, "Efficient hardware realization of convolutional neural networks using intra-kernel regular pruning," in *2018 IEEE 48th International Symposium on Multiple-Valued Logic (ISMVL)*. IEEE, 2018, pp. 180–185.
- [15] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," *Advances in neural information processing systems*, vol. 29, 2016.
- [16] F. Yu, K. Huang, M. Wang, Y. Cheng, W. Chu, and L. Cui, "Width & depth pruning for vision transformers," in *AAAI Conference on Artificial Intelligence (AAAI)*, vol. 2022, 2022.
- [17] F. Zhao, Y. Zeng, and J. Bai, "Toward a brain-inspired developmental neural network based on dendritic spine dynamics," *Neural Computation*, vol. 34, no. 1, pp. 172–189, 2022.
- [18] F. Zhao and Y. Zeng, "Dynamically optimizing network structure based on synaptic pruning in the brain," *Frontiers in Systems Neuroscience*, vol. 15, p. 55, 2021.
- [19] F. Zhao, T. Zhang, Y. Zeng, and B. Xu, "Towards a brain-inspired developmental neural network by adaptive synaptic pruning," in *International Conference on Neural Information Processing*. Springer, 2017, pp. 182–191.
- [20] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural networks*, vol. 10, no. 9, pp. 1659–1671, 1997.
- [21] W. Gerstner and W. M. Kistler, *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.
- [22] Y. Chen, Z. Yu, W. Fang, T. Huang, and Y. Tian, "Pruning of deep spiking neural networks through gradient rewiring," in *IJCAI*, 2021.
- [23] L. Deng, Y. Wu, Y. Hu, L. Liang, G. Li, X. Hu, Y. Ding, P. Li, and Y. Xie, "Comprehensive snn compression using admm optimization and activity regularization," *IEEE transactions on neural networks and learning systems*, no. 99, pp. 1–15, 2021.
- [24] J. Chen, H. Yuan, J. Tan, B. Chen, C. Song, and D. Zhang, "Resource constrained model compression via minimax optimization for spiking neural networks," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 5204–5213.
- [25] L. Meng, G. Qiao, X. Zhang, J. Bai, Y. Zuo, P. Zhou, Y. Liu, and S. Hu, "An efficient pruning and fine-tuning method for deep spiking neural network," *Applied Intelligence*, vol. 53, no. 23, pp. 28910–28923, 2023.
- [26] B. Chakraborty, B. Kang, H. Kumar, and S. Mukhopadhyay, "Sparse spiking neural network: Exploiting heterogeneity in timescales for pruning recurrent snn," *arXiv preprint arXiv:2403.03409*, 2024.
- [27] N. Rathi, P. Panda, and K. Roy, "Stdsp-based pruning of connections and weight quantization in spiking neural networks for energy-efficient recognition," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 4, pp. 668–677, 2018.
- [28] Y. Shi, L. Nguyen, S. Oh, X. Liu, and D. Kuzum, "A soft-pruning method applied during training of spiking neural networks for in-memory computing applications," *Frontiers in neuroscience*, vol. 13, p. 405, 2019.
- [29] T. N. Nguyen, B. Veeravalli, and X. Fong, "Connection pruning for deep spiking neural networks with on-chip learning," in *International Conference on Neuromorphic Systems 2021*, 2021, pp. 1–8.
- [30] F. Liu, W. Zhao, Y. Chen, Z. Wang, and F. Dai, "Dynsnn: A dynamic approach to reduce redundancy in spiking neural networks," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 2130–2134.
- [31] E. G. Gray, "Axo-somatic and axo-dendritic synapses of the cerebral cortex: an electron microscope study," *Journal of anatomy*, vol. 93, no. Pt 4, p. 420, 1959.
- [32] D. O. Hebb, "The first stage of perception: growth of the assembly," *The Organization of Behavior*, vol. 4, pp. 60–78, 1949.
- [33] I. Skaliola, "Experience-dependent plasticity in the developing brain," in *International Congress Series*, vol. 1241. Elsevier, 2002, pp. 313–320.
- [34] J. T. Bruer, "Neural connections: Some you use, some you lose," *The Phi Delta Kappan*, vol. 81, no. 4, pp. 264–277, 1999.
- [35] N. Toni, P.-A. Buchs, I. Nikonenko, C. Bron, and D. Muller, "Ltp promotes formation of multiple spine synapses between a single axon terminal and a dendrite," *Nature*, vol. 402, no. 6760, pp. 421–425, 1999.
- [36] N. Becker, C. J. Wierenga, R. Fonseca, T. Bonhoeffer, and U. V. Nägerl, "Ltp induction causes morphological changes of presynaptic boutons and reduces their contacts with spines," *Neuron*, vol. 60, no. 4, pp. 590–597, 2008.
- [37] H. T. Chugani, "Neuroimaging of developmental nonlinearity and developmental pathologies," *Developmental neuroimaging: Mapping the development of brain and behavior*, pp. 187–195, 1996.
- [38] H. Colman, J. Nabekura, and J. Lichtman, "Alterations in synaptic strength preceding axon withdrawal," *Science*, vol. 275, no. 5298, pp. 356–361, 1997.
- [39] J. T. Trachtenberg, B. E. Chen, G. W. Knott, G. Feng, J. R. Sanes, E. Welker, and K. Svoboda, "Long-term in vivo imaging of experience-dependent synaptic plasticity in adult cortex," *Nature*, vol. 420, no. 6917, pp. 788–794, 2002.
- [40] S. Furber, R. W. Oppenheim, and D. Prevette, "Naturally-occurring neuron death in the ciliary ganglion of the chick embryo following removal of preganglionic input: evidence for the role of afferents in ganglion cell survival," *Journal of Neuroscience*, vol. 7, no. 6, pp. 1816–1832, 1987.
- [41] E. L. Bienenstock, L. N. Cooper, and P. W. Munro, "Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex," *Journal of Neuroscience*, vol. 2, no. 1, pp. 32–48, 1982.
- [42] J.-P. Pfister and W. Gerstner, "Triplets of spikes in a model of spike timing-dependent plasticity," *Journal of Neuroscience*, vol. 26, no. 38, pp. 9673–9682, 2006.
- [43] K. Yamazaki, V.-K. Vo-Ho, D. Bulsara, and N. Le, "Spiking neural networks and their applications: A review," *Brain Sciences*, vol. 12, no. 7, p. 863, 2022.
- [44] L. F. Abbott, "Lapicque's introduction of the integrate-and-fire model neuron (1907)," *Brain Research Bulletin*, vol. 50, pp. 303–304, 1999.
- [45] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *Frontiers in neuroscience*, vol. 12, p. 331, 2018.
- [46] Y. Zeng, D. Zhao, F. Zhao, G. Shen, Y. Dong, E. Lu, Q. Zhang, Y. Sun, Q. Liang, Y. Zhao *et al.*, "Braincog: A spiking neural network based brain-inspired cognitive intelligence engine for brain-inspired ai and brain simulation," *arXiv preprint arXiv:2207.08533*, 2022.
- [47] J. E. Raymond, K. L. Shapiro, and K. M. Arnell, "Temporary suppression of visual processing in an rsvp task: An attentional blink?" *Journal of experimental psychology: Human perception and performance*, vol. 18, no. 3, p. 849, 1992.
- [48] J. T. Coull, "fmri studies of temporal attention: allocating attention within, or towards, time," *Cognitive Brain Research*, vol. 21, no. 2, pp. 216–226, 2004.
- [49] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Frontiers in neuroscience*, vol. 9, p. 437, 2015.
- [50] A. Amir, B. Taba, D. Berg, T. Melano, McKinstry *et al.*, "A low power, fully event-based gesture recognition system," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7243–7252.
- [51] Y. LeCun, "The mnist database of handwritten digits," <http://yann.lecun.com/exdb/mnist/>, 1998.
- [52] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
- [53] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [54] B. Chakraborty, X. She, and S. Mukhopadhyay, "A fully spiking hybrid neural network for energy-efficient object detection," *IEEE Transactions on Image Processing*, vol. 30, pp. 9014–9029, 2021.
- [55] W. Guo, M. E. Fouad, H. E. Yantir, A. M. Eltawil, and K. N. Salama, "Unsupervised adaptive weight pruning for energy-efficient neuromorphic systems," *Frontiers in Neuroscience*, p. 1189, 2020.
- [56] E. O. Neftci, B. U. Pedroni, S. Joshi, M. Al-Shedivat, and G. Cauwenberghs, "Stochastic synapses enable efficient brain-inspired learning machines," *Frontiers in neuroscience*, vol. 10, p. 241, 2016.
- [57] X. Shi, J. Ding, Z. Hao, and Z. Yu, "Towards energy efficient spiking neural networks: An unstructured pruning framework," in *The Twelfth International Conference on Learning Representations*, 2023.
- [58] P. R. Huttenlocher, "Synaptogenesis, synapse elimination, and neural plasticity in human cerebral cortex." 1994.
- [59] ———, "Morphometric study of human cerebral cortex development," *Neuropsychologia*, vol. 28, no. 6, pp. 517–527, 1990.
- [60] P. Penzes, M. E. Cahill, K. A. Jones, J.-E. VanLeeuwen, and K. M. Woolfrey, "Dendritic spine pathology in neuropsychiatric disorders," *Nature neuroscience*, vol. 14, no. 3, pp. 285–293, 2011.
- [61] J. R. Glausier and D. A. Lewis, "Dendritic spine pathology in schizophrenia," *Neuroscience*, vol. 251, pp. 90–107, 2013.

- [62] J. J. Hutsler and H. Zhang, "Increased dendritic spine densities on cortical projection neurons in autism spectrum disorders," *Brain research*, vol. 1309, pp. 83–94, 2010.
- [63] S. Srinivas and R. V. Babu, "Learning the architecture of deep neural networks," in *Proceedings of the British Machine Vision Conference (BMVC)*, pages, 2016, pp. 104–1.
- [64] X. Ding, G. Ding, J. Han, and S. Tang, "Auto-balanced filter pruning for efficient convolutional neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [65] K. Zhao, A. Jain, and M. Zhao, "Automatic attention pruning: Improving and automating model pruning using attentions," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023, pp. 10470–10486.
- [66] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1389–1397.
- [67] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2736–2744.
- [68] J.-H. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5058–5066.
- [69] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," *arXiv preprint arXiv:1608.08710*, 2016.
- [70] J.-H. Luo and J. Wu, "An entropy-based pruning method for cnn compression," *arXiv preprint arXiv:1706.05791*, 2017.
- [71] S. Lin, R. Ji, C. Yan, B. Zhang, L. Cao, Q. Ye, F. Huang, and D. Doermann, "Towards optimal structured cnn pruning via generative adversarial learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2790–2799.
- [72] M. Lin, R. Ji, Y. Wang, Y. Zhang, B. Zhang, Y. Tian, and L. Shao, "Hrank: Filter pruning using high-rank feature map," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1529–1538.
- [73] M. Lin, R. Ji, Y. Zhang, B. Zhang, Y. Wu, and Y. Tian, "Channel pruning via automatic structure search," *arXiv preprint arXiv:2001.08565*, 2020.
- [74] T. Dettmers and L. Zettlemoyer, "Sparse networks from scratch: Faster training without losing performance," 2020.
- [75] Y. Wang, X. Zhang, L. Xie, J. Zhou, H. Su, B. Zhang, and X. Hu, "Pruning from scratch," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 12273–12280.
- [76] P. Rakic, J.-P. Bourgeois, M. F. Eckenhoff, N. Zecevic, and P. S. Goldman-Rakic, "Concurrent overproduction of synapses in diverse regions of the primate cerebral cortex," *Science*, vol. 232, no. 4747, pp. 232–235, 1986.
- [77] Y. Qi, J. Shen, Y. Wang, H. Tang, H. Yu, Z. Wu, and G. Pan, "Jointly learning network connections and link weights in spiking neural networks," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 1597–1603.
- [78] M.-M. Poo, "Spike timing-dependent plasticity: Hebb's postulate revisited," *International Journal of Developmental Neuroscience*, vol. 26, no. 8, pp. 827–828, 2008.
- [79] C. C. Bell, V. Z. Han, Y. Sugawara, and K. Grant, "Synaptic plasticity in a cerebellum-like structure depends on temporal order," *Nature*, vol. 387, no. 6630, pp. 278–281, 1997.
- [80] W. Gerstner, R. Kempter, J. L. V. Hemmen, and H. Wagner, "A neuronal learning rule for sub-millisecond temporal coding," *Nature*, vol. 383, no. 6595, pp. 76–78, 1996.
- [81] A. Kirkwood, M. G. Rioult, and M. F. Bear, "Experience-dependent modification of synaptic plasticity in visual cortex," *Nature*, vol. 381, no. 6582, pp. 526–528, 1996.
- [82] M. F. Bear, L. N. Cooper, and F. F. Ebner, "A physiological basis for a theory of synapse modification," *Science*, vol. 237, no. 4810, pp. 42–48, 1987.
- [83] W. C. Abraham, S. E. Mason-Parker, M. F. Bear, S. Webb, and W. P. Tate, "Heterosynaptic metaplasticity in the hippocampus *in vivo*: a bcm-like modifiable threshold for ltp," *Proceedings of the National Academy of Sciences*, vol. 98, no. 19, pp. 10924–10929, 2001.
- [84] S. M. Dudek and M. F. Bear, "Homosynaptic long-term depression in area cal of hippocampus and effects of n-methyl-d-aspartate receptor blockade," *Proceedings of the National Academy of Sciences*, vol. 89, no. 10, pp. 4363–4367, 1992.
- [85] A. Kirkwood and M. F. Bear, "Homosynaptic long-term depression in the visual cortex," *Journal of Neuroscience*, vol. 14, no. 5, pp. 3404–3412, 1994.
- [86] A. Artola, S. Bröcher, and W. Singer, "Different voltage-dependent thresholds for inducing long-term depression and long-term potentiation in slices of rat visual cortex," *Nature*, vol. 347, no. 6288, pp. 69–72, 1990.



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



Bing Han received the B.Eng. degree in intelligent science and technology from University of Science and Technology Beijing, Beijing, China, in 2021. Now she is the Ph.D. candidate in the Brain-inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences, supervised by Prof. Yi Zeng. Her current research interests are brain-inspired structural development algorithms for spiking neural networks.



Feifei Zhao is currently an Associate Professor in the Brain-inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences(CASIA), China. She received the Ph.D. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2019. Her current research interests include Brain-inspired Developmental and Evolutionary Spiking Neural Networks, Empathy driven AI Ethics and Safety.



Guobin Shen received his B.Eng. degree from Sun Yat-sen University in Guangzhou, Guangdong, China. He is now a PhD candidate in the Brain-inspired Cognitive Intelligence Lab, at the Institute of Automation, Chinese Academy of Sciences, under the supervision of Prof. Yi Zeng. His research focuses on biologically-inspired learning algorithms and spiking neural network architecture design and training strategies.