# Enhancing Efficient Continual Learning with Dynamic Structure Development of Spiking Neural Networks

**Bing Han**[1,2], **Feifei Zhao**[1], **Yi Zeng**[1,2,3,4*], **Wenxuan Pan** [1,2] and **Guobin Shen** [1,3]
[1]Brain-inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3] School of Future Technology, University of Chinese Academy of Sciences
[4]Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences
{hanbing2021, zhaofeifei2014, yi.zeng, panwenxuan2020, shenguobin2021}@ia.ac.cn

## Abstract

Children possess the ability to learn multiple cognitive tasks sequentially, which is a major challenge toward the long-term goal of artificial general intelligence. Existing continual learning frameworks are usually applicable to Deep Neural Networks (DNNs) and lack the exploration on more brain-inspired, energy-efficient Spiking Neural Networks (SNNs). Drawing on continual learning mechanisms during child growth and development, we propose Dynamic Structure Development of Spiking Neural Networks (DSD-SNN) for efficient and adaptive continual learning. When learning a sequence of tasks, the DSD-SNN dynamically assigns and grows new neurons to new tasks and prunes redundant neurons, thereby increasing memory capacity and reducing computational overhead. In addition, the overlapping shared structure helps to quickly leverage all acquired knowledge to new tasks, empowering a single network capable of supporting multiple incremental tasks (without the separate sub-network mask for each task). We validate the effectiveness of the proposed model on multiple class incremental learning and task incremental learning benchmarks. Extensive experiments demonstrated that our model could significantly improve performance, learning speed and memory capacity, and reduce computational overhead. Besides, our DSD-SNN model achieves comparable performance with the DNNs-based methods, and significantly outperforms the state-of-the-art (SOTA) performance for existing SNNs-based continual learning methods.

## 1 Introduction

Children are able to incrementally learn new tasks to acquire new knowledge, however, this is a major challenge for Deep Neural Networks (DNNs) and Spiking Neural Networks (SNNs). When learning a series of different tasks sequentially, DNNs and SNNs forget the previously acquired knowledge and fall into catastrophic forgetting [French, 1999]. Despite some preliminary solutions that have recently been proposed for DNNs-based continual learning, there is still a lack of in-depth inspiration from brain continual learning mechanisms and exploration on SNNs-based models.

The studies attempt to address the continual learning problem of DNNs under task incremental learning (recognition within the classes of a known task) and class incremental learning (recognition within all learned classes) scenarios. Related works can be roughly divided into three categories:

**a) Regularization.** Employing maximum a posterior estimation minimizes the changes of important weights [Li and Hoiem, 2017; Kirkpatrick *et al.*, 2017; Zenke *et al.*, 2017]. These methods require strong model assumptions, such as the EWC [Kirkpatrick *et al.*, 2017] supposing that new weights are updated to local regions of the previous task weights, which are highly mathematical abstractions and poorly biologically plausibility.

**b) Replay and retrospection.** Reviewing a portion of the samples of the old tasks while learning the new task [Lopez-Paz and Ranzato, 2017; van de Ven *et al.*, 2020; Kemker and Kanan, 2018], is currently considered as the superior class incremental learning method. The samples of old tasks are stored in additional memory space or generated by additional generation networks, resulting in extra consumption.

**c) Dynamic network structure expansion.** [Rusu *et al.*, 2016; Siddiqui and Park, 2021] proposed progressive neural networks that extend a new network for each task, causing a linear increase in network scale. To reduce network consumption, a sub-network of the whole is selected for each task using pruning and growth algorithms [Yoon *et al.*, 2018; Dekhovich *et al.*, 2023], evolutionary algorithms [Fernando *et al.*, 2017] or reinforcement learning (RL) algorithms [Xu and Zhu, 2018; Gao *et al.*, 2022]. However, these methods require storing a mask for each sub-network, which to some extent amounts to storing a separate network for each task, rather than a brain-inspired overall network capable of performing multiple sequential tasks simultaneously.

To the best of our knowledge, there is little research on SNNs-based continual learning. Spiking neural networks, as third-generation neural networks [Maass, 1997; Zhao *et al.*, 2022b], simulate the information processing mechanisms of

*Corresponding author.

the brain, and thus serve well as an appropriate level of abstraction for integrating inspirations from brain multi-scale biological plasticity to achieve child-like continual learning. The existing HMN algorithm [Zhao *et al.*, 2022c] uses a DNN network to decide the sub-network of SNN for each task, and is only applicable to two-layer fully connected networks for the N-MNIST dataset. There is still a lack of SNNs-based continual learning methods that could incorporate in-depth inspiration from the brain's continual learning mechanisms, while achieving comparable performance with DNNs under complex continual learning scenarios.

Structural development mechanisms allow the brain's nervous system to dynamically expand and contract, as well as flexibly allocate and invoke neural circuits for efficient continual learning [Silva *et al.*, 2009]. Motivated by this, this paper proposes Dynamic Structure Development of Spiking Neural Networks (DSD-SNN) for efficient and adaptive continual learning. DSD-SNN is designed as an SNN architecture that can be dynamically expanded and compressed, empowering a single network to learn multiple incremental tasks simultaneously, overcoming the problem of needing to assign masks to each task faced by DNNs-based continual learning methods. We validate the effectiveness of our proposed model on multiple class incremental learning (CIL) and task incremental learning (TIL) benchmarks, achieving comparable or better performance on MNIST, N-MNIST, and CIFAR-100 datasets. Especially, the proposed DSD-SNN model achieves an accuracy of 77.92% $\pm$ 0.29% on CIFAR100, only using 37.48% of the network parameters.

The main contributions of this paper can be summarized as follows:

- DSD-SNN dynamically grows new neurons to learn newly arrived tasks, while extremely compressing the network to increase memory capacity and reduce computational overhead.
- DSD-SNN maximally utilizes the previously learned tasks to help quickly adapt and infer new tasks, enabling efficient and adaptive continual learning (no need to identify separate sub-network mask for each task).
- The experimental results demonstrate the remarkable superiority of DSD-SNN model on performance, learning speed, memory capacity and computational overhead compared with the state-of-the-art (SOTA) SNNs-based continual learning algorithms, and comparable performance with DNNs-based continual learning algorithms.

## 2 Related Work

This paper mainly focuses on dynamic network structure expansion algorithms based on structural plasticity, which can be divided into progressive neural networks (PNN) and sub-network selection algorithms. In fact, the existing network structure expansion algorithms are mostly DNNs-based continual learning, with little exploration on SNNs.

**Progressive neural networks.** [Rusu *et al.*, 2016] first proposes the progressive neural network and applies it to multiple continual reinforcement learning tasks. The PNN expands a new complete network for each new task and fixes the networks of the old tasks. In addition, lateral connections are introduced between the networks to effectively leverage the knowledge already learned. PIL [Siddiqui and Park, 2021] extends the PNN to large-scale convolutional neural networks for image classification tasks. However, the PNN algorithms extremely increase the network storage and computational consumption during continual learning. In contrast, as development matures and cognition improves, the number of brain synapses decreases by more than 50% [Huttenlocher, 1990], forming a highly sparse brain structure perfect for continual learning. The PNN blindly expand the structure causing catastrophic effects in the case of massive sequential tasks.

**Sub-network selection algorithm.** A part of the network nodes is selected to be activated for a given task. PathNet [Fernando *et al.*, 2017] is first proposed to select path nodes (each node contains a set of neurons) for each task using the genetic algorithm. RPS-Net [Rajasegaran *et al.*, 2019a] randomly activates multiple input-to-output paths connected by convolutional blocks, and chooses the highest-performing ones as the final path. In addition, RCL [Xu and Zhu, 2018] employ additional RL networks to learn the number of neurons required for a new task, while CLEAS [Gao *et al.*, 2022] uses RL to directly determine the activation and death of each neuron. HMN [Zhao *et al.*, 2022c] uses a hybrid network learning framework that uses an ANN modulation network to determine the activation of neurons for a SNN prediction network, but is only applied to small-scale networks for simple scenarios. A sub-network mask learning process based on pruning strategy is proposed by [Dekhovich *et al.*, 2023], which is applied to CIL combined with the replay strategy. The above algorithms select sub-networks for each task separately, failing to maximize the reuse of acquired knowledge to support new task learning.

To solve this problem, DER [Yan *et al.*, 2021] prunes a sparse convolutional feature extractor for each task, and then merges the output of the convolution extractor into the previous tasks. CLNP [Golkar *et al.*, 2020] grows new neurons for a new task based on the old network, and DEN [Yoon *et al.*, 2018] expands when the already learned network is insufficient for the new task, while reusing the existing neurons. These several works require storing an additional sub-network mask for each task, which both increases additional storage consumption and is not consistent with the overall developmental learning process of the brain.

Considering the various limitations of existing works above, the DSD-SNN proposed in this paper, which is a pioneering algorithm on SNNs-based continual learning, enables the capacity of a single network to learn multiple sequential tasks simultaneously, while reusing the acquired knowledge and significantly increasing the memory capacity.

## 3 Method

### 3.1 Continual Learning Definition

We are expected to sequentially learn $\Gamma$ tasks, $\Gamma = \{T_1, ..., T_N\}$. Each task $T_i$ takes the form of a classification problem with its own dataset: $D_{T_i} = \{(x_j, y_j)\}_{j=1}^{N_{T_i}}$, where

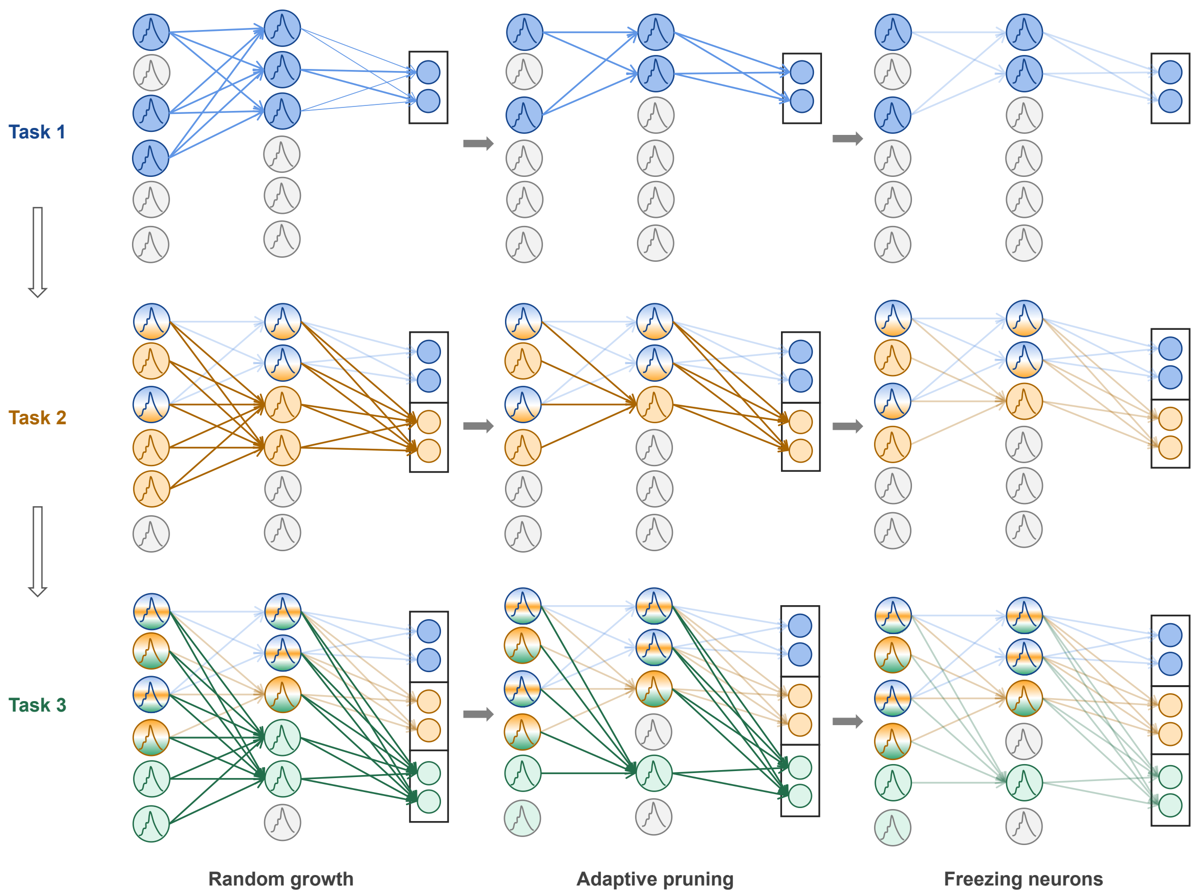


Figure 1: The DSD-SNN model realizes multi-task incremental learning through random growth, adaptive pruning, and freezing of neurons.

$x_j \in \chi, y_i \in \{1, ..., C_{T_i}\}$, $\chi$ is the input image space, $N_{T_i}$ and $C_{T_i}$ are the number of samples and classes of task $T_i$. For the task incremental learning scenario, $T_i$ is knowable in the testing process, setting requires to optimize:

$$\max_{\theta} E_{T_i \sim \Gamma}[E_{(x_j,y_j)\sim T_i}[logp_\theta(y_j|x_j, T_i)]] \quad (1)$$

where $\theta$ is the network parameters. When $T_i$ is unknown in testing, more complex class incremental learning scenarios solve the following problems:

$$\max_{\theta} E_{T_i \sim \Gamma}[E_{(x_j,y_j)\sim T_i}[logp_\theta(y_j|x_j)]] \quad (2)$$

### 3.2 DSD-SNN Architecture

The design of the DSD-SNN algorithm is inspired by the dynamic allocation, reorganization, growth, and pruning of neurons during efficient continual learning in the brain. As depicted in Fig. 1, the proposed DSD-SNN model includes three modules (random growth, adaptive pruning, freezing neurons) to accomplish multi-task incremental learning.

**Random growth.** When a new task is coming, the DSD-SNN model first randomly assigns and grows a portion of untrained empty neurons to form a new pathway. And the new task-related classification neurons are added to the output layer as shown in Fig. 1. Newly grown neurons receive the output of all non-empty neurons of the previous layer (both newly grown neurons and already frozen neurons in the previous tasks). Therefore, all features learned from previous tasks can be captured and reused by the neural pathways of the new task. Then, the DSD-SNN algorithm can take full advantage of the features learned from the previous task to help the new task converge quickly, while the newly grown neurons can also focus on learning features specific to the new task.

**Adaptive pruning.** During the learning process of the current task, the DSD-SNN algorithm adaptively detects relatively inactive neurons in the current pathway based on synaptic activity and prunes those redundant neurons to save resources. The pruned neurons are re-initialized as empty neurons that can be assigned to play a more important role in future tasks. Pruning only targets those neurons that are newly grown for the current task and does not include neurons that were frozen in the previous tasks. Adaptive pruning can substantially expand the memory capacity of the network to learn and memorize more tasks under a fixed scale.

**Freezing neurons.** The contributing neurons that are retained after pruning will be frozen, enabling the DSD-SNN model to learn new tasks without forgetting the old tasks. The frozen neurons can be connected to newly grown neurons to provide acquired knowledge. During the training of new

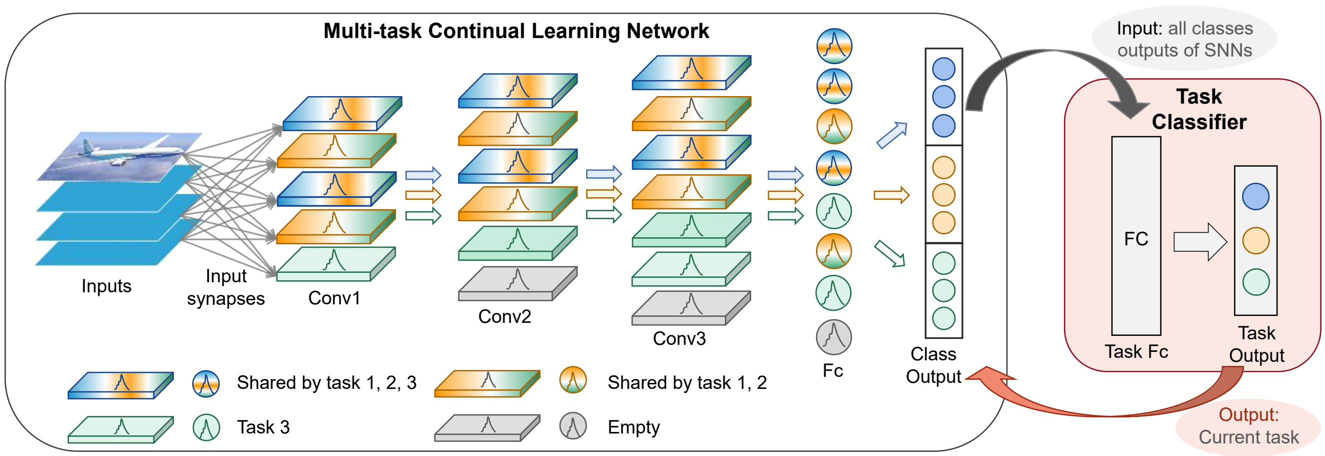


Figure 2: The architecture of the DSD-SNN model.

task $T_i$, all input synapses of the frozen neuron are no longer updated, only the newly added output synapses to the new neurons can be updated. The DSD-SNN model with neuron growth, pruning, and freezing can memorize previous knowledge and reuse the acquired knowledge to learn new tasks for efficient continual learning.

The deep SNN with multiple convolutional and fully connected layers is constructed to implement task incremental learning and class incremental learning, as shown in Fig. 2. During the training process, we sequentially input training samples of each task and update the synapses newly added to the network. In the testing process, test samples of all learned tasks are fed into our overall multi-task continual learning network, so that a single DSD-SNN model can achieve all tasks without the need to identify separate sub-network mask for each task.

To address more complex class incremental learning, we add a two-layer network as the task classifier. The task classifier receives inputs from the classes outputs of the continual learning network, and outputs which task the current sample belongs to (as in the red box in Fig. 2). According to the inferred task $\hat{T}_i$ obtained from the task classifier, the DSD-SNN model chooses the maximum output class of the $\hat{T}_i$ task in the continual learning network as the predicted class.

## 3.3 DSD-SNN Computational Details

So far in this section, we have described how our model efficiently and adaptively accomplishes continual learning. We now introduce the detailed growth and pruning scheme that we use throughout this paper.

### Neuronal Growth and Allocation

During brain development, neurons and synapses are first randomly and excessively grown and then reshaped based on the external experience [Jun and Jin, 2007; Elman *et al.*, 1996]. In the DSD-SNN model, the SNN is first initialized to consist of $N^l$ neurons in each layer $l$. In the beginning, all neurons in the network are unassigned empty neurons $N_{empty}$. When the new task $T_i$ arrives, we randomly grow $\rho\% \times N^l$ neurons from the empty neurons for each layer, denoted as $N_{new}$. After training and pruning for task $T_i$ , all retained neurons $N_{new}$ are frozen, added to $N_{frozen}$.

To better utilize the acquired knowledge, the newly grown neurons $N^l_{new}$ in each layer not only receive the output of the new growth neurons $N^{l-1}_{new}$ in the previous layer, but also receive the output of the frozen neurons $N^{l-1}_{frozen}$ in the previous layer, as follows.

$$\{N^{l-1}_{frozen}, N^{l-1}_{new}\} \rightarrow N^l_{new} \tag{3}$$

Where $\rightarrow$ represents the input connections. For the frozen neurons $N^{l-1}_{frozen}$, growth does not add input connections to avoid interference with the memory of previous tasks.

Note that we do not assign task labels to frozen and new growth neurons in either the training or testing phase of continual learning. That is, the DSD-SNN algorithm uses the entire network containing all neurons that have learned previous tasks to do prediction and inference. Thus, our model is able to learn multiple sequential tasks simultaneously without storing separate sub-network masks.

### Neuronal Pruning and Deactivation

Neuroscience researches have demonstrated that after the overgrowth in infancy, the brain network undergoes a long pruning process in adolescence, gradually emerging into a delicate and sparse network [Huttenlocher and others, 1979; Huttenlocher, 1990; Zhao *et al.*, 2022a]. Among them, input synapses are important factors to determine the survival of neurons according to the principle of "use it or lose it" [Furber *et al.*, 1987; Bruer, 1999; Zhao and Zeng, 2021]. For SNN, neurons with input synapse weights close to 0 are more difficult to accumulate membrane potentials beyond the spiking threshold, resulting in firing spikes less and contributing to the outputs less. Therefore, we used the sum of input synapse weights $S^l_i$ to assess the importance of neurons $i$ in the $l$ layer as in Eq. 4.

$$S^l_i = \sum_{j=1}^{M_{l-1}} W_{ij} \tag{4}$$

Where $W_{ij}$ is the synapse weights from presynaptic neuron $j$ to postsynaptic neuron $i$, $M_{l-1}$ is the number of presynaptic neurons.

During the training of new tasks, we monitor the importance of newly grown neurons $N_{new}$ and prune redundant neurons whose values of $S_i$ are continuously getting smaller. Here, we define a pruning function as follows:

$$\phi_{P_i^l} = \alpha * Norm(S_i^l) - \rho_p \tag{5}$$

$$P_i^l = \gamma P_i^l + e^{-\frac{epoch}{\eta}} \phi_{P_i^l} \tag{6}$$

Where $Norm(S_i^l)$ refers to linearly normalize $S_i^l$ to $0 \sim 1$. $\alpha = 2$ and $\rho_p$ control the pruning strength. $\rho_p$ includes $\rho_c$ and $\rho_f$ for the convolutional and fully connected layers, respectively. $P_i^l$ is initialized to 5. $\gamma = 0.99$ and $\eta$ controls the update rate as [Han *et al.*, 2022b]. $e^{-\frac{epoch}{\eta}}$ decreases exponentially with increasing epoch, which is consistent with the speed of the pruning process in biological neural networks that are first fast, then slow, and finally stable [Huttenlocher and others, 1979; Han *et al.*, 2022a].

The pruning functions are updated at each epoch, then we prune neurons with the pruning function $P_i^l < 0$. We structurally prune channels in the convolutional layer and prune neurons in the fully connected layer, removing their input connections and output connections.

### 3.4 SNNs Information Transmission

Different from DNNs, SNNs use spiking neurons with discrete 0/1 output, which are able to integrate spatio-temporal information. Specifically, we employ the leaky integrate-and-fire (LIF) neuron model [Abbott, 1999] to transmit and memorize information. In the spatial dimension, LIF neurons integrate the output of neurons in the previous layer through input synapses. In the temporal dimension, LIF neurons accumulate membrane potentials from previous time steps via internal decay constants $\tau$. Incorporating the spatio-temporal information, the LIF neuron membrane potential $U_i^{t,l}$ at time step $t$ is updated by the following equation:

$$U_i^{t,l} = \tau(1 - U_i^{t-1,l}) + \sum_{j=1}^{M_{l-1}} W_{ij} O_j^{t,l-1} \tag{7}$$

When the neuronal membrane potential exceeds the firing threshold $V_{th}$, the neuron fires spike, and its output $O_i^{t,l}$ is equal to 1; Conversely, the neuron outputs 0. The discrete spiking outputs of LIF neurons conserve consumption as the biological brain, but hinder gradient-based backpropagation. To address this problem, [Wu *et al.*, 2018] first proposed the method of surrogate gradient. In this paper, we use Qgategrad [Qin *et al.*, 2020] surrogate gradient method with constant $\lambda = 2$ to approximate the spiking gradient, as follows:

$$\frac{O_i^{t,l}}{U_i^{t,l}} = \begin{cases} 0, & |U_i^{t,l}| > \frac{1}{\lambda} \\ -\lambda^2 |U_i^{t,l}| + \lambda, & |U_i^{t,l}| \le \frac{1}{\lambda} \end{cases} \tag{8}$$

Overall, We present the specific procedure of our DSD-SNN algorithm as Algorithm 1.

**Algorithm 1:** The DSD-SNN Continual Learning.

**Input:** Dataset $D_{T_i}$ for each task $T_i$;
Initialize empty network $Net$;
Constant parameters of growth $\rho\%$ and pruning $\rho_c, \rho_f$.
**Output:** Prediction Class in task $T_i$ (TIL) or in all tasks (CIL).
**for** *each sequential task* $T_i$ **do**
  Growing new neurons to $Net$ as Eq. 3;
  **for** *epoch* $= 0$; *epoch* $< E$; *epoch* $++$ **do**
    SNN forward prediction $Net\,(D_{T_i})$ as Eq. 7;
    SNN backpropagation to update new connections as Eq. 8;
    Assessing importance for newly grown neurons as Eq. 4;
    Calculating the neuronal pruning function as Eq. 5 and Eq. 6;
    Pruning redundant neurons with $P_i^l < 0$;
  **end**
  Freezing retained neurons in $Net$;
**end**

## 4 Experiments

### 4.1 Datasets and Models

To validate the effectiveness of our DSD-SNN algorithm, we conduct extensive experiments and analyses on the spatial MNIST [LeCun *et al.*, 1998], CIFAR100 [Xu *et al.*, 2015] and neuromorphic temporal N-MNIST datasets [Orchard *et al.*, 2015] based on the brain-inspired cognitive intelligence engine BrainCog [Zeng *et al.*, 2022]. The specific experimental datasets and models are as follows:

- Permuted MNIST: We permute the MNIST handwritten digit dataset to ten tasks via random permutations of the pixels. Each task contains ten classes, divided into 60,000 training samples and 10,000 test samples. As for the SNN model, we use the SNN with two convolutional layers, one fully-connected layer, and the multi-headed output layer.
- Permuted N-MNIST: We randomly permute the N-MNIST ( the neuromorphic capture of MNIST) to ten tasks. And we employ the same sample division and the same SNN structure as MNIST.
- Split CIFAR100: The more complex natural image dataset CIFAR100 is trained in several splits including 10 steps (10 new classes per step), 20 steps (5 new classes per step). SNN model consisting of eight convolutional layers, one fully connected and multi-headed output layer are used to generate the predicted class.

For the task classifier, we use networks containing a hidden layer with 100 hidden neurons for MNIST and N-MNIST, and 500 hidden neurons for CIFAR100. To recognize tasks better, we replay 2000 samples for each task as [Rebuffi *et al.*, 2017; Rajasegaran *et al.*, 2019b; Rajasegaran *et al.*, 2019a]. Our code is available at https://github.com/BrainCog-X/Brain-Cog/tree/main/examples/Structural_Development/DSD-SNN.

### 4.2 Comparisons of Performance

As shown in Fig. 3a, our DSD-SNN model maintains high accuracy with increasing number of learned tasks. This demonstrates that the proposed model overcomes catastrophic forgetting on all MNIST, neuromorphic N-MNIST and more complex CIFAR100 datasets, achieving robustness and generalization capability on both TIL and CIL. To validate the effectiveness of our dynamic structure development module, we compare the learning process of DSD-SNN with other DNNs-based continual learning and transfer them to SNN as Fig. 3b. The experimental results indicate that DSD-SNN realizes superior performance in learning and memorizing more incremental tasks, exhibiting larger memory capacity compared to the DNNs-based continual learning baselines.

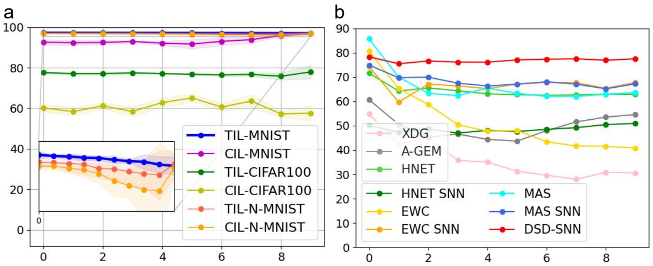


Figure 3: The average accuracy with increasing number of tasks. **(a)** Our DSD-SNN for MNIST, N-MNIST and CIFAR100. **(b)** Comparison of our DSD-SNN with other methods for CIFAR100.

The comparison results of the average accuracy with existing continual learning algorithms based on DNN and SNN are shown in Table 1 and Table 2. In the TIL scenario, our DSD-SNN achieves an accuracy of 97.30% $\pm$ 0.09% with a network parameter compression rate of 34.38% for MNIST, which outperforms most DNNs-based algorithms such as EWC [Kirkpatrick *et al.*, 2017], GEM [Lopez-Paz and Ranzato, 2017], and RCL [Xu and Zhu, 2018]. In particular, our algorithm achieves a higher performance improvement of 0.70% over the DEN [Yoon *et al.*, 2018] model (which is also based on growth and pruning). For the temporal neuromorphic N-MNIST dataset, our DSD-SNN algorithm is superior to the existing HMN algorithm which combines SNN with DNN [Zhao *et al.*, 2022c]. Meanwhile, our DSD-SNN model achieves 92.69% $\pm$ 0.57% and 96.94% $\pm$ 0.05% accuracy in CIL scenarios for MNIST and N-MNIST, respectively.

| Method | Dataset | Acc |
|---|---|---|
| EWC [Kirkpatrick *et al.*, 2017] | MNIST | 81.60% |
| GEM [Lopez-Paz and Ranzato, 2017] | MNIST | 92.00% |
| DEN [Yoon *et al.*, 2018] | MNIST | 96.60% |
| RCL [Xu and Zhu, 2018] | MNIST | 96.60% |
| CLNP [Golkar *et al.*, 2020] | MNIST | 98.42% $\pm$ 0.04 % |
| **Our DSD-SNN** | MNIST | **97.30% $\pm$ 0.09 %** |
| HMN(SNN+DNN) [Zhao *et al.*, 2022c] | N-MNIST | 78.18% |
| **Our DSD-SNN** | N-MNIST | **97.06% $\pm$ 0.09 %** |

Table 1: Accuracy of task incremental learning compared to other works for MNIST and N-MNIST datasets.

From Table 2, our DSD-SNN outperforms PathNet [Fernando *et al.*, 2017], DEN [Yoon *et al.*, 2018], RCL [Xu and Zhu, 2018] and HNET [Von Oswald *et al.*, 2020], which are also structural extension methods, in both TIL and CIL scenarios for 10 steps CIFAR100. iCaRL [Rebuffi *et al.*, 2017] and DER++ [Yan *et al.*, 2021] achieve higher accuracy of 84.20% in TIL scenarios than our 77.92%, but they are inferior in CIL scenarios (51.40% and 55.30%) than our 60.47%. Moreover, the DSD-SNN compresses the network to only 37.48% after learning all tasks, further saving energy consumption. For 20 steps CIFAR100 with more tasks, our DSD-SNN achieves even higher accuracy 81.17% in TIL scenario and has excellent experimental results consistent with 10 steps. To the best of our knowledge, this is the first time that the energy-efficient deep SNNs have been used to solve CIFAR100 continual learning and achieve comparable performance with DNNs.

In summary, the DSD-SNN model significantly outperforms the SNNs-based continual learning model on the N-MNIST dataset. On MNIST and CIFAR100 datasets, the proposed model achieves comparable performance with DNNs-based models and performs well on both TIL and CIL.

### 4.3 Effects of Efficient Continual Learning

Fig. 4 depicts the performance of the DSD-SNN model for task incremental learning on multiple datasets. The experimental results demonstrate that our SNNs-based model could improve the convergence speed and performance of new tasks during sequential continual learning, possessing the forward transfer capability. The newer tasks achieve higher performance from the beginning for MNIST and CIFAR100 datasets, indicating that the previously learned knowledge is fully utilized to help the new tasks. Also, the new tasks converge to higher performance faster, suggesting that the network has a strong memory capacity to continuously learn and remember new tasks. Similar comparable results can be obtained on the N-MNIST dataset.

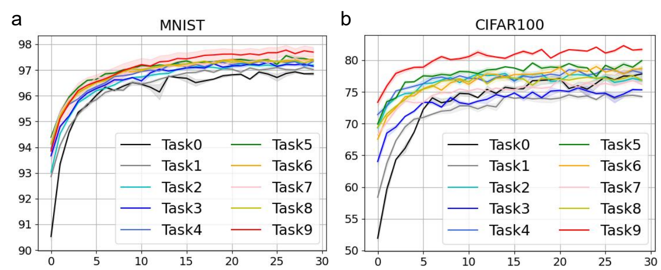


Figure 4: During the continual learning process of each task, the changes of accuracy with epochs.

### 4.4 Ablation Studies

**Effects of each component.** To verify the effectiveness of the growth and pruning components in DSD-SNN model, we compare the number of network parameters (Fig. 5a) and performance (Fig. 5b) of DSD-SNN, DSD-SNN without pruning, and DSD-SNN without reused growth during multi-task continual learning. The experimental results show that the number of parameters in the DSD-SNN model fluctuates up and finally stabilizes at 37.48% for CIFAR100, achieving superior accuracy on multi-task continual learning. In contrast,

| Method | 10steps TIL Acc (%) | 10steps CIL Acc (%) | 20steps TIL Acc (%) | 20steps CIL Acc (%) |
|---|---|---|---|---|
| EWC [Kirkpatrick *et al.*, 2017] | $61.11 \pm 1.43$ | $17.25 \pm 0.09$ | $50.04 \pm 4.26$ | $4.63 \pm 0.04$ |
| MAS [Aljundi *et al.*, 2018] | $64.77 \pm 0.78$ | $17.07 \pm 0.12$ | $60.40 \pm 1.74$ | $4.66 \pm 0.02$ |
| PathNet [Fernando *et al.*, 2017] | 53.10 | 18.50 | - | - |
| SI [Zenke *et al.*, 2017] | $64.81 \pm 1.00$ | $17.26 \pm 0.11$ | $61.10 \pm 0.82$ | $4.63 \pm 0.04$ |
| DEN [Yoon *et al.*, 2018] | 58.10 | - | - | - |
| RCL [Xu and Zhu, 2018] | 59.90 | - | - | - |
| iCaRL [Rebuffi *et al.*, 2017] | $84.20 \pm 1.04$ | $51.40 \pm 0.99$ | $85.70 \pm 0.68$ | $47.80 \pm 0.48$ |
| HNET [Von Oswald *et al.*, 2020] | $63.57 \pm 1.03$ | - | $70.48 \pm 0.25$ | - |
| DER++ [Yan *et al.*, 2021] | $84.20 \pm 0.47$ | $55.30 \pm 0.10$ | $86.60 \pm 0.50$ | $46.60 \pm 1.44$ |
| FOSTER [Wang *et al.*, 2022] | - | 72.90 | - | 70.65 |
| DyTox [Douillard *et al.*, 2022] | - | $73.66 \pm 0.02$ | - | $72.27 \pm 0.18$ |
| **Our DSD-SNN** | **$77.92 \pm 0.29$** | **$60.47 \pm 0.72$** | **$81.17 \pm 0.73$** | **$57.39 \pm 1.97$** |

Table 2: Accuracy comparisons with DNNs-based algorithms for CIFAR100.

the network scale of the model without pruning rises rapidly and quickly fills up the memory capacity, leading to a dramatic drop in performance after learning six tasks. The above results reveal that the pruning process of DSD-SNN not only reduces the computational overhead but also improves the performance and memory capacity.

For the growth module of DSD-SNN, we eliminate the addition of connections from frozen neurons to verify the effectiveness of reusing acquired knowledge in improving learning for new tasks. From Fig. 5a and b, DSD-SNN without reused growth suffers from catastrophic forgetting when there is no additional conservation of sub-network masks. The scale of the non-reused network is very small, and the failure to reuse acquired knowledge significantly degrades the performance of the model on each task. Therefore, we can conclude that reusing and sharing acquired knowledge in our DSD-SNN model achieves excellent forward transfer capability.

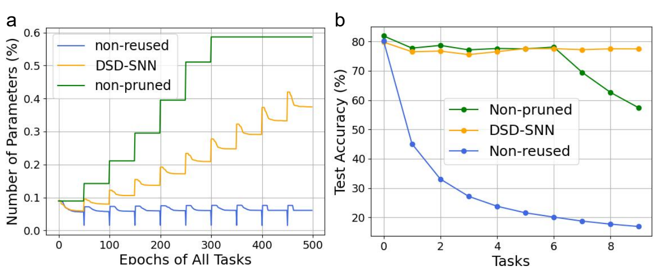


Figure 5: Number of network parameters **(a)** and accuracy **(b)** of our DSD-SNN, non-pruned and non-reused model for CIFAR100.

**Effects of different parameters.** We analyze the effects of different growing and pruning parameters (the growth scale $\rho$ and pruning intensity $\rho_c, \rho_f$). For the growth parameter $\rho$, the results are very close in the range of 5-15% for MNIST in Fig. 6a, as well as in the range of 7.5-15% for CIFAR100 in Fig. 6b. Only in the larger case, there is a performance degradation in the later learning task (8th task), due to the larger growth scale of the previous task resulting in insufficient space to learn new knowledge in the later tasks.

Fig. 6c and d describe the effects of pruning strength $\rho_c$, $\rho_f$ on performance. The larger $\rho_c, \rho_f$, the more convolutional channels and fully connected neurons are pruned. We found that the accuracy is very stable at less than $\rho_c = 0.50, \rho_f = 1.00$ for MNIST and $\rho_c = 0.75, \rho_f = 1.25$ for CIFAR100, but the accuracy declines at larger $\rho_c, \rho_f$ due to the overpruning. The DSD-SNN model is more adaptable to pruning parameters on the CIFAR100 dataset because it has a larger parameter space of SNN model. These ablation experiments demonstrate that our DSD-SNN is very robust for different growth and pruning parameters across multiple datasets.

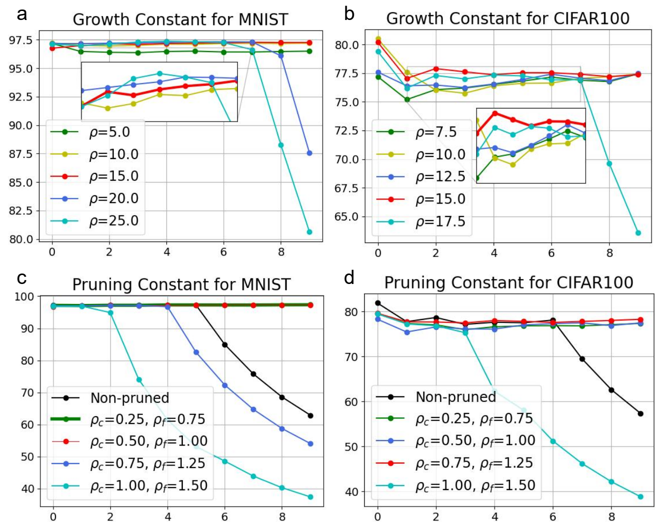


Figure 6: The effect of pruning and growth parameters on accuracy in multi-task continual learning.

## 5 Conclusion

Inspired by the brain development mechanism, we propose a DSD-SNN model based on dynamic growth and pruning to enhance efficient continual learning. Applied to both TIL and CIL scenarios based on the deep SNN, the proposed model can fully reuse the acquired knowledge to help improve the performance and learning speed of new tasks, and combine with pruning mechanism to significantly reduce the computational overhead and enhance the memory capacity. Our DSD-SNN model belongs to the very few explorations on SNNs-based continual learning. The proposed algorithm surpasses the SOTA performance achieved by SNNs-based continual learning algorithm and achieves comparable performance with DNNs-based continual learning algorithms.

## Acknowledgements

This work is supported by the National Key Research and Development Program (Grant No. 2020AAA0107800), the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB32070100), the National Natural Science Foundation of China (Grant No. 62106261).

## Contribution Statement

B.H. and F.Z are equal contribution and serve as co-first authors. B.H., F.Z. and Y.Z. designed the study. B.H., F.Z. W.P. and G.S.performed the experiments and the analyses. B.H., F.Z. and Y.Z. wrote the paper.

## References

[Abbott, 1999] Larry F Abbott. Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain research bulletin*, 50(5-6):303–304, 1999.

[Aljundi *et al.*, 2018] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154, 2018.

[Bruer, 1999] John T Bruer. Neural connections: Some you use, some you lose. *The Phi Delta Kappan*, 81(4):264–277, 1999.

[Dekhovich *et al.*, 2023] Aleksandr Dekhovich, David MJ Tax, Marcel HF Sluiter, and Miguel A Bessa. Continual prune-and-select: class-incremental learning with specialized subnetworks. *Applied Intelligence*, pages 1–16, 2023.

[Douillard *et al.*, 2022] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9285–9295, 2022.

[Elman *et al.*, 1996] Jeffrey L Elman, Elizabeth A Bates, and Mark H Johnson. *Rethinking innateness: A connectionist perspective on development*, volume 10. MIT press, 1996.

[Fernando *et al.*, 2017] Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.

[French, 1999] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

[Furber *et al.*, 1987] Susan Furber, Ronald W Oppenheim, and David 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*, 7(6):1816–1832, 1987.

[Gao *et al.*, 2022] Qiang Gao, Zhipeng Luo, Diego Klabjan, and Fengli Zhang. Efficient architecture search for continual learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[Golkar *et al.*, 2020] Siavash Golkar, Michael Kagan, and Kyunghyun Cho. Continual learning via neural pruning. In *International Conference on Learning Representations*, 2020.

[Han *et al.*, 2022a] Bing Han, Feifei Zhao, Yi Zeng, and Wenxuan Pan. Adaptive sparse structure development with pruning and regeneration for spiking neural networks. *arXiv preprint arXiv:2211.12219*, 2022.

[Han *et al.*, 2022b] Bing Han, Feifei Zhao, Yi Zeng, and Guobin Shen. Developmental plasticity-inspired adaptive pruning for deep spiking and artificial neural networks. *arXiv preprint arXiv:2211.12714*, 2022.

[Huttenlocher and others, 1979] Peter R Huttenlocher et al. Synaptic density in human frontal cortex-developmental changes and effects of aging. *Brain Res*, 163(2):195–205, 1979.

[Huttenlocher, 1990] Peter R Huttenlocher. Morphometric study of human cerebral cortex development. *Neuropsychologia*, 28(6):517–527, 1990.

[Jun and Jin, 2007] Joseph K Jun and Dezhe Z Jin. Development of neural circuitry for precise temporal sequences through spontaneous activity, axon remodeling, and synaptic plasticity. *PloS one*, 2(8):e723, 2007.

[Kemker and Kanan, 2018] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. In *International Conference on Learning Representations*, 2018.

[Kirkpatrick *et al.*, 2017] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Li and Hoiem, 2017] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[Lopez-Paz and Ranzato, 2017] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

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

[Orchard *et al.*, 2015] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:437, 2015.

[Qin *et al.*, 2020] Haotong Qin, Ruihao Gong, Xianglong Liu, Mingzhu Shen, Ziran Wei, Fengwei Yu, and Jingkuan Song. Forward and backward information retention for accurate binary neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2250–2259, 2020.

[Rajasegaran *et al.*, 2019a] Jathushan Rajasegaran, Munawar Hayat, Salman Khan, Fahad Shahbaz Khan, and Ling Shao. Random path selection for incremental learning. *Advances in Neural Information Processing Systems*, 3, 2019.

[Rajasegaran *et al.*, 2019b] Jathushan Rajasegaran, Munawar Hayat, Salman H Khan, Fahad Shahbaz Khan, and Ling Shao. Random path selection for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[Rebuffi *et al.*, 2017] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[Rusu *et al.*, 2016] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. In *In Proceedings of Conference on Neural Information Processing Systems*, 2016.

[Siddiqui and Park, 2021] Zahid Ali Siddiqui and Unsang Park. Progressive convolutional neural network for incremental learning. *Electronics*, 10(16):1879, 2021.

[Silva *et al.*, 2009] Alcino J Silva, Yu Zhou, Thomas Rogerson, Justin Shobe, and J Balaji. Molecular and cellular approaches to memory allocation in neural circuits. *Science*, 326(5951):391–395, 2009.

[van de Ven *et al.*, 2020] Gido M van de Ven, Hava T Siegelmann, and Andreas S Tolias. Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, 11(1):1–14, 2020.

[Von Oswald *et al.*, 2020] Johannes Von Oswald, Christian Henning, Benjamin F Grewe, and João Sacramento. Continual learning with hypernetworks. In *International Conference on Learning Representations*, 2020.

[Wang *et al.*, 2022] Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXV*, pages 398–414. Springer, 2022.

[Wu *et al.*, 2018] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018.

[Xu and Zhu, 2018] Ju Xu and Zhanxing Zhu. Reinforced continual learning. *Advances in Neural Information Processing Systems*, 31, 2018.

[Xu *et al.*, 2015] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[Yan *et al.*, 2021] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2021.

[Yoon *et al.*, 2018] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018.

[Zeng *et al.*, 2022] Yi Zeng, Dongcheng Zhao, Feifei Zhao, Guobin Shen, Yiting Dong, Enmeng Lu, Qian Zhang, Yinqian Sun, Qian Liang, Yuxuan 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.

[Zenke *et al.*, 2017] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995. PMLR, 2017.

[Zhao and Zeng, 2021] Feifei Zhao and Yi Zeng. Dynamically optimizing network structure based on synaptic pruning in the brain. *Frontiers in Systems Neuroscience*, 15:620558, 2021.

[Zhao *et al.*, 2022a] Feifei Zhao, Yi Zeng, and Jun Bai. Toward a brain-inspired developmental neural network based on dendritic spine dynamics. *Neural Computation*, 34(1):172–189, 2022.

[Zhao *et al.*, 2022b] Feifei Zhao, Yi Zeng, Bing Han, Hongjian Fang, and Zhuoya Zhao. Nature-inspired self-organizing collision avoidance for drone swarm based on reward-modulated spiking neural network. *Patterns*, 3(11):100611, 2022.

[Zhao *et al.*, 2022c] Rong Zhao, Zheyu Yang, Hao Zheng, Yujie Wu, Faqiang Liu, Zhenzhi Wu, Lukai Li, Feng Chen, Seng Song, Jun Zhu, et al. A framework for the general design and computation of hybrid neural networks. *Nature communications*, 13(1):1–12, 2022.