Supplementary material for CogniSNN: An Exploration to Random Graph Architecture based Spiking Neural Networks with Enhanced Depth-Scalability and Path-Plasticity

Yongsheng Huang[1,2], **Peibo Duan**[1,*], **Zhipeng Liu**[1], **Kai Sun**[3], **Changsheng Zhang**[1], **Bin Zhang**[1] and **Mingkun Xu**[2,*]

[1]School of Software, Northeastern University, Shenyang, China
[2]Guangdong Institute of Intelligence Science and Technology, Zhuhai, China
[3]Department of Data Science & AI, Monash University, Melbourne, Australia
{2371447, 2310543}@stu.neu.edu.cn, duanpeibo@swc.neu.edu.cn, {zhangbin, zhangchangsheng}@mail.neu.edu.cn, kai.sun1@monash.edu, xumingkun@gdiist.cn

1 Details of Experiments

1.1 Dataset

We evaluated the proposed method on three widely-used neuromorphic benchmark datasets: DVS-Gesture, CIFAR10-DVS, and N-Caltech101.

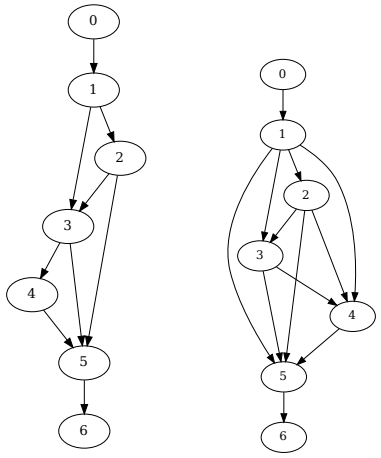
DVS-Gesture: This dataset focuses on gesture recognition, capturing 11 gesture categories from different individuals. It includes 1,464 samples (1176 for training and 288 for testing) with a spatial dimension of 128×128. We processed the dataset directly using the method provided by SpikingJelly, without applying any additional preprocessing or data augmentation.

CIFAR10-DVS: This dataset is a spiking version of the static CIFAR10 dataset, with the same 10 categories and 1,000 event stream records per category. Similar to DVS-Gesture, the spatial dimension of each sample is 128×128. And we utilized the provided package in SpikingJelly to process the dataset without additional preprocessing, splitting it into training and testing sets in a 9:1 ratio.

N-Caltech101: This dataset is a spiking version of the Caltech101 dataset, with 101 categories, containing about 40 to 800 images per category. It contains 8709 samples, each with a spatial dimension of 180×240. We downsampled images to 128×128 using bilinear interpolation and split it into training and testing sets in a 9:1 ratio. For training set, we applied the random cropping and horizontal flipping to enhance data diversity and model robustness, following common practices in SNNs.

1.2 Implementation Details

We conducted experiments using the PyTorch and SpikingJelly, training all models on an NVIDIA RTX 4090 GPU with a batch size of 8. The Stochastic Gradient Descent (SGD) optimizer was utilized with a momentum of 0.9 and no weight decay. For the Leaky Integrate-and-Fire (LIF) neurons, the membrane potential time constant was set to $\tau = 2.0$ and the firing threshold was set to $v_{thr} = 1$. Moreover, to accelerate the training process, we utilized Automatic Mixed Precision (AMP) scaling.



(a) ER-driven RGA. (b) WS-driven RGA.

Figure 1: Visualization of RGA with 7 nodes.

For the CIFAR10-DVS and N-Caltech101 datasets, the models were trained for 192 epochs with an initial learning rate of 0.001, decaying by a factor of 10 every 64 epochs.

For the DVSGesture dataset, we used a complex Cosine Annealing Learning Rate Scheduler with an initial learning rate of 0.001 and a period of $T_{max} = 64$, training for 300 epochs.

2 Random Graph Models

The Erdős–Rényi (ER) graph is a type of random graph model characterized by two parameters, typically denoted as ER(N, P), where N represents the number of nodes and P represents the probability of an edge is connected between two nodes.

The Watts-Strogatz (WS) graph is a small-world network model that effectively reflects the clustering and short-path characteristics in real-world networks. Generally, a WS graph is generated using three parameters: N, K, and P, denoted as

WS(N, K, P). Specifically, the WS graph starts by creating a ring network with N nodes, where each node is connected to its K (K is an even number) nearest neighbor nodes. Then, with the condition of avoiding duplicate edges and self-loops, each existing edge is randomly reconnected with a probability P.

For neuromorphic dataset classification tasks, we utilized two architectures: ER(5, 0.75) and WS(5, 4, 0.75). In continual learning experiments, to better explore the impact of the critical path, we employed WS(30, 4, 0.75) and ER(30, 0.2). It is important to note that the random graphs generated by the random graph models are not necessarily directly usable as network model graphs. Therefore, for each graph, we introduced a start node that connects to all nodes with in-degree 0, and an end node that connects to all nodes with out-degree 0. Consequently, the final number of nodes for the four graphs is 7, 7, 32, and 32. These graph structures are shown in Figures 1 and 2.

3 Betweenness Centrality of Node and Edge

For a node v, its betweenness centrality $C_{B_n}(v)$ [Freeman, 1977] is defined as:

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\phi_{st}(v)}{\phi_{st}}, \tag{1}$$

where s and t denote the start and end nodes, ϕ_{st} is the total number of shortest paths from s to t, and $\phi_{st}(v)$ is number of the shortest paths from s to t passing through the node v.

Similarly, for an edge e, its betweenness centrality $C_B(e)$ [Girvan and Newman, 2002] is defined as:

$$C_B(e) = \sum_{s \neq t \in V} \frac{\phi_{st}(e)}{\phi_{st}}, \tag{2}$$

where $\phi_{st}(v)$ is the number of the shortest paths from s to t that pass through the edge e.

4 Visualization of the Activation Frequency of ResNodes

Firstly, we made a visualization of the average activation rate of each node in the CogniSNN processing of the CIFAR10 dataset, which is the inspiration point for us to find the critical path using betweenness centrality. Specifically, the numbers in each node represent the average activation rate of the node. As can be seen in Figure 9, among the network, the centre nodes tend to have higher activation rates, while the edge nodes always have smaller activation rates, with a distribution similar to the betweenness centrality of graph theory.

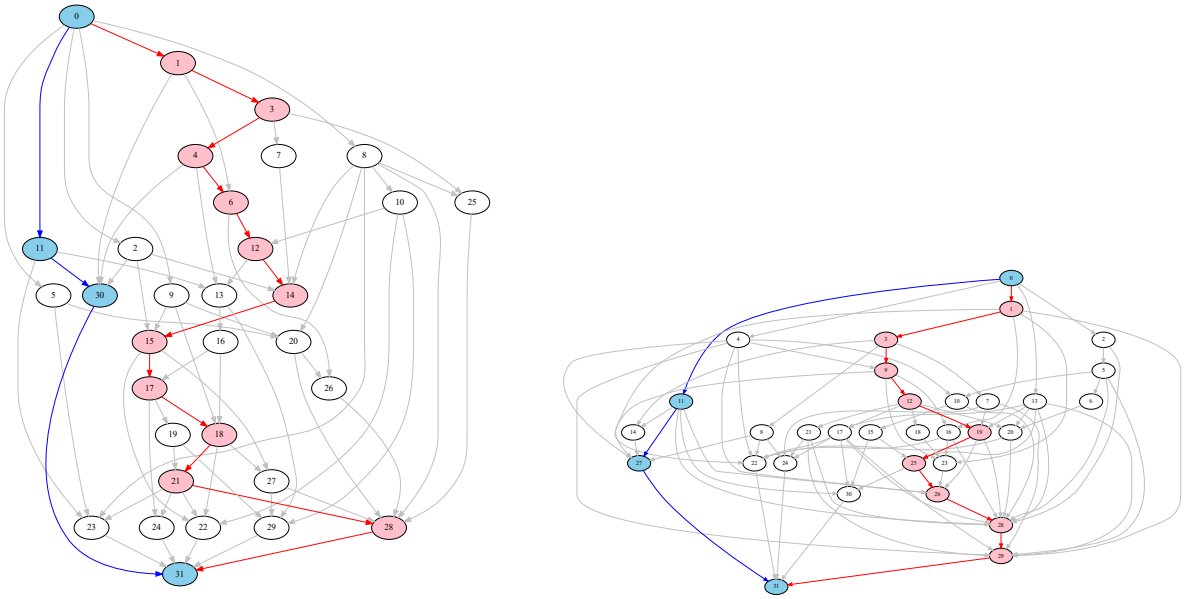
5 Accuracy Change Curves

In this section, we report the training procedure to validate the performance of CogniSNN on three neuromorphic datasets with different timesteps and different random graph architectures(ER and WS). For all experiments, we use the same random seed to ensure the reproducibility of the experiments. The complete training process is shown in Figures 3 to 8.

References

[Freeman, 1977] LC Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 1977.

[Girvan and Newman, 2002] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.



(a) WS-driven RGA. (b) ER-driven RGA.

Figure 2: Visualization of RGA with 32 nodes and its critical paths.

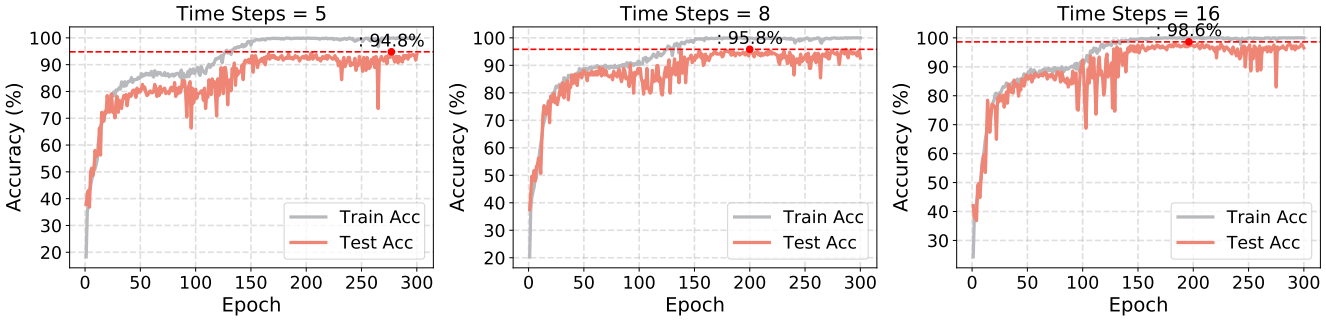


Figure 3: ER-RGA-7 based CogniSNN training process on DVSGesture under different time steps.

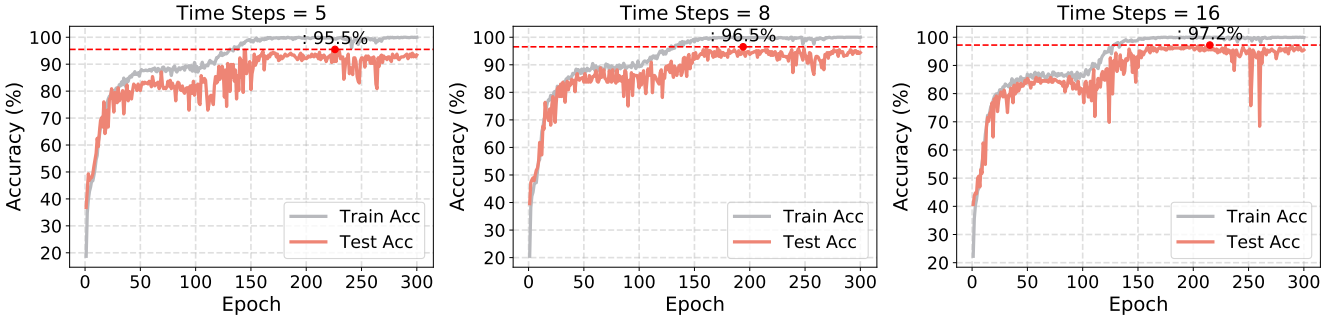


Figure 4: WS-RGA-7 based CogniSNN training process on DVSGesture under different time steps.

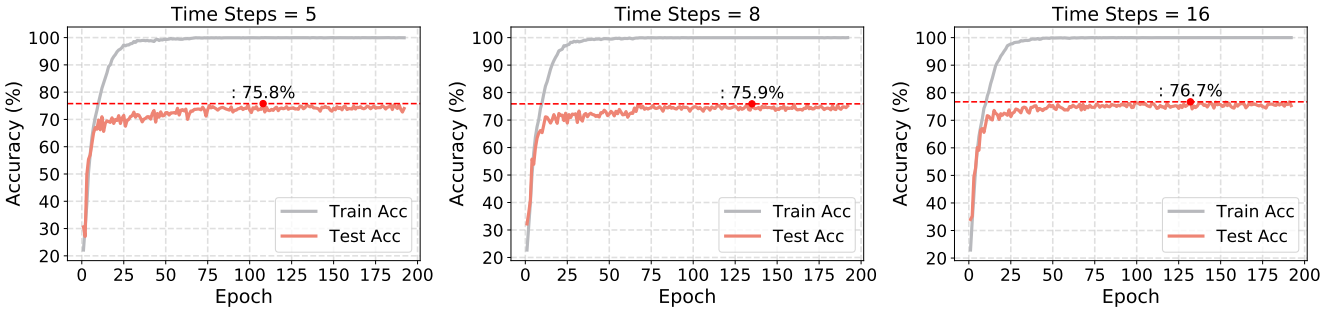


Figure 5: ER-RGA-7 based CogniSNN training process on CIFAR10-DVS under different time steps.

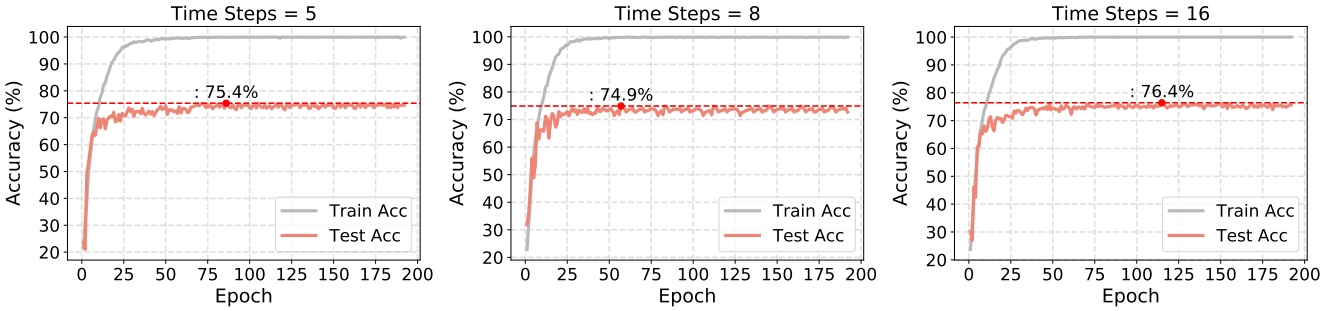


Figure 6: WS-RGA-7 based CogniSNN training process on CIFAR10-DVS under different time steps.

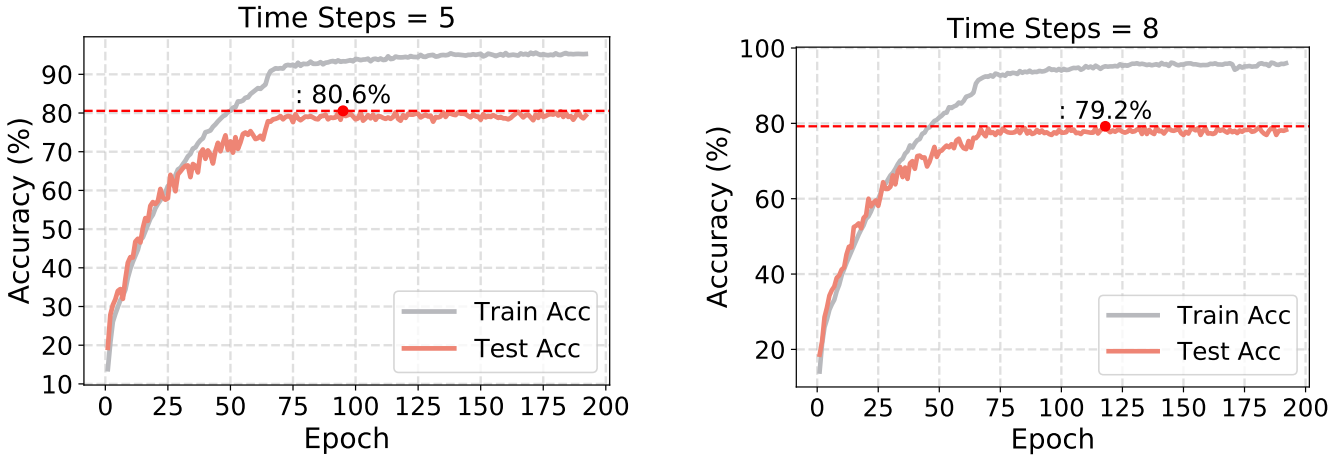


Figure 7: ER-RGA-7 based CogniSNN training process on NCaltech101 under different time steps.

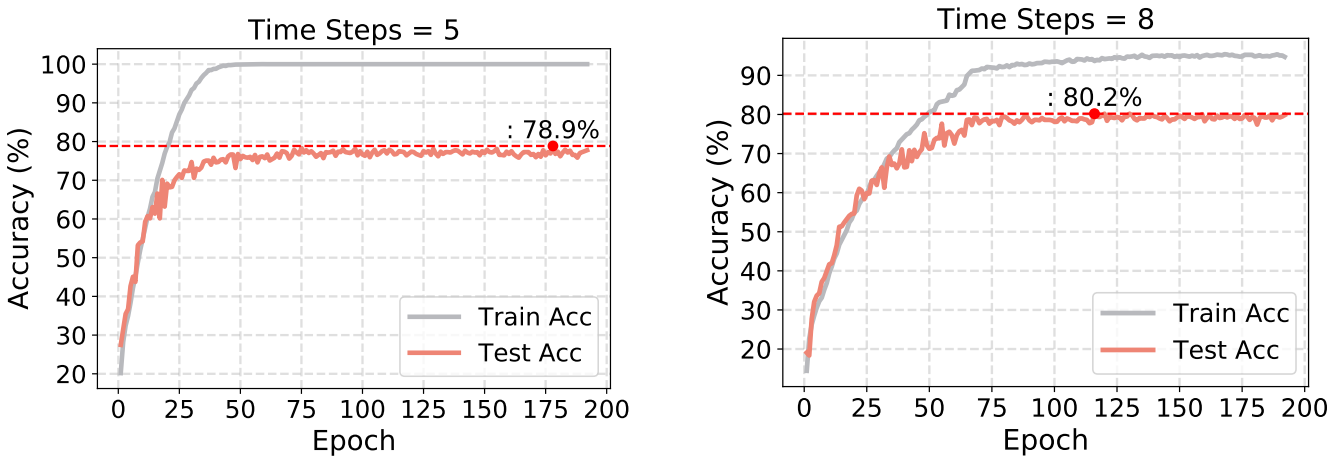


Figure 8: WS-RGA-7 based CogniSNN training process on NCaltech101 under different time steps.

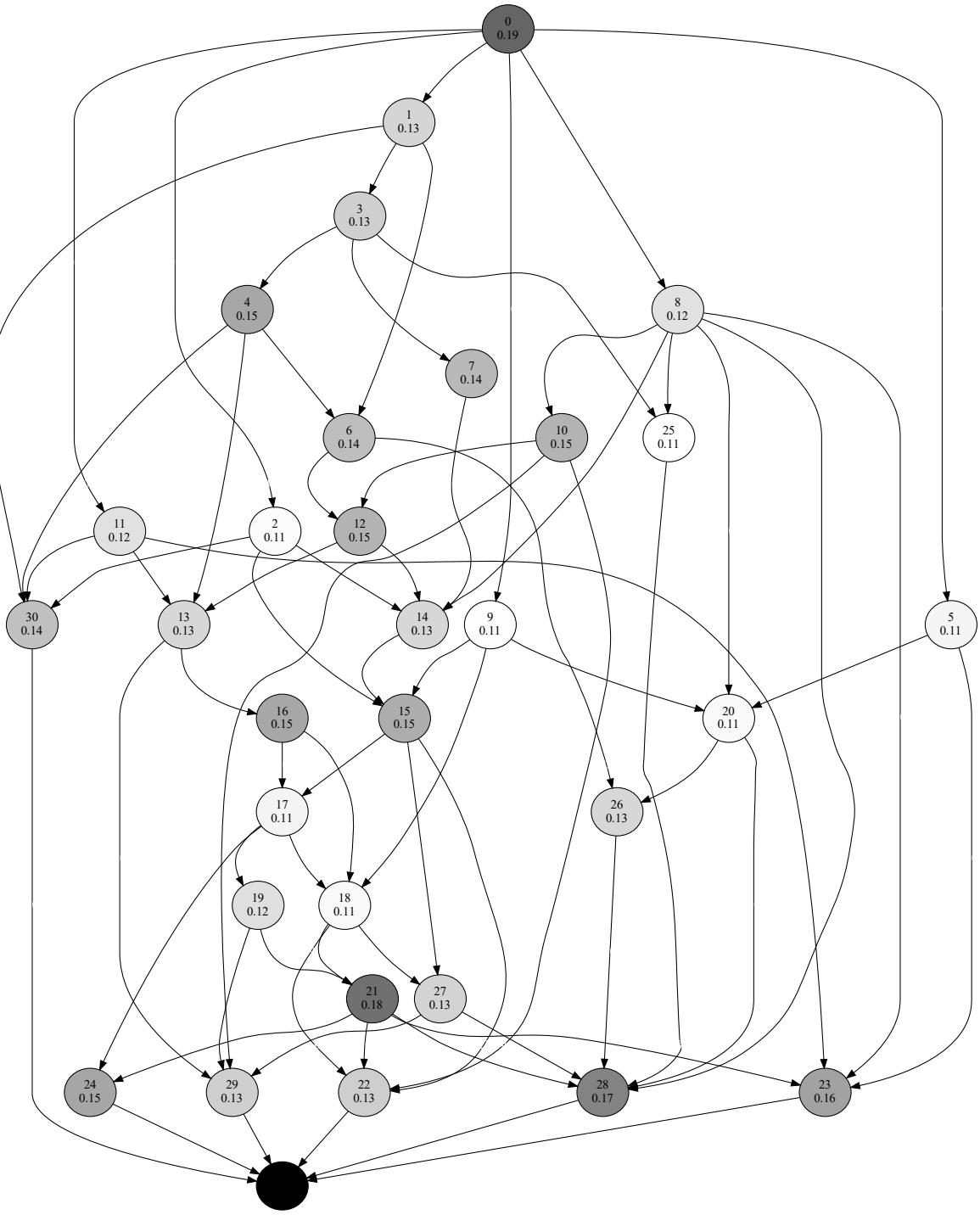


Figure 9: Visualizing the Activation Frequency of ResNodes