Reconstructing Spiking Neural Networks Using a Single Neuron with Autapses

Supplementary Material

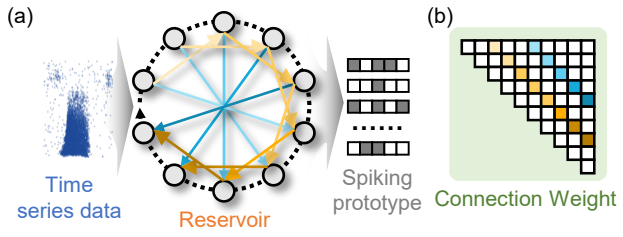

Figure 9. RC architecture of the TDA-SNN model. (a) Overall RC architecture constructed from the TDA-SNN model. (b) Autaptic weight matrix corresponding to the RC framework.

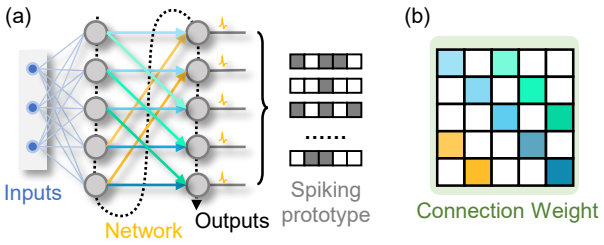

Figure 10. MLP architecture of the TDA-SNN model. (a) Overall MLP architecture constructed from the TDA-SNN model. (b) Autaptic weight matrix corresponding to the MLP framework.

7. Weight Matrix of Autapses

7.1. Reservoir Computing Model with TDA-LIF

Based on the equivalent formulation introduced in Sec. 3.2, external inputs are injected into each internal temporal node, and the spike activities of all nodes are recorded over the external data time window T [7, 8]. The resulting firing patterns are then compared for recognition, as illustrated in Fig. 9(a). The autaptic weight matrix W_a characterizes the temporal feedback structure among the unfolded temporal nodes, where each element specifies an autaptic connection from one node to another. Owing to the directed nature of temporal feedback, W_a takes an upper-triangular form, with each parallel diagonal corresponding to a specific autaptic delay d. As shown in Fig. 9(b), two such diagonals appear, representing delays of $d = 3$ (yellow) and $d = 6$ (blue). The resulting unfolded structure, derived from and functionally equivalent to the TDA-LIF neuron, forms the complete reservoir-computing (RC) architecture.

7.2. Multilayer Perceptron Model with TDA-LIF

Similarly, an external layer is added before the first TDA-SNN layer to form a multilayer perceptron (MLP), and the spike activity of the final layer is recorded over the external data time window T [9]. The resulting spike patterns are then compared for recognition, as shown in Fig. 10(a). The autaptic connection matrix W_a describes the temporal feedback structure across temporal nodes: each element specifies an autaptic connection from a node in the previous layer to one in the next layer, and each parallel diagonal corresponds to a particular autaptic delay. As illustrated in Fig. 10(b), three diagonals appear, representing delays of $d = 2$ (yellow), $d = 5$ (blue), and $d = 7$ (green). This delayed-feedback structure unfolds directly from the TDA-LIF neuron and is functionally equivalent to it, thereby forming the complete spiking MLP architecture.

8. Spatiotemporal Prototype Learning

To enable class-specific spatiotemporal representations within TDA-SNN, we construct a set of binary prototypes corresponding to the desired spiking patterns of each class. These prototypes serve as reference patterns against which the encoded network outputs are compared, facilitating classification and similarity measurement in the spatiotemporal domain.

Specifically, a set of binary prototypes $\mathbf{K} \in \{0,1\}^{C\times T}$ is defined to represent target spiking patterns for C classes over T timesteps. Let $f(x;\theta)$ denote the encoded spatiotemporal representation of input x. To measure the similarity between the encoded representation and each class prototype, we compute

$$d_i(x) = -\|f(x;\theta) - \mathbf{k}_i\|_2^2, \tag{8}$$

where $d_i(x)$ denotes the similarity score between $f(x;\theta)$ and the i-th prototype $\mathbf{k}_i \in \mathbf{K}$.

During inference, each input sample is assigned to the class of the closest prototype in the spatiotemporal feature space. The prototype-based classification procedure follows the method described in [2, 10, 11].

9. Datasets and Experimental Setup

We evaluate TDA-SNN across multiple benchmarks under different structural mappings. DEAP [5] and SHD [3] are used to evaluate the reservoir-computing setting; MNIST [4], Fashion-MNIST (fMNIST) [12], and DVS Gesture [1] are used to validate the MLP setting; and DVS Gesture together with CIFAR10 [6] are further used to study the convolution-like setting.

All models are implemented in PyTorch and trained on an NVIDIA A100 GPU using the Adam optimizer with

Table 3. The hyperparameters for different datasets.

Dataset	bs	lr	T	dt	v_{th}	τ
DEAP	200	0.002	6	100	1.0	0.5
SHD	512	0.01	100	14	0.3	0.3
MNIST	512	0.002	4	-	0.3	0.5
fMNIST	512	0.002	4	-	0.3	0.5
DVS Gesture	128	0.002	8	125	1.1	0.67
CIFAR10	200	0.002	4	-	1.0	0.5
CIFAR100	512	0.002	4	-	1.0	0.5

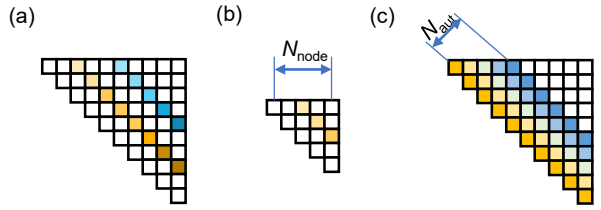

Figure 11. Evolution of connection weights in the RC architecture. (a) Random delay strategy. (b) Low-node configuration. (c) Maximum connection strategy.

a cosine learning-rate decay schedule. Unless otherwise stated, training is conducted for 100 epochs. For RC and MLP experiments, results are averaged over 10 independent runs and reported as mean $\pm$ standard deviation; due to the higher computational cost of convolutional experiments, those results are averaged over 5 runs. To ensure a fair comparison, TDA-SNN and the corresponding standard SNN baselines use aligned training protocols and matched architectural scales for each setting.

The complete set of hyperparameters is summarized in Table 3, including batch size (bs), learning rate (lr), time window T, slice width dt, membrane threshold v_{th}, and membrane decay factor τ.

In our experiments, the standard SNN evolves over an external time window, whereas TDA-SNN introduces an additional internal evolution time. Consequently, the membrane potential dynamics in TDA-SNN are decoupled from those of the standard SNN: the latter progresses along external timesteps, while the former evolves over internal temporal nodes. In TDA-SNN, the external time window is treated as an independent dimension and does not directly participate in the neuron dynamics.

10. Weight Matrices Under RC and MLP Architectures

In the RC architecture, the TDA-SNN connection matrix is an upper-triangular square matrix. As shown in Fig. 11, the autaptic selection strategy and the number of nodes jointly determine its structure. The random delay (RD) strategy produces a sparser matrix with generally longer-delay connections. Increasing the number of nodes expands the space of possible connections, whereas using fewer nodes naturally reduces the total number of available connections. In contrast, selecting autapses along the main diagonal toward the upper-right corner under the maximum-connections (MC) strategy yields the densest connectivity while simultaneously minimizing autaptic latency.

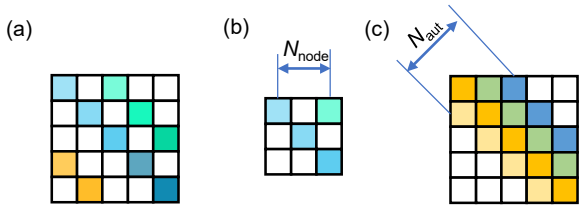

Figure 12. Evolution of connection weights in the MLP architecture. (a) Random delay strategy. (b) Low-node configuration. (c) Maximum connection strategy.

For the MLP architecture, the TDA-SNN connection matrix width and height correspond to the number of input and output nodes, respectively. Fig. 12 illustrates how autapse selection strategies shape the matrix. Using the MC strategy, autapses are selected along the diagonal and distributed to neighboring nodes, maximizing connectivity while minimizing latency. In contrast, the RD strategy yields sparser, longer-delayed connections. This demonstrates that, in feedforward structures, structured delay assignment enables efficient temporal propagation across layers, analogous to the role of weight connectivity in conventional MLPs.

11. Significance Analysis of Experimental Results

The statistical significance results are summarized in Table 4. Overall, the effect of autaptic selection strategy depends on both the dataset and the delay budget. On DEAP, most comparisons are not statistically significant, suggesting that the task is relatively insensitive to the specific delay-selection strategy. On SHD, significant differences appear mainly at intermediate delays, indicating that strategy choice matters in specific temporal regimes. On MNIST, many pairwise comparisons are significant, especially for MC-RD and for RD-T Inv. at moderate-to-large delays, showing that strategy selection can substantially affect performance. fMNIST and DVS Gesture also exhibit several significant differences, although the pattern varies with both the comparison pair and the delay value rather than following a uniform trend. These observations are consistent with the ablation results in showing that the impact of delay assignment is dataset-dependent and becomes particularly evident in specific delay regimes.

Table 4. Paired t-tests of Autaptic Selection Strategies Across Different Delays. * denotes $p < 0.05$.

Dataset	Delay	1	2	4	8	16	32	64
DEAP	MC-RD	0.79	0.66	0.65	0.73	0.51	0.05	1.0
SHD	MC-RD	0.53	*	*	*	0.40	0.81	1.0
MNIST	MC-RD	*	*	*	*	*	*	*
	MC-T Inv.	*	0.06	*	0.65	0.97	0.95	*
	RD-T Inv.	0.35	0.88	*	*	*	*	*
fMNIST	MC-RD	*	*	0.59	*	*	*	*
	MC-T Inv.	*	*	*	0.54	0.60	0.53	*
	RD-T Inv.	0.53	0.68	0.06	*	*	*	*
DVS Gesture	MC-RD	*	0.58	0.30	0.06	*	*	*
	MC-T Inv.	*	0.26	0.08	0.17	0.92	0.48	*
	RD-T Inv.	0.26	0.57	0.44	0.72	*	*	*

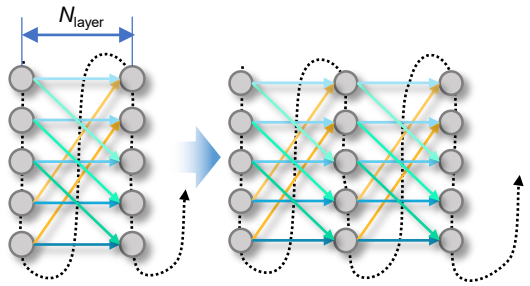

Figure 13. Extending TDA-SNN to multi-layer deep models under the MLP architecture.

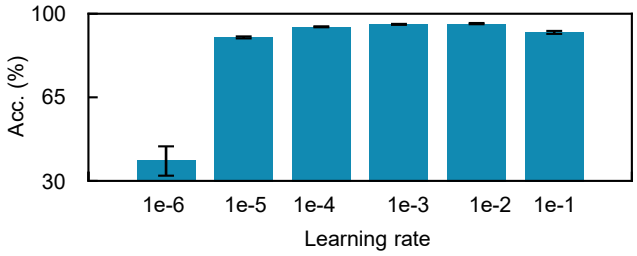

Figure 14. Learning rate sensitivity study for the MLP architecture on MNIST.

12. Impact of Layer Number in MLP Structures

TDA-SNN can be extended to deep architectures by increasing the number of internal nodes, allowing a single TDA-LIF neuron to emulate multiple layers through its temporal unfolding. As shown in Fig. 13, connections with the same color denote autapses sharing the same delay value, which remain consistent across layers. This preserves the structured temporal feedback while enabling the construction of deeper computational models.

13. Additional Ablation Studies

We additionally provide two supplementary ablation studies: learning-rate sensitivity on MNIST and an ablation with different numbers of nodes on CIFAR100. These experiments further examine the robustness of TDA-SNN under different optimization and architectural settings.

13.1. MNIST

For MNIST, the learning-rate ablation is conducted using the hyperparameter setting in Table 3, varying the initial learning rate from 10^{-6} to 10^{-1} while keeping the remaining settings unchanged. The corresponding mean accuracies with standard deviations are visualized in Fig. 14. Although the best performance is achieved at a learning rate of 10^{-2} (95.86±0.26%), we use 10^{-3} (95.55±0.20%) as the default setting because it remains close to the optimum while providing a more stable optimization process and stronger convergence robustness. This choice is particularly important for SNNs, whose discrete spikes and highly nonlinear dynamics make training sensitive to overly large learning rates, which can lead to oscillation and instability.

13.2. CIFAR100

For CIFAR100, we provide an additional ablation with different numbers of nodes under a more challenging visual classification setting. Following the setting of Section 4.3.1, we use a two-layer network, and the hyperparameter configuration is given in Table 3. The number of nodes is varied over $\{16, 32, 64, 128, 256, 512\}$ while the remaining archi-

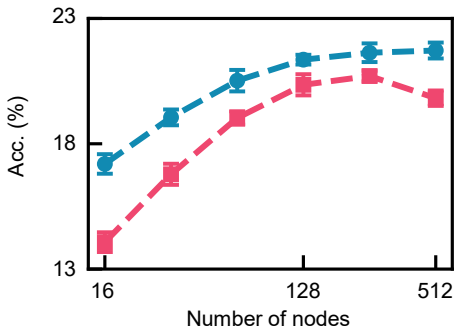

Figure 15. Number of nodes ablation on CIFAR100 for STD-SNN and TDA-SNN.

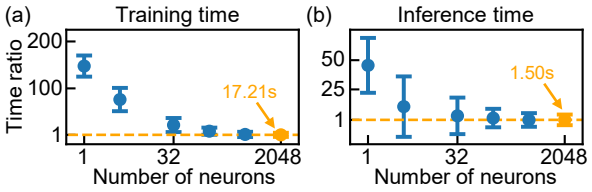

Figure 16. Training (a) and inference (b) time ratios of TDA-SNN relative to STD-SNN on CIFAR10. The yellow marker denotes STD-SNN (ratio = 1.0). The x-axis is logarithmic.

tectural and training settings are kept fixed. As shown in Fig. 15, STD-SNN improves steadily from 17.20±0.39% at 16 nodes to 21.72±0.32% at 512 nodes. TDA-SNN also improves substantially, from 14.08±0.39% at 16 nodes to a best accuracy of 20.70±0.21% at 256 nodes. When the number of nodes is further increased to 512, the accuracy decreases slightly to 19.82±0.31%, suggesting that simply increasing model size does not always yield additional gains on this challenging dataset. Nevertheless, TDA-SNN consistently improves as the number of nodes increases over a broad range and retains strong performance at medium-to-large scales, demonstrating that the proposed model remains effective even in a more difficult setting.

14. Implementation Details for Convolution-like Settings

The network adopted a standard two-layer convolutional backbone, consisting of a convolutional layer (8 output channels, 7×7 kernel size, stride 1, padding 3), followed by 4×4 max-pooling and flattening, and a fully connected layer with an output feature size of 512 for classification. The TDA-SNN variant replaced standard synapses in convolutional layers with time-delayed autapses configured using all available autaptic delays.

15. Parallelization and Scalability in Convolution-like Settings

To further analyze the temporal overhead of TDA-SNN, we study how parallelization changes the runtime characteristics in convolution-like settings. The extreme latency reported is specific to the single-neuron setting, which is designed to probe the limit of spatial compression. As shown in Fig. 16, increasing the number of parallel neurons reduces both training and inference time. On CIFAR10, relative to STD-SNN, the training-time overhead decreases from 178× to 46×, and with 512 neurons the inference time reaches 0.92× of STD-SNN. These results further support that TDA-SNN offers a flexible space–time trade-off rather than a fixed high-latency operating point.

We additionally evaluate a multi-neuron extension to examine scalability when the compression constraint is relaxed. Using 512 neurons for DVS Gesture and 256 neurons for CIFAR10 improves accuracy from 57.78% to 62.43% on DVS Gesture and from 37.64% to 40.42% on CIFAR10. Although a gap to STD-SNN remains, this upward trend confirms that the proposed TDA mechanism scales effectively when moving beyond the extreme single-neuron regime.

References

[1] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, et al. A low power, fully event-based gesture recognition system. In *CVPR*, pages 7243–7252, 2017. 5, 1

[2] Wuque Cai, Hongze Sun, Qianqian Liao, Jiayi He, Duo Chen, Dezhong Yao, and Daqing Guo. Robust Spatiotemporal Prototype Learning for Spiking Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2025. 4, 1

[3] Benjamin Cramer, Yannik Stradmann, Johannes Schemmel, and Friedemann Zenke. The heidelberg spiking data sets for the systematic evaluation of spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33 (7):2744–2757, 2020. 4, 1

[4] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012. 4, 1

[5] Sander Koelstra, Christian Muhl, Mohammad Soleymani, Jong-Seok Lee, Ashkan Yazdani, Touradj Ebrahimi, Thierry Pun, Anton Nijholt, and Ioannis Patras. DEAP: A database for emotion analysis; using physiological signals. *IEEE transactions on affective computing*, 3(1):18–31, 2011. 4, 1

[6] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 5, 1

[7] Hao Peng, Pei Chen, Na Yang, Kazuyuki Aihara, Rui Liu, and Luonan Chen. One-core neuron deep learning for time series prediction. *National Science Review*, 12(2):nwae441, 2025. 1, 2

[8] Florian Stelzer and Serhiy Yanchuk. Emulating complex networks with a single delay differential equation. *The European Physical Journal Special Topics*, 230(14):2865–2874, 2021. 1

[9] Florian Stelzer, André Röhm, Raul Vicente, Ingo Fischer, and Serhiy Yanchuk. Deep neural networks using a single neuron: folded-in-time architecture using feedback-modulated delay loops. *Nature communications*, 12(1):5164, 2021. 1, 2

[10] 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. 1, 4

[11] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. Direct training for spiking neural networks: Faster, larger, better. In *AAAI*, pages 1311–1318, 2019. 1, 4

[12] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 4, 1