

3DSMT: A Hybrid Spiking Mamba-Transformer for Point Cloud Analysis (Appendix)

Anonymous submission

Spiking Neuron Model

Spiking neuron is the fundamental unit of SNN, we choose Integrate-and-Fire (IF) (Bulsara et al. 1996) model as the spiking neuron in the proposed 3DSMT. The dynamics of a IF neuron can be formulated as follows:

$$u_t = V_{t-1} + I_t \quad (1)$$

$$S_t = \Theta(u_t - V_{th}) \quad (2)$$

$$V_t = u_t(1 - S_t) + V_{reset}S_t \quad (3)$$

Where V_{t-1} represents the membrane potential of the neuron at time step $t - 1$, which undergoes a spike trigger evaluation. I_t represents the input current at time step t . u_t denotes the membrane potential of the neuron after incorporating neuron dynamics at time step t . In Equation (2), $\Theta(\cdot)$ symbolizes the spike trigger evaluation at time step t . When the membrane potential u_t exceeds the firing threshold V_{th} , the neuron emits a spike, setting $S_t = 1$; otherwise, $S_t = 0$. In Equation (3), V_t represents the membrane potential of the neuron at time step t after the spike evaluation. If no spike is produced, V_t is equivalent to u_t ; otherwise, it is reset to the potential V_{reset} .

Dataset

ModelNet40 (Wu et al. 2015) is a widely used benchmark dataset in point cloud processing, specifically designed for 3D object classification tasks. It contains 12,311 manually synthesized CAD models from 40 common furniture and daily item categories (e.g., tables, chairs, airplanes, cars, etc.). These models are generated by uniformly sampling point cloud surfaces, with each point cloud typically containing 10,000 points. ModelNet40 serves as a standard testbed for evaluating the generalization ability of point cloud classification algorithms, thanks to its clear object structures, high-quality point clouds, and moderate scale.

ScanObjectNN (Uy et al. 2019) aims to bridge the gap between synthetic datasets and real-world applications by providing 3D point cloud data directly scanned from real scenes. It includes approximately 15,000 object instances across 15 categories, with three key variants offering different challenge levels: OBJ_BG (Object with Background) contains target object point clouds cropped from scenes while retaining original background points to force models to learn recognition amid noise; OBJ_ONLY (Object

Only) provides ‘pure’ object point clouds with manually removed backgrounds, focusing on geometric identification; PB_T50_RS (Partially Bounded with Trash objects, 50% rotation, and Sensor noise)—the most challenging variant—includes point clouds with severe occlusion, background noise, irrelevant trash objects, 50% random sample rotation, and simulated sensor noise, evaluating model robustness in complex real-world messy scenarios.

ShapeNetPart (Yi et al. 2016) is a subset of the ShapeNetCore (Chang et al. 2015) dataset, focusing on the task of 3D object part segmentation. It is based on the ShapeNetCore dataset, with part annotations made on its models. It covers 16 object categories and contains 16,881 3D models. In addition to geometric data, the dataset also includes detailed part annotation information in the .json format. ShapeNetPart plays an important role in research on 3D object part segmentation algorithms, providing strong support for training neural networks for accurate part segmentation of objects.

More Experiment Results

This section will provide a detailed introduction to the specific experimental setup and some supplementary experimental results.

Experiment Configuration.

Experiments are conducted on a server equipped with an AMD EPYC 7571 2.1 GHz 32-core processor and an NVIDIA RTX 3090 GPU with 24GB video memory. Our implementation is based on PyTorch and SpikingJelly (Fang et al. 2023).

Classification on ModelNet40. We use $N = 1024$ points as input and apply scaling and translation for data augmentation. The model is trained using the AdamW optimizer with cosine learning rate decay, an initial learning rate of 0.0005, a weight decay of 0.05, a batch size of 24, and 300 training epochs.

Classification on ScanObjectNN. We employ rotation as data augmentation with a point cloud size of $N = 2048$. During training, we utilize the AdamW optimizer with cosine decay, setting the initial learning rate to 0.0005, weight decay to 0.05, batch size to 24, and train for 300 epochs.

Few-shot Classification on ModelNet40. We adopt the ModelNet40 dataset and conduct experiments under the n -

way m -shot setting, where n denotes the number of randomly sampled classes and m represents the number of samples per class. The model is trained using only $n \times m$ samples. During testing, 20 novel samples per class are randomly selected as test data. We evaluate combinations of $n \in \{5, 10\}$ and $m \in \{10, 20\}$, reporting the mean accuracy and standard deviation over 10 independent runs. The AdamW optimizer with cosine decay is used with an initial learning rate of 0.0005, weight decay of 0.05, batch size of 24, and 150 training epochs.

Part Segmentation on ShapeNetPart. We utilize input point clouds of $N = 2048$ points without normals and employ cross-entropy as the loss function. Training is performed using the AdamW optimizer with cosine decay, an initial learning rate of 0.0002, a batch size of 16, and 300 training epochs.

Data Augmentation.

We define two different data augmentation methods: Translation-Scaling (TS) and Rotation (R). We also conduct comparative experiments on different datasets, with the results shown in the table 1. On ModelNet40, the 3DSMT model with TS data augmentation achieves improved performance, while the R data augmentation method performs better on the most challenging variant dataset (PB_T50_RS) of ScanObjectNN.

Methods	ModelNet40		ScanObjectNN(PB_T50_RS)	
	OA (%)	mAcc (%)	OA (%)	mAcc (%)
TS	94.7	91.8	94.1	91.0
R	89.3	85.6	90.4	86.3

Table 1: Performance comparisons of different data augmentation methods on both the ModelNet40 and the ScanObjectNN(PB_T50_RS) datasets.

Spiking Neurons	OA (%)	mAcc (%)
LIF	93.8	90.9
IF	94.7	91.8

Table 2: Effect of different spiking neurons to the accuracy performance on the ModelNet40 dataset.

Effect of Different Spiking Neurons.

The core of SNN is spiking neurons. The IF neuron only requires accumulation and threshold comparison, making it easy to implement on neuromorphic chips with extremely low energy consumption, especially suitable for resource-constrained edge devices. The leakage mechanism of LIF neurons requires additional circuits to simulate membrane potential decay, increasing hardware complexity and energy consumption. We conducted an ablation study on the performance impact of these two neurons on 3DSMT, and the results are shown in the Table 2. The IF neuron is more suitable for application in large-scale complex point cloud models.

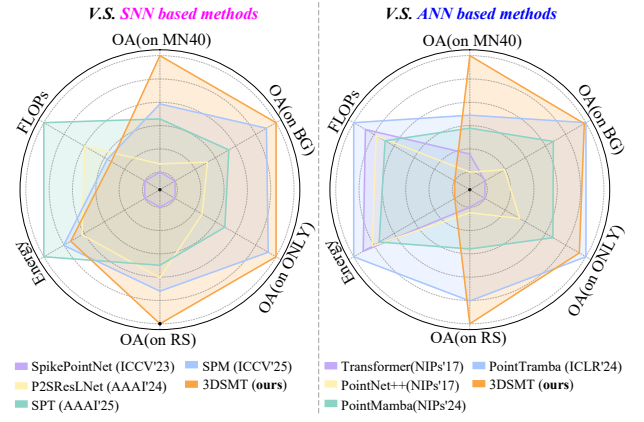


Figure 1: Comparison of our proposed method with SNN and ANN based methods in terms of energy consumption, FLOPs, overall accuracy (OA) on ModelNet40 dataset (denoted as MN40) and three variants of ScanObjectNN dataset, including OBJ_BG (denoted as BG), OBJ_ONLY (denoted as ONLY) and PB_T50_RS (denoted as RS).

Analysis on Efficiency.

We evaluate the efficiency of the 3DSMT model based on two metrics: runtime duration and memory consumption. The experimental equipment and specific configurations are described in Section Experiment Configuration. The ablation experiments adopt a setting of 3 Timestep and calculate the results for one epoch. As shown in Table 3, 3DSMT can be implemented on a single GPU with 24GB video memory, featuring relatively short training and testing times and low video memory usage, providing a new solution for large-scale point cloud processing.

Metric	Train	Test
Runtime	150s	27s
Memory	19.6G	9.2G

Table 3: Analysis of Efficiency on the ModelNet40 dataset.

Comparison to ANN and SNN-based Methods

To clearly demonstrate the excellent balance between performance and energy consumption of 3DSMT, we conducted a multi-dimensional method comparison (as shown in Figure 1): When compared with SNN-based methods, 3DSMT achieved leading OA accuracy in multiple point cloud classification tasks such as ModelNet40(MN40), OBJ_BG (BG), OBJ_ONLY (ONLY), and PB_T50_RS (RS). Meanwhile, it met the requirements of low FLOPs and low energy consumption, breaking through the trade-off bottleneck between accuracy and efficiency of traditional SNN. When compared with ANN-based methods, 3DSMT not only surpassed most ANN methods in OA accuracy but also achieved extremely low FLOPs and energy consumption loss. The proposed 3DSMT demonstrates the comprehensive advantage of "high accuracy - low energy" in point cloud processing,

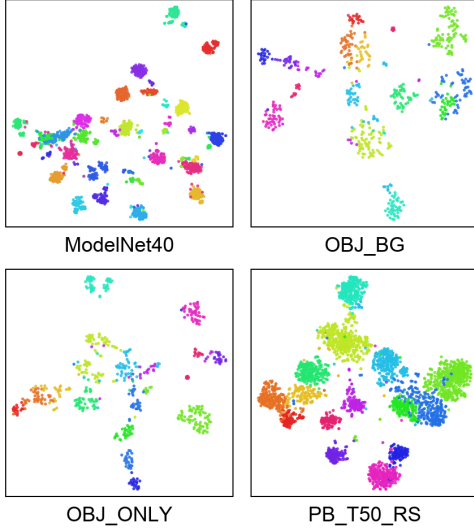


Figure 2: Visualizing Classification Feature Learning on Different Datasets.

providing new ideas for the design of point cloud models that balance performance and deployment requirements.

Energy Calculation

In artificial neural network (ANN), the number of floating-point operations (FLOPs) is an important metric for evaluating model computational complexity. In contrast, spiking neural network (SNN) use synaptic operations (SOPs) as the fundamental measure of computation. SOPs refer to the synaptic transmission process that occurs when spikes are generated. Unlike the continuous activation in ANN, SNN only trigger computations when spikes occur, significantly reducing energy consumption. The calculation formula for SOPs is as follows:

$$\text{SOPs}^l = f_r \times T \times \text{FLOPs}(l) \quad (4)$$

where l denotes the index of the spiking layer in 3DSMT; f_r refers spike firing rate; T is the time step; $\text{FLOPs}(l)$ represents the number of floating-point operations (specifically multiply-and-accumulate operations, MAC); SOPs represents the quantity of spike-driven accumulation (AC) operations. 3DSMT is designed with biological plausibility in mind, and ideally, it can be directly deployed on neuromorphic hardware. The inference energy consumption of 3DSMT can be expressed as:

$$E_{3\text{DSMT}} = E_{\text{MAC}} \times (\text{FLOPs}_{\text{Conv}}^1 + \text{FLOPs}_{\text{FC}}) + E_{\text{AC}} \times \left(\sum_{l=1}^L \text{SOPs}^l \right) \quad (5)$$

where $\text{FLOPs}_{\text{Conv}}^1$ represents the floating-point operations of the first floating-point convolutional layer in the Spiking Patch Embedding module, and FLOPs_{FC} represents the floating-point operations of the fully connected (FC) layers

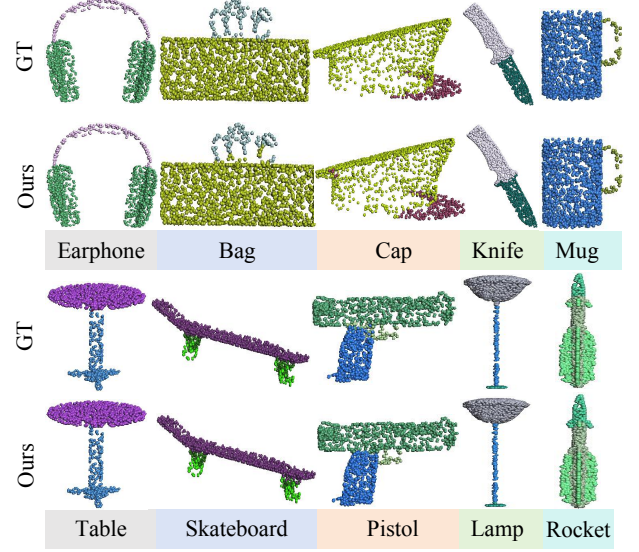


Figure 3: Visualization of Part segmentation results on ShapeNetPart. Top line is our prediction and bottom line is ground truth.

in various downstream tasks. We assume that both MAC and AC operations are implemented on SpikingJelly (Fang et al. 2023), where $E_{\text{MAC}} = 4.6 \text{ pJ}$ and $E_{\text{AC}} = 0.9 \text{ pJ}$ (Kundu, Pedram, and Bearel 2021; Hu et al. 2021; Horowitz 2014; Kundu et al. 2021).

Visualization

To explore the feature representation capability of the 3DSMT model, we use t-SNE (Van der Maaten and Hinton 2008) to map high-dimensional features into a 2D space for visual analysis of feature clustering effects across different datasets. As shown in Figure 2, feature points of the ModelNet40 dataset exhibit a diffused distribution with partial inter-class overlaps, though core feature clusters show obvious clustering trends. Feature points of the OBJ_BG dataset form distinct cluster separations with clear class boundaries, while those of the OBJ_ONLY dataset are highly aggregated with minimal inter-class overlap. For the PB_T50_RS dataset, although feature distributions slightly diffuse due to complex noise, major class clusters maintain compact structures. These results verify the strong capability of 3DSMT in capturing features of unordered point clouds.

We also provide more part segmentation visualizations to quantitatively demonstrate the effectiveness and performance of 3DSMT, as shown in Figure 3. The part segmentation results of 3DSMT are almost identical to the ground truth across different categories. Slightly more complex categories may exhibit minor discrepancies, but these do not significantly affect the overall performance.

Neuromorphic Hardware Rationality

We expect to directly deploy 3DSMT on neuromorphic hardware. However, designing RC circuits for processing

3D point clouds on neuromorphic hardware requires additional engineering efforts. Point cloud data typically exists in floating-point format, so it is a natural choice to adopt an Embedding method with "multiply-accumulate (MAC)" operations at the network's initial stage to generate spike signals for each point. Additionally, to pursue higher classification accuracy, the final fully connected (FC) layer needs to perform some MAC operations, which are difficult to implement on neuromorphic hardware. Although there is currently no method combining laser scanning with an event-driven mechanism to generate 3D point cloud spikes from the source, theoretically, our Spiking Local Offset Attention, Spiking Mamba Block, and Spiking Position Encoding can all be implemented on neuromorphic hardware. They can convert floating-point point clouds into spike form before performing convolution (Conv) operations and execute accumulation (AC) operations to accumulate the weights of postsynaptic neurons.

References

- Bulsara, A. R.; Elston, T. C.; Doering, C. R.; Lowen, S. B.; and Lindenberg, K. 1996. Cooperative behavior in periodically driven noisy integrate-fire models of neuronal dynamics. *Physical Review E*, 53(4): 3958.
- Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.
- Fang, W.; Chen, Y.; Ding, J.; Yu, Z.; Masquelier, T.; Chen, D.; Huang, L.; Zhou, H.; Li, G.; and Tian, Y. 2023. Spiking-jelly: An open-source machine learning infrastructure platform for spike-based intelligence. *Science Advances*, 9(40): eadi1480.
- Horowitz, M. 2014. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 10–14. IEEE.
- Hu, Y.; Wu, Y.; Deng, L.; and Li, G. 2021. Advancing Residual Learning towards Powerful Deep Spiking Neural Networks. *arXiv preprint arXiv:2112.08954*.
- Kundu, S.; Datta, G.; Pedram, M.; and Beerel, P. A. 2021. Spike-thrift: Towards energy-efficient deep spiking neural networks by limiting spiking activity via attention-guided compression. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision (WACV)*, 3953–3962.
- Kundu, S.; Pedram, M.; and Beerel, P. A. 2021. Hire-snn: Harnessing the inherent robustness of energy-efficient deep spiking neural networks by training with crafted input noise. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 5209–5218.
- Uy, M. A.; Pham, Q.-H.; Hua, B.-S.; Nguyen, T.; and Yeung, S.-K. 2019. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1588–1597.
- Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).
- Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3D shapenets: A deep representation for volumetric shapes. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1912–1920.
- Yi, L.; Kim, V. G.; Ceylan, D.; Shen, I.-C.; Yan, M.; Su, H.; Lu, C.; Huang, Q.; Sheffer, A.; and Guibas, L. 2016. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6): 1–12.