Primary Visual Cortex Inspired Point Cloud Analysis Framework

Jisheng Dang*[1], Delin Deng*[1], Bimei Wang†[2], Jingze Wu[3], Hui Zhang[4], Haijiang Li[5], Jingmei Jiao[5], Dengyue Pan[1], Mangang Xie[4], Jizhao Liu[1],

[1]School of Information Science and Engineering, Lanzhou University
[2]College of Cyber Security, Jinan University
[3]School of Systems Science and Engineering, Sun Yat-sen University
[4]College of Computer Science and Engineering, Northwest Normal University
[5]School of Electronics and Information Engineering, Lanzhou Jiaotong University
wangbm@stu2021.jnu.edu.cn

Abstract

Despite significant advancements in point cloud analysis, reducing energy consumption and improving robustness remain understudied, largely due to the inherent limitations of Convolutional Neural Networks (CNNs). To address this, we take the cue from the primary visual cortex and propose a Dendritic-Connected Continuous-Coupled Neural Network (DC-CCNN), a novel Brain-Inspired Neural Network (BINN) architecture tailored for point cloud analysis. By leveraging the unique characteristics of point clouds, our design combines discrete and continuous encoding, replacing traditional Multilayer Perceptrons (MLPs) with more efficient and robust BINNs. Our approach substantially improves the performance of Brain-Inspired Neural Networks on point analysis tasks and maintaining performance comparable to state-of-the-art methods. Furthermore, DC-CCNN exhibits enhanced robustness against various point cloud deformations and corruptions. Our experimental results demonstrate that DC-CCNN achieves competitive performance on benchmark datasets, making it a promising alternative to traditional deep learning methods for point cloud analysis. With its high efficiency and robustness, DC-CCNN has the potential for widespread adoption in 3D computer vision, robotics, and autonomous systems.

Introduction

Point cloud analysis plays a central role in 3D understanding, drawing increasing attention from both academia and industry (Zheng et al. 2025). Unlike 2D images, point clouds consist of unordered, irregularly spaced 3D points, making conventional image processing ineffective. Sparsity and noise further increase the difficulty. To address these challenges, various approaches have been explored, including geometric methods, deep learning, and Graph Neural Networks (GNNs). Deep learning methods for point clouds are generally categorized by their data representation strategies. Indirect methods project point clouds onto grids (e.g., voxels (Lei et al. 2024) or images (Dang et al. 2023b; Dang and Yang 2022; Dang et al. 2025, 2024a)) to reuse 2D CNNs, but this often causes information loss. Direct methods, such as PointNet (Qi et al. 2017a), PointNet++ (Qi et al. 2017b), and PointMLP (Ma et al. 2022), process raw point sets using shared MLPs, improving efficiency and accuracy. However, most models assume clean data and degrade under corruption (Dong et al. 2023). Robust 3D classification remains an open challenge. Despite progress, efficiently handling large-scale, sparse, and noisy point clouds remains difficult. CNN-based methods often suffer from high computational cost, limited robustness to deformation, and poor adaptability to irregular structures (Dang and Yang 2020, 2021; Dang et al. 2024b). Their reliance on dense convolutions restricts deployment on low-power devices and hampers generalization under occlusion or noise.

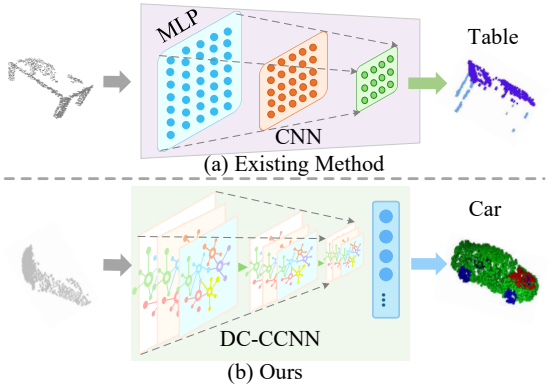

Figure 1: Comparison of CNN-based methods and our DC-CCNN for 3D point cloud classification. DC-CCNN achieves similar accuracy with significantly lower energy consumption.

To address these limitations, we propose *Dendritic-Connected Continuous-Coupled Neural Network (DC-CCNN)*, a novel Brain-Inspired Neural Network (BINN) architecture designed for robust and efficient point cloud analysis. As illustrated in Figure 1, DC-CCNN leverages Spiking Neural Networks (SNNs) (Wu et al. 2025), incorporating bi-

*These authors contributed equally.
†Corresponding author.

ological sensitivity to spatial and temporal patterns. Through spike coding and event-driven computation, it reduces complexity while improving robustness to deformation, occlusion, and noise. The integration of discrete and continuous dynamics enables high efficiency with low energy consumption, achieving performance competitive with state-of-the-art models using fewer computational resources. DC-CCNN offers strong potential for applications in 3D vision, robotics, and autonomous driving. The main contributions of this work are:

- We propose DC-CCNN, a brain-inspired neural architecture integrating discrete and continuous coding to reduce computational cost while maintaining high performance.
- DC-CCNN improves robustness against noise, deformation, and occlusion, outperforming conventional CNNs in stability and adaptability.
- Extensive experiments on benchmark datasets validate its effectiveness and efficiency, positioning DC-CCNN as a competitive alternative for point cloud analysis.

Related Work

Point Cloud Analysis Methods. Classical point cloud methods rely on geometric and statistical techniques such as segmentation, feature extraction, and voxelization (Li et al. 2022; Gonizzi Barsanti et al. 2024). While computationally efficient, they are limited by sensitivity to noise and poor scalability to large, sparse, and unordered data.

Convolutional Neural Networks. Deep learning has become the dominant paradigm for point cloud analysis. PointNet (Qi et al. 2017a) addresses unordered data using symmetric functions, while PointNet++ (Qi et al. 2017b) introduces hierarchical feature learning. Despite strong performance, CNN-based models are resource-intensive (Ma et al. 2024), facing scalability and latency challenges on large-scale data.

Spiking Neural Networks. Spiking Neural Networks. (SNNs) (Yi et al. 2023) emulate biological neurons through discrete, asynchronous spikes (Hodgkin and Huxley 1952), enabling event-driven computation well-suited for sparse data. They have shown strong energy efficiency in sequential tasks (Zhong et al. 2024; Zhang et al. 2025; Liu et al. 2025) and HAR, achieving up to 94% energy savings. Recent efforts apply SNNs to point cloud processing (Ren et al. 2024; Wu et al. 2024; Takaghaj and Sampson 2024), leveraging their compatibility with spatio-temporal data. Models like Spiking PointNet (Ren et al. 2024) reduce training overhead and sometimes outperform DNNs, while PointLCA-Net (Takaghaj and Sampson 2024) incorporates the Locally Competitive Algorithm (LCA) to enhance classification. Architectures include feedforward and recurrent forms (e.g., RNNs, LSTMs), which capture temporal dependencies and mitigate gradient issues. However, the robustness of SNNs under corrupted or noisy point clouds remains underexplored.

Method

Early point cloud models lacked robustness to noise, sparsity, and irregularity. To address these challenges, we proposed a BINN module, as illustrated in Figure 2, which is inspired by biological neural systems. The module is sensitive to spatio-temporal signals and mimics brain-like processing to enhance performance. Spiking PointNet (Ren et al. 2024) later introduced BINN concepts into point cloud analysis, incorporating SNN features such as thresholding and exponential decay to reduce energy consumption. However, SNNs remain simplified abstractions of spiking neurons, differing from biological cortical structures. Their periodic responses under repetitive input contrast with the chaotic behavior of biological neurons, leading to information loss and limited feature extraction. As a result, Spiking PointNet suffers from reduced accuracy and limited practical applicability. To overcome these issues, we developed DC-CCNN, which introduces neuronal firing randomness and continuous outputs to mitigate information loss and better represent point cloud characteristics. Convolutional operations further improve local feature extraction, yielding a more effective processing framework.

Spiking Neural Networks

In complex structures like point clouds, SNNs mimic the brain's spike-based communication. Neurons receive spike trains from the previous layer and update their membrane potential, commonly modeled by the Leaky Integrate-and-Fire (LIF) equation:

$$\tau_m \frac{du}{dt} = -(u - u_{rest}) + R \cdot I(t), \quad u < V_{th}, \tag{1}$$

where I is the input current, V_{th} the firing threshold, and R, τ_m the resistance and time constant. When u reaches V_{th}, a spike is emitted and u resets to u_{rest} (typically zero).

To enable training in modern frameworks (e.g., TensorFlow, PyTorch), an iterative LIF model was proposed:

$$u_i[t+1] = \lambda(u_i[t] - V_{th} s_i[t]) + \sum_j w_{ij} s_j[t] + b_i, \tag{2}$$

$$s_i[t+1] = H(u_i[t+1] - V_{th}), \tag{3}$$

where w_{ij} is the synaptic weight, b_i the bias, and $s_j[t]$ the spike from neuron j at time t. $H(\cdot)$ is the Heaviside function, and $\lambda < 1$ is the leak factor (typically 0.2–0.25). $s_i[t]$ denotes the spike output of neuron i at time t.

Continuous-Coupled Neural Network

To more accurately model the primary visual cortex and address issues such as the refractory period present in existing models, we introduce the Continuous-Coupled Neural Network (CCNN). CCNN introduces stochastic firing dynamics as opposed to the deterministic nature of Pulse-Coupled Neural Networks (PCNNs). The neuron firing state in CCNN fluctuates due to a fluctuating threshold, mimicking the chaotic dynamics observed in real primary visual cortex neurons. The membrane potential x follows a Gaussian distribution for both the resting and firing states. The firing probability is computed using Bayesian inference.

The core dynamics of the CCNN model are described by the following recurrence relations:

$$F_{ij}(n) = e^{-a_f} F_{ij}(n-1) + V_F M_{ijkl} Y_{kl}(n-1) + S_{ij}, \tag{4}$$

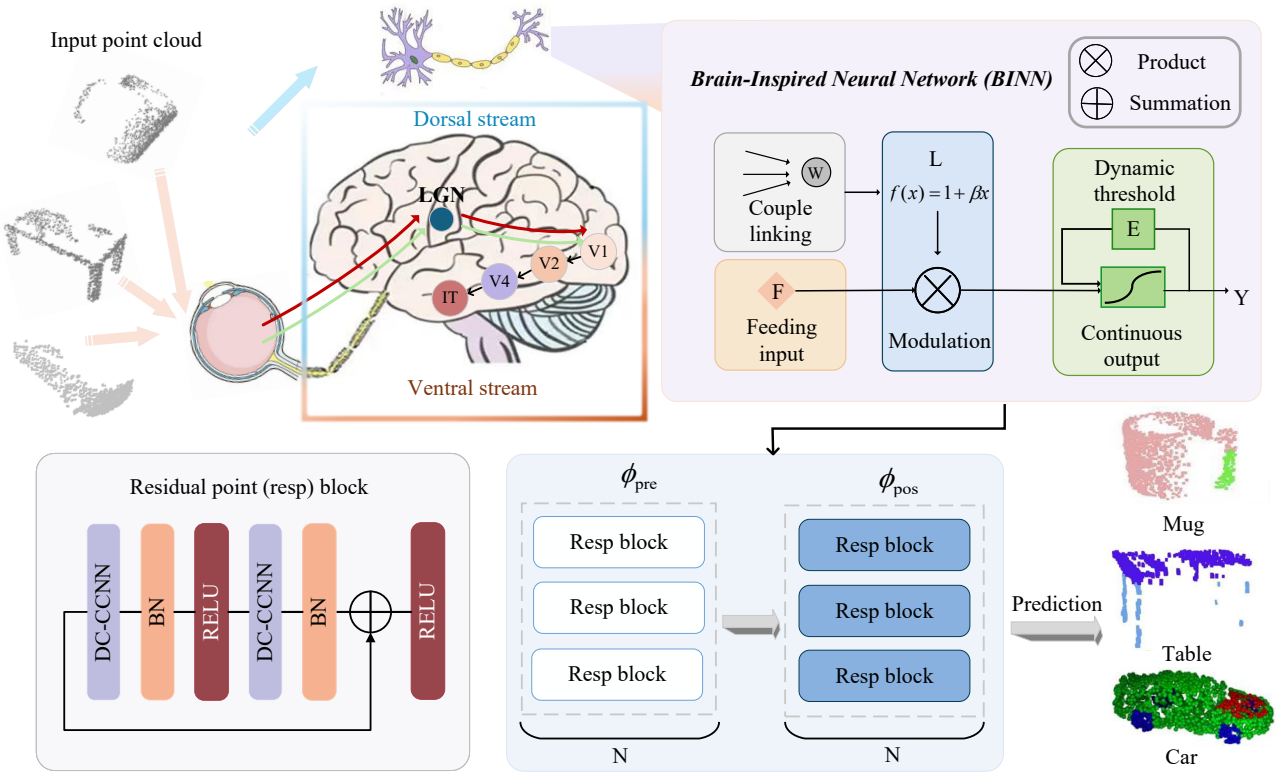

Figure 2: Pipeline overview of DC-CCNN for 3D point cloud classification. Inspired by the visual cortex, the framework mimics the dorsal stream for spatial processing and the ventral stream for object recognition. Input point clouds are first processed by CNNs to extract spatial features, which are mapped into structured representations via DC-CCNN. The resulting features are then classified, emulating intertemporal cortex functionality.

$$L_{ij}(n) = e^{-a_l} L_{ij}(n-1) + V_L W_{ijkl} Y_{kl}(n-1), \quad (5)$$

$$U_{ij}(n) = F_{ij}(n)(1 + bL_{ij}(n)), \quad (6)$$

$$Y_{ij}(n) = \frac{1}{1 + e^{-(U_{ij}(n) - E_{ij}(n))}}, \quad (7)$$

where M_{ijkl} and W_{ijkl} are the synaptic weights, S_{ij} is the external stimulus, and b is the linking strength. The parameters a_f, a_l, and a_e control the exponential decay factors of previous input states. The firing threshold $E_{ij}(n)$ evolves according to the following equation:

$$E_{ij}(n) = e^{-a_e} E_{ij}(n-1) + V_E Y_{ij}(n-1). \quad (8)$$

The firing process is treated as a random fluctuation due to the hyperpolarization process, where both the resting potential and firing threshold are modeled as Gaussian-distributed random variables. The probability of firing is computed using:

$$P(Y_1|X) = \frac{1}{1 + e^{-(2m_1 - 2m_0)x + m_1^2 - m_0^2 2s^2 + \ln \frac{P(Y_0)}{P(Y_1)}}}. \quad (9)$$

The model's neurons exhibit periodic dynamics under constant stimuli and chaotic behavior under periodic stimuli. This is a key difference from PCNN, where the output remains periodic even with a periodic input. In contrast, CCNN shows chaotic characteristics under AC stimuli, aligning with the observed behaviors in real primary visual cortex neurons.

We propose a brain-inspired model for point cloud analysis that enhances spatiotemporal feature extraction. The framework combines a BINN module, which encodes spatial and temporal characteristics via spike-based processing, with a CCNN that models complex dynamics through iterative computation. The pipeline consists of three stages: (1) preprocessing for denoising and normalization, (2) BINN-based spatiotemporal spike encoding, and (3) CCNN-based pattern extraction and uncertainty handling. This BINN–CCNN synergy improves both dynamic modeling and spatiotemporal representation in point cloud data.

DC-CCNN Framework

The BINN module first processes the input by simulating the pulse transmission mechanism of biological neural systems through its spiking neuron model. BINNs excel at handling sparse and irregular data, enabling them to capture both temporal and spatial features in the input. Particularly for dynamic and sparse data, BINNs can extract key features through spike generation and transmission. These initially extracted features serve as the foundation for subsequent processing by the CCNN module. The CCNN module receives the feature data output from the BINN module.

Its pulse convolutional layer further processes these features using a unique structure and state variables, including the input signal accumulation state F, the linking state L, and the dynamic threshold E.

Accumulation of Output Features of F. The input signal accumulation state F aggregates the characteristic signals output by the BINN module , simulating the continuous perception of stimulus intensity in the biological visual system and further enhancing feature representation.

$$F_t = \left(1 - \frac{\Delta t}{\tau_F}\right) F_{t-1} + X. \tag{10}$$

Among them, F_t represents the accumulation state at time t; X is the input signal; τ_F is the time-constant, which controls the decay rate of the input signal, Δt represents the time step between the current and the previous time point; at the moment $t = 0$, $F = X$, and the input X is the 1×1 feature matrix corresponding to the input of the convolutional kernel.

The Spatial Neighborhood Coupling Effect of L. The Linking state L models spatial neighborhood coupling by integrating the spatial relationships of the BINN output features. It simulates the effect of a pixel (neuron) being excited by its surrounding pixels, enabling the network to better capture spatial structure information.

$$L_t = \left(1 - \frac{\Delta t}{\tau_L}\right) L_{t-1} + Conv(F), \tag{11}$$

where L_t represents the coupling state at time t; $Conv(F)$ is a convolution operation that maps the accumulation state F to the same number of channels as L; τ_L is the time-constant, which controls the decay rate of the coupling state, Δt represents the time step between the current and the previous time point; another definition is $F_{projected} = Conv(F)$.

Dynamic Threshold Control of E. The dynamic threshold E regulates neuronal activation over time by adapting to accumulated input signals and local spatial interactions. This mechanism simulates homeostatic regulation in biological neurons, enabling the model to suppress background noise while enhancing sensitivity to salient patterns. The threshold rises or falls based on the difference between spike excitation U and the current threshold E, promoting adaptive filtering under varying input conditions.

$$E_t = \left(1 - \frac{\Delta t}{\tau_E}\right) E_{t-1} + V_E \cdot sigmoid(U - E), \tag{12}$$

where U is the spike excitation, $U = F_{\text{projected}} \cdot (1 + \beta \cdot L_t)$; β is the coupling-strength control parameter, E_t is the dynamic threshold at time t, τ_E is the time-constant, which controls the decay rate of the dynamic threshold; $sigmoid(U - E)$ is the threshold adjustment amount; V_E is the adjustment - rate control parameter.

$$Y = sigmoid(U - E). \tag{13}$$

The sigmoid function is used to smooth the output.

Method	OA(%)
Spiking PointNet (Ren et al. 2024)	83.5
RS-CNN (Liu et al. 2019c)	92.9
DensePoint (Liu et al. 2019b)	93.2
PointASNL (Yan et al. 2020)	92.9
PosPool (Liu et al. 2020)	93.2
Point Trans (Engel, Belagiannis, and Dietmayer 2021)	92.8
MLMSPT (Zhong and Han 2021)	92.9
PCT (Guo et al. 2021)	93.2
PointMLP (Ma et al. 2022)	93.2
DC-CCNN (Ours)	**93.5**

Table 1: Classification results on ModelNet40 dataset.

Method	mAcc(%)	OA(%)
BGA-DGCNN (Uy et al. 2019)	75.7	79.7
BGA-PN++ (Uy et al. 2019))	77.5	80.2
DRNet (Qiu, Anwar, and Barnes 2021a)	78.0	80.3
GBNet (Qiu, Anwar, and Barnes 2021b)	77.8	80.5
SimpleView (Goyal et al. 2021)	-	80.5
PRANet (Cheng et al. 2021)	79.1	82.1
MVTN (Hamdi, Giancola, and Ghanem 2021)	-	82.8
PointMLP (Ma et al. 2022)	83.9	85.4
DC-CCNN (Ours)	**84.1**	**87.3**

Table 2: Classification results on ScanObjectNN dataset.

Experiment

In this section, we conduct a thorough evaluation of DC-CCNN across multiple benchmarks, supported by detailed ablation studies that showcase its effectiveness through both quantitative and qualitative analysis.

Datasets

We evaluate our model on three standard point cloud benchmarks. ModelNet40 (Wu et al. 2015) includes 12,311 CAD models from 40 categories, with 9,843 used for training and 2,468 for testing. ScanObjectNN (Uy et al. 2019) comprises over 15,000 real-world object point clouds from RGB-D indoor scenes; we use the PB-T50-RS split. ShapeNetPart (Yi et al. 2016) contains 16,881 shapes from 16 categories, each annotated with 50 fine-grained part labels, making it suitable for part segmentation.

Implement Details

We simulate the primary visual cortex to improve neural efficiency in dynamic input processing. This work explores the relationship between point and visual streams, drawing on the distinct roles of the dorsal (spatial and motion perception) and ventral (object recognition) pathways. By integrating a CCNN into this framework, we aim to better replicate brain-like processing for dynamic perception tasks. The CCNN module is combined with traditional MLPs to form DC-CCNN (Figure 2), designed to extract features from

Method	Cls. mIoU	Inst. mIoU	aero	bag	cap	car	chair	phone	guitar	knife	lamp	laptop	motorbike	mug	pistol	rocket	skateboard	table
PointNet++ (Qi et al. 2017b)	81.9	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
Kd-Net (Klokov and Lempitsky 2017)	-	82.3	80.1	74.6	74.3	70.3	88.6	73.5	90.2	87.2	81.0	94.9	57.4	86.7	78.1	51.8	69.9	80.3
SO-Net (Li, Chen, and Lee 2018)	-	84.9	82.8	77.8	88.0	77.3	90.6	73.5	90.7	83.9	82.8	94.8	69.1	94.2	80.9	53.1	72.9	83.0
DGCNN (Wang et al. 2019)	82.3	85.2	84.0	**83.4**	86.7	77.8	90.6	74.7	91.2	87.5	82.8	95.7	66.3	94.9	81.1	63.5	74.5	82.6
P2Sequence (Liu et al. 2019a)	-	85.2	82.6	81.8	87.5	77.3	90.8	77.1	91.1	86.9	83.9	95.7	70.8	94.6	79.3	58.1	75.2	82.8
PointASNL (Yan et al. 2020)	-	86.1	84.1	84.7	87.9	79.7	**92.2**	73.7	91.0	87.2	84.2	95.8	74.4	95.2	81.0	63.0	76.3	83.2
PCNN (Atzmon, Maron, and Lipman 2018)	81.8	85.1	82.4	80.1	85.5	79.5	90.8	73.2	91.3	86.0	**85.0**	95.7	73.2	94.8	83.3	51.0	75.0	81.8
SpiderCNN (Xu et al. 2018)	82.4	85.3	83.5	81.0	87.2	77.5	90.7	76.8	91.1	87.3	83.3	95.8	70.2	93.5	82.7	59.7	75.8	82.8
PointMLP (Ma et al. 2022)	83.9	85.9	**83.4**	**83.4**	87.3	**80.5**	89.9	**78.4**	92.0	88.1	82.6	**96.2**	**78.1**	95.9	85.4	64.8	83.0	**84.7**
DC-CCNN(Ours)	**84.6**	**86.5**	**84.3**	83.0	**89.4**	78.4	**92.2**	77.1	**92.5**	**88.6**	82.6	95.3	76.2	**96.3**	**85.9**	**68.6**	**83.5**	83.2

Table 3: Part segmentation results on the point cloud segmentation ShapeNetPart dataset.

Dataset	Model	Standard	Global Sparsity					Occlusion				
			512	256	200	150	128	0.1	0.3	0.5	0.7	1.0
ModelNet40 (Wu et al. 2015)	PointMLP (Ma et al. 2022)	93.2	92.1	86.6	75.9	51.7	37.8	92.7	92.4	91.7	86.1	47.8
	DC-CCNN (Ours)	**93.4**	**92.6**	**89.7**	**83.8**	**71.5**	**60.3**	**93.1**	**93.2**	**92.3**	**88.1**	**51.3**
ScanObjectNN (Uy et al. 2019)	PointMLP (Ma et al. 2022)	85.4	81.1	77.3	69.1	44.2	39.7	83.7	77.4	72.0	61.6	50.5
	DC-CCNN (Ours)	**87.3**	**84.3**	**81.4**	**74.8**	**63.7**	**54.8**	**86.0**	**83.5**	**79.3**	**70.4**	**63.9**

Table 4: Classification results on the ModelNet40 dataset (Wu et al. 2015) and ScanObjectNN (Uy et al. 2019) dataset with density corruption, including overall accuracy metric (%). **Bold** represent the top-1 results.

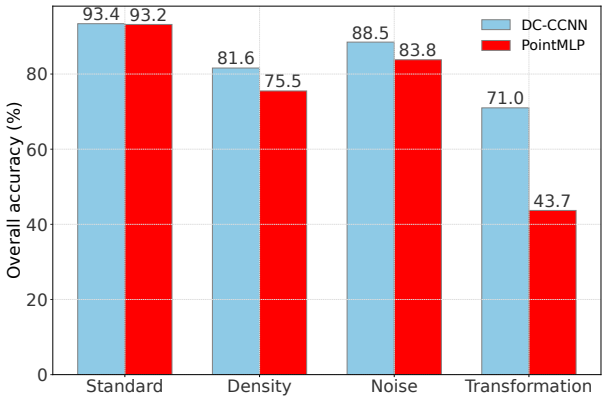

Figure 3: Comparison of classification accuracy between PointMLP and DC-CCNN on ModelNet40 Dataset (Wu et al. 2015) and three common corruption patterns.

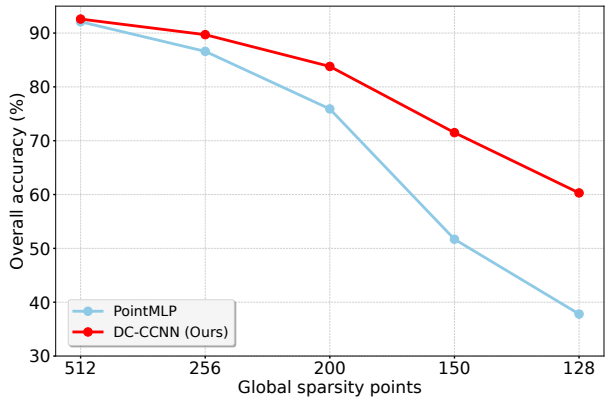

Figure 4: Comparison of PointMLP and its DC-CCNN enhanced variants on the ModelNet40 dataset (Wu et al. 2015) with global sparsity corruption.

event streams, particularly in the visual domain. The architecture incorporates convolutional and pooling layers for high-level feature extraction, enabling modular integration of CCNN and MLPs. This hybrid design retains the feature learning strength of MLPs while leveraging CCNN's lack of a refractory period for real-time, high-speed processing.

To enhance training and robustness, residual modules with skip connections mitigate gradient vanishing and explosion, promoting stable signal propagation and faster convergence. In DC-CCNN, these blocks support effective impulse signal transmission. Within CCNN, neurons communicate and update via coupling links. Input signals are modulated through a dynamic threshold function, enabling adaptive firing behavior. This mechanism simulates biologically observed dynamics by adjusting neuronal output based on input strength. Code will be included in the supplementary material.

Quantitative Analysis

We conduct comprehensive experiments on both object classification and part segmentation tasks using three widely adopted benchmark datasets: ModelNet40, ScanObjectNN, and ShapeNetPart. The results consistently demonstrate the effectiveness and robustness of our DC-CCNN framework

Dataset	Model	Standard	Guassion				Salt-and-Pepper					Spatial
			0.01	0.03	0.05	0.09	0.01	0.03	0.1	0.3	0.5	Random
ModelNet40 (Wu et al. 2015)	PointMLP (Ma et al. 2022)	93.2	88.7	88.4	88.0	79.7	92.3	92.0	89.5	86.6	49.2	43.7
	DC-CCNN (Ours)	**93.4**	**91.9**	**90.9**	**90.7**	**90.8**	**92.7**	**92.9**	**92.4**	**91.5**	**62.8**	**71.0**
ScanObjectNN (Uy et al. 2019)	PointMLP (Ma et al. 2022)	85.4	80.0	80.0	78.6	64.9	84.3	82.8	74.5	67.3	37.3	47.9
	DC-CCNN (Ours)	**87.3**	**83.8**	**83.1**	**83.3**	**79.6**	**84.8**	**83.2**	**79.1**	**73.6**	**54.1**	**68.5**

Table 5: Classification results on the ModelNet40 dataset (Wu et al. 2015) and ScanObjectNN dataset (Uy et al. 2019) with corruption are reported, including overall accuracy (%) for both noise and spatial corruption. **Bold** represent the top-1 results.

Model	Standard	Global Sparsity					Salt-and-Pepper				
		512	256	200	150	128	0.02	0.03	0.1	0.3	0.5
PointMLP (Ma et al. 2022)	93.2	92.1	86.6	75.9	51.7	37.8	92.3	92.0	89.5	86.6	49.2
DC-CCNN (**Signal Accumulation**)	**93.5**	92.3	89.6	**84.9**	**73.0**	**60.9**	**92.8**	92.8	92.3	90.8	53.0
DC-CCNN (**Spatial Coupling**)	93.3	92.1	88.7	82.4	69.4	58.2	61.0	92.7	**92.4**	90.2	**66.9**
DC-CCNN (**Dynamic Threshold**)	93.2	92.2	88.3	81.1	68.2	54.3	92.2	92.8	92.2	90.6	58.1
DC-CCNN (**Ours**)	93.4	**92.6**	**89.7**	83.8	71.7	60.3	92.7	**92.9**	**92.4**	**91.5**	62.8

Table 6: Ablation studies were conducted on the ModelNet40 dataset (Wu et al. 2015) with corruption, evaluating overall classification accuracy (%) under varying levels of global sparsity and salt-and-pepper noise. **Bold** and underline represent the top-1 and top-2 results.

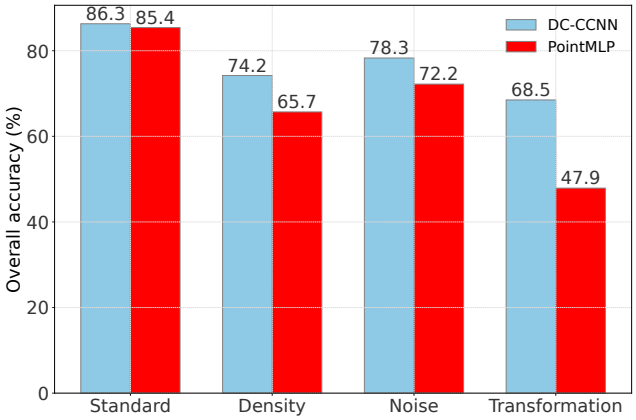

Figure 5: Comparison of classification accuracy between PointMLP and DC-CCNN on ScanObjectNN Dataset (Uy et al. 2019) and three common corruption patterns.

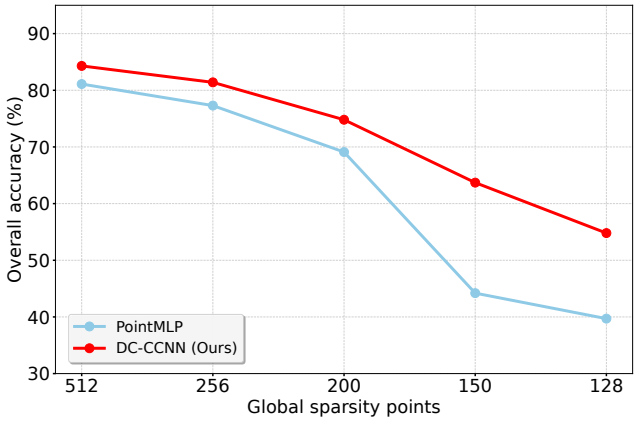

Figure 6: Comparison of PointMLP and its DC-CCNN enhanced variants on the ScanObjectNN dataset (Uy et al. 2019) with global sparsity corruption.

across clean, noisy, and fine-grained scenarios, including various corruption types and point densities. Specifically, on the clean and well-structured ModelNet40 dataset (Wu et al. 2015), our DC-CCNN achieves an overall accuracy of 93.5%, which is 10.04% higher than the brain-inspired Spiking PointNet (Ren et al. 2024), setting a new state-of-the-art among similar biologically motivated models for static object classification.

On the challenging real-world dataset ScanObjectNN (Uy et al. 2019), which contains occlusions, background clutter, and sensor noise, our model achieves 86.3% overall accuracy, outperforming all existing methods and confirming its strong generalization ability in complex environments, as shown in Table 2. In the part segmentation task on ShapeNetPart (Yi et al. 2016), DC-CCNN obtains a class-wise mIoU (Cls.mIoU) of 84.6% and an instance-wise mIoU (Inst.mIoU) of 86.5%, as presented in Table 3. These results confirm that our SNN-based architecture not only excels in classification tasks but also generalizes effectively to fine-grained segmentation.

Method	Cls.mIoU(%)	Inst.mIoU(%)	OA(%)
Signal Accumulation	**84.9**	85.7	86.4
Spatial Coupling	83.7	86.4	86.3
Dynamic Threshold	83.9	85.9	85.2
DC-CCNN (Ours)	84.6	**86.5**	**87.3**

Table 7: Ablation studies conducted on the ShapeNetPart dataset for part segmentation tasks. Results are reported using class-wise mIoU (Cls.mIoU) and instance-wise mIoU (Inst.mIoU), both in percentage (%). Ablation studies on the ScanObjectNN dataset, reported by overall accuracy (OA).

Robustness Studies

To assess the robustness of DC-CCNN under real-world degradations, we adopt a benchmark inspired by ModelNet40-C (Sun et al. 2022) and PointCloud-C (Ren, Pan, and Liu 2022), covering typical perturbations in autonomous driving and indoor perception. Following established protocols, we evaluate model performance under three corruption types: density-based sparsity, random noise, and geometric transformations.

Density Corruption. Point clouds from LIDAR, VR scanners, or CAD models often vary in density (Sun et al. 2022), especially in resource-limited settings where downsampling reduces computational cost. We simulate *global sparsity* by randomly retaining 512, 256, 200, 150, and 128 points from the original 1024, testing the model's ability to handle significant point loss. For *occlusion*, we apply DBSCAN (Ester et al. 1996) to identify spatial clusters and proportionally discard points (Zhou, Park, and Koltun 2018), emulating partial visibility in real-world scenes.

Noise Corruption. To simulate sensor noise (Wolff et al. 2016), we add *Gaussian* noise with varying standard deviations. We also introduce *salt-and-pepper* noise by randomly setting a percentage of points to 0 or 1, representing outliers and sensor glitches (Dang et al. 2023a).

Transformation Corruption. To evaluate the model's invariance to object pose and scale (Zhao et al. 2020), we perform spatial transformations including translation, rotation, and scaling, covering both global and local spatial distortions.

Robustness results are reported in Table 4 and Table 5. Across all corruption types, DC-CCNN consistently outperforms PointMLP. On ModelNet40, it achieves 60.3% accuracy under extreme sparsity (128 points), 22.5% higher than PointMLP. Under heavy Gaussian noise (std = 0.09), it maintains 90.8% accuracy, exceeding the baseline by over 11%. With 50% salt-and-pepper noise, DC-CCNN scores 62.8% vs. 49.2%. Similar improvements appear on ScanObjectNN, with gains exceeding 9% under sparsity and nearly 11% under noise. Notably, DC-CCNN maintains strong clean-data performance while significantly improving robustness under degradation.

Figures 3 and 4 visualize DC-CCNN's accuracy under increasing sparsity and perturbations on ModelNet40, confirming its advantage across challenging scenarios. Similar trends are observed on ScanObjectNN, as shown in Figures 5 and 6. Together with the quantitative results, these comparisons highlight the effectiveness of our biologically inspired design in enhancing robustness across both synthetic and real-world corruptions.

Ablation Studies

To evaluate the contribution of each core module in DC-CCNN, we conduct ablation studies on ModelNet40 (Wu et al. 2015), ScanObjectNN (Uy et al. 2019), and ShapeNetPart (Yi et al. 2016). We analyze three biologically inspired components—Signal Accumulation, Spatial Coupling, and Dynamic Threshold—using single-module variants and comparing them with the full model.

On ModelNet40, we assess robustness under varying sparsity and salt-and-pepper noise. As shown in Table 6, with only 128 points retained, DC-CCNN achieves 60.3% accuracy, surpassing all variants and exceeding PointMLP by 22.5%. Signal Accumulation is most effective under extreme sparsity, giving the best accuracy at 128 and 150 points. Under salt-and-pepper noise, DC-CCNN reaches 92.9% accuracy at 0.03 corruption and maintains 62.8% at 0.5. Spatial Coupling performs best under heavy noise (66.9% at 0.5), demonstrating resilience to structured perturbations. These results show that all modules enhance robustness: signal accumulation improves sparsity tolerance, and spatial coupling strengthens resistance to local corruption.

On ScanObjectNN, which includes real-world noise and occlusion, the full DC-CCNN attains 87.3% overall accuracy, outperforming the spatial coupling variant (86.3%) in Table 7. While spatial interactions aid in cluttered scenes, the full model generalizes better, suggesting that temporal and dynamic components prevent overfitting to local patterns.

On ShapeNetPart, each module improves different segmentation aspects. According to Table 7, signal accumulation yields the highest class-level mIoU, whereas spatial coupling gives the best instance-level mIoU. The full model integrates both advantages, achieving top performance on both metrics and demonstrating that jointly modeling temporal and spatial structure benefits fine-grained part segmentation.

In summary, each biologically inspired module contributes uniquely—signal accumulation enhances temporal stability, spatial coupling captures local structure, and dynamic thresholding enables adaptive noise filtering—while their integration provides state-of-the-art robustness and accuracy across tasks.

Conclusion

In this paper, we introduced the DC-CCNN, a novel brain-inspired neural network architecture designed specifically for point cloud analysis. By addressing the limitations of traditional CNNs and MLPs, DC-CCNN leverages discrete and continuous encoding to achieve state-of-the-art performance in brain-inspired methods while maintaining competitive results on benchmark datasets. Our experimental results demonstrate that DC-CCNN not only achieves advanced performance but also exhibits enhanced robustness against various point cloud deformations and corruptions.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. U24B20186). This work was also supported by the Supercomputing Center of Lanzhou University.

References

Atzmon, M.; Maron, H.; and Lipman, Y. 2018. Point convolutional neural networks by extension operators. *arXiv preprint arXiv:1803.10091*.

Cheng, S.; Chen, X.; He, X.; Liu, Z.; and Bai, X. 2021. Pranet: Point relation-aware network for 3d point cloud analysis. *IEEE Transactions on Image Processing*, 30: 4436–4448.

Dang, J.; and Yang, J. 2020. Hpgcnn: hierarchical parallel group convolutional neural networks for point clouds processing. In *Proceedings of the Asian conference on computer vision*.

Dang, J.; and Yang, J. 2021. HIGCNN: Hierarchical Interleaved Group Convolutional Neural Networks for Point Clouds Analysis. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Dang, J.; and Yang, J. 2022. LHPHGCNN: Lightweight hierarchical parallel heterogeneous group convolutional neural networks for point cloud scene prediction. *IEEE Transactions on Intelligent Transportation Systems*, 23(10): 18903–18915.

Dang, J.; Zheng, H.; Lai, J.; Yan, X.; and Guo, Y. 2023a. Efficient and robust video object segmentation through isogenous memory sampling and frame relation mining. *IEEE Transactions on Image Processing*, 32: 3924–3938.

Dang, J.; Zheng, H.; Wang, B.; Wang, L.; and Guo, Y. 2024a. Temporo-spatial parallel sparse memory networks for efficient video object segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 25(11): 17291–17304.

Dang, J.; Zheng, H.; Xu, X.; and Guo, Y. 2023b. Unified spatio-temporal dynamic routing for efficient video object segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 25(5): 4512–4526.

Dang, J.; Zheng, H.; Xu, X.; Wang, L.; and Guo, Y. 2024b. Beyond appearance: Multi-frame spatio-temporal context memory networks for efficient and robust video object segmentation. *IEEE Transactions on Image Processing*.

Dang, J.; Zheng, H.; Xu, X.; Wang, L.; Hu, Q.; and Guo, Y. 2025. Adaptive sparse memory networks for efficient and robust video object segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2): 3820–3833.

Dong, Y.; Kang, C.; Zhang, J.; Zhu, Z.; Wang, Y.; Yang, X.; Su, H.; Wei, X.; and Zhu, J. 2023. Benchmarking robustness of 3d object detection to common corruptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1022–1032.

Engel, N.; Belagiannis, V.; and Dietmayer, K. 2021. Point transformer. *IEEE access*, 9: 134826–134840.

Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X.; et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, volume 96, 226–231.

Gonizzi Barsanti, S.; Marini, M. R.; Malatesta, S. G.; and Rossi, A. 2024. Evaluation of denoising and voxelization algorithms on 3d point clouds. *Remote Sensing*, 16(14): 2632.

Goyal, A.; Law, H.; Liu, B.; Newell, A.; and Deng, J. 2021. Revisiting point cloud shape classification with a simple and effective baseline. In *International Conference on Machine Learning*, 3809–3820. PMLR.

Guo, M.-H.; Cai, J.-X.; Liu, Z.-N.; Mu, T.-J.; Martin, R. R.; and Hu, S.-M. 2021. Pct: Point cloud transformer. *Computational Visual Media*, 7: 187–199.

Hamdi, A.; Giancola, S.; and Ghanem, B. 2021. Mvtn: Multi-view transformation network for 3d shape recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1–11.

Hodgkin, A. L.; and Huxley, A. F. 1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4): 500.

Klokov, R.; and Lempitsky, V. 2017. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *Proceedings of the IEEE International Conference on Computer Vision*, 863–872.

Lei, Y.; Zhu, H.; Yuan, J.; Xiang, G.; Zhong, X.; and He, S. 2024. DenseTrack: Drone-based crowd tracking via density-aware motion-appearance synergy. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 2050–2058.

Li, J.; Chen, B. M.; and Lee, G. H. 2018. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9397–9406.

Li, X.; Li, C.; Rahaman, M. M.; Sun, H.; Li, X.; Wu, J.; Yao, Y.; and Grzegorzek, M. 2022. A comprehensive review of computer-aided whole-slide image analysis: from datasets to feature extraction, segmentation, classification and detection approaches. *Artificial Intelligence Review*, 55(6): 4809–4878.

Liu, W.; Deng, Y.; Chen, K.; Zhong, X.; Yu, Z.; and Huang, T. 2025. SOTA: Spike-navigated optimal transport saliency region detection in composite-bias videos. *arXiv preprint arXiv:2505.00394*.

Liu, X.; Han, Z.; Liu, Y.-S.; and Zwicker, M. 2019a. Point2sequence: Learning the shape representation of 3d point clouds with an attention-based sequence to sequence network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 8778–8785.

Liu, Y.; Fan, B.; Meng, G.; Lu, J.; Xiang, S.; and Pan, C. 2019b. Densepoint: Learning densely contextual representation for efficient point cloud processing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5239–5248.

Liu, Y.; Fan, B.; Xiang, S.; and Pan, C. 2019c. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8895–8904.

Liu, Z.; Hu, H.; Cao, Y.; Zhang, Z.; and Tong, X. 2020. A closer look at local aggregation operators in point cloud analysis. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, 326–342. Springer.

Ma, X.; Qin, C.; You, H.; Ran, H.; and Fu, Y. 2022. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. *arXiv preprint arXiv:2202.07123*.

Ma, Y.; Liu, H.; Pei, Y.; and Guo, Y. 2024. Heterogeneous graph learning for scene graph prediction in 3d point clouds. In *European Conference on Computer Vision*, 274–291. Springer.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30.

Qiu, S.; Anwar, S.; and Barnes, N. 2021a. Dense-resolution network for point cloud classification and segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 3813–3822.

Qiu, S.; Anwar, S.; and Barnes, N. 2021b. Geometric back-projection network for point cloud classification. *IEEE Transactions on Multimedia*, 24: 1943–1955.

Ren, D.; Ma, Z.; Chen, Y.; Peng, W.; Liu, X.; Zhang, Y.; and Guo, Y. 2024. Spiking pointnet: Spiking neural networks for point clouds. *Advances in Neural Information Processing Systems*, 36.

Ren, J.; Pan, L.; and Liu, Z. 2022. Benchmarking and analyzing point cloud classification under corruptions. In *International Conference on Machine Learning*, 18559–18575. PMLR.

Sun, J.; Zhang, Q.; Kailkhura, B.; Yu, Z.; Xiao, C.; and Mao, Z. M. 2022. Benchmarking robustness of 3d point cloud recognition against common corruptions. *arXiv preprint arXiv:2201.12296*.

Takaghaj, S. M.; and Sampson, J. 2024. Efficient spatio-temporal signal recognition on edge devices using PointLCA-Net. *arXiv preprint arXiv:2411.14585*.

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.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5): 1–12.

Wolff, K.; Kim, C.; Zimmer, H.; Schroers, C.; Botsch, M.; Sorkine-Hornung, O.; and Sorkine-Hornung, A. 2016. Point cloud noise and outlier removal for image-based 3D reconstruction. In *2016 Fourth International Conference on 3D Vision (3DV)*, 118–127. IEEE.

Wu, Q.; Zhang, Q.; Tan, C.; Zhou, Y.; and Sun, C. 2024. Point-to-spike residual learning for energy-efficient 3d point cloud Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 6092–6099.

Wu, S.; Huang, H.; Wang, S.; Chen, G.; Zhou, C.; and Yang, D. 2025. Neural heterogeneity enhances reliable neural information processing: Local sensitivity and globally input-slaved transient dynamics. *Science Advances*, 11(14): eadr3903.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1912–1920.

Xu, Y.; Fan, T.; Xu, M.; Zeng, L.; and Qiao, Y. 2018. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *Proceedings of the European Conference on Computer Vision*, 87–102.

Yan, X.; Zheng, C.; Li, Z.; Wang, S.; and Cui, S. 2020. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5589–5598.

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.

Yi, Z.; Lian, J.; Liu, Q.; Zhu, H.; Liang, D.; and Liu, J. 2023. Learning rules in spiking neural networks: A survey. *Neurocomputing*, 531: 163–179.

Zhang, T.; Yu, K.; Zhong, X.; Wang, H.; Xu, Q.; and Zhang, Q. 2025. STAA-SNN: Spatial-temporal attention aggregator for spiking neural networks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 13959–13969.

Zhao, Y.; Wu, Y.; Chen, C.; and Lim, A. 2020. On isometry robustness of deep 3d point cloud models under adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1201–1210.

Zheng, H.; Zhong, X.; Liu, B.; Xiao, Y.; Wen, B.; and Li, X. 2025. PAD: Phase-amplitude decoupling fusion for multi-modal land cover classification. *arXiv preprint arXiv:2504.19136*.

Zhong, Q.; and Han, X.-F. 2021. Point cloud learning with transformer. *arXiv preprint arXiv:2104.13636*.

Zhong, X.; Hu, S.; Liu, W.; Huang, W.; Ding, J.; Yu, Z.; and Huang, T. 2024. Towards low-latency event-based visual recognition with hybrid step-wise distillation spiking neural networks. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 9828–9836.

Zhou, Q.-Y.; Park, J.; and Koltun, V. 2018. Open3D: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*.