Noise-Injected Spiking Graph Convolution for Energy-Efficient 3D Point Cloud Denoising

Zikuan Li[1], **Qiaoyun Wu**[3*], **Jialin Zhang**[2], **Kaijun Zhang**[2], **Jun Wang**[1,2]

[1]School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics
[2]College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics
[3]School of Artificial Intelligence, Anhui University
lzkuan@nuaa.edu.cn, wuqiaoyun@ahu.edu.cn, wjun@nuaa.edu.cn

Abstract

Spiking neural networks (SNNs), inspired by the spiking computation paradigm of the biological neural systems, have exhibited superior energy efficiency in 2D classification tasks over traditional artificial neural networks (ANNs). However, the regression potential of SNNs has not been well explored, especially in 3D point cloud processing. In this paper, we propose noise-injected spiking graph convolutional networks to leverage the full regression potential of SNNs in 3D point cloud denoising. Specifically, we first emulate the noise-injected neuronal dynamics to build noise-injected spiking neurons. On this basis, we design noise-injected spiking graph convolution for promoting disturbance-aware spiking representation learning on 3D points. Starting from the spiking graph convolution, we build two SNN-based denoising networks. One is a purely spiking graph convolutional network, which achieves low accuracy loss compared with some ANN-based alternatives, while resulting in significantly reduced energy consumption on two benchmark datasets, PU-Net and PC-Net. The other is a hybrid architecture that combines ANN-based learning with a high performance-efficiency trade-off in just a few time steps. Our work lights up SNN's potential for 3D point cloud denoising, injecting new perspectives of exploring the deployment on neuromorphic chips while paving the way for developing energy-efficient 3D data acquisition devices.

Code — https://github.com/Miraclelzk/NI-SGCN

Introduction

Intelligent robotic systems are often equipped with a 3D data acquisition device, which provides detailed point clouds for environment perception. However, raw point clouds are often corrupted by noise, stemming from sensor imperfections, environmental interference, and the probabilistic nature of data capture. Point cloud denoising endeavors to extricate coherent geometrical information from raw scans, thereby enhancing the precision of the data and fortifying its applicability in advanced perception tasks. With the continuous advancement of deep learning, learning-based point cloud denoising methods have achieved remarkable progress (Rakotosaona et al. 2020; Zhang et al. 2020; Luo and Hu 2021). However, these ANN-based methods typically demand numerous multiply-accumulate operations, leading to high computational costs and energy consumption. This poses significant challenges when deploying such methods on 3D scanning devices, which have resources and battery constraints.

Spiking Neural Networks (SNNs), using spike-driven communication, have emerged as a prominent solution to mitigate the exiguity of energy efficiency inherent in contemporary deep learning. In SNNs, all information is encoded within spiking signals rather than floating-point representations, which allows SNNs to adopt spike-based accumulate (AC) operations instead of energy-hungry multiply-accumulate (MAC) operations, thus leading to extremely low energy consumption (Kim et al. 2020). The expeditious progression in the domain of neuromorphic engineering has culminated in the advent of sophisticated chips like Loihi (Davies et al. 2018), which significantly augment the energy efficiency inherent to SNNs. This advancement brings us to the future of integrating intelligent neuromorphic processors into the fabric of quotidian existence.

Despite SNNs have been successfully applied in the field of neuromorphic computing (Cao et al. 2024), the current body of research on SNNs has been predominantly directed towards classification endeavors, inadvertently neglecting the exploration of their regression competencies, notably within the domain of 3D point cloud denoising. Hence, to develop a SNN-based regression algorithm supporting effective point cloud denoising whilst maintaining a profile of low energy expenditure, is imperative. However, two key concerns deserve much attention in extending SNNs to the domain: (1) stochastic noise will bring spiking disturbance to deterministic models, directly undermining their robustness, and (2) semantically similar structures within a point cloud can facilitate the mutual perception of perturbations, thereby enhancing the discriminative capacity of the system to detect and respond to noises in the data.

This paper explores SNNs for energy-efficient 3D point cloud denoising. For the first concern, we draw inspiration from the inherent nondeterministic and noisy nature of neural computations, and build noise-injected spiking neurons to yield flexible and and reliable learning on 3D points. For the second concern, we borrow concepts from EdgeConv (Wang et al. 2019), especially learning to capture

*The corresponding authors.

semantically similar structures by dynamically updating a graph of point relationships. We integrate our spiking neurons into the convolution and design the noise-injected spiking graph convolution, which increases the representation power for discerning disturbances while efficiently propagating information via sparse spiking signals. On this basis, we propose two variants of noise-injected spiking graph convolutional networks (NI-SGCN) for the denoising of 3D point cloud data. The first model is articulated in its entirety within the spiking paradigm, denoted as NI-PSGCN, exemplifying the Purest form of spiking computation. The second model, NI-HSGCN, represents a Hybrid construct that integrates select learning operations from artificial neural networks (ANNs), thereby harnessing the complementary strengths of both SNNs and ANNs within a cohesive architecture. The main contributions are as follows:

- We define the noise-injected spiking neuron for nondeterministic spiking learning on 3D points. We show the neuron leads to SNNs with competitive performance and improved robustness when facing challenging disturbances compared with deterministic spiking neurons.
- We design the noise-injected spiking graph convolution, capable of exploiting semantically similar structures to facilitate spiking representation learning while also enhancing the information flow efficiency.
- We develop two noise-injected spiking graph convolutional networks for 3D point cloud denoising, which significantly reduces energy consumption. To the best of our knowledge, our work is the first to employ spiking neural networks for energy-efficient point cloud denoising, while maintaining high accuracy on the PU-Net dataset (Yu et al. 2018) and PC-Net dataset (Rakotosaona et al. 2020).

Related Work

Denoise on Point Clouds

Most learning-based point cloud denoising methods evolve from foundational point cloud processing techniques (Preiner et al. 2014). The development of PointNet and PointNet++ (Qi et al. 2017) enables the direct convolution of point sets, paving the way for more advanced approaches. Building on these advancements, Wang $et\ al.$ (Wang et al. 2019) introduce a graph convolutional architecture that uses nearest-neighbor graphs derived from point sets to generate rich feature representations.

PointCleanNet (PCN) (Rakotosaona et al. 2020) employs a two-level network to first remove outlier points and then learn the motion coordinates of the noisy point cloud, transforming it into a cleaner version. Pointfilter (Zhang et al. 2020) uses clean normals as a supervisory signal to analyze the model's latent surface information, effectively removing noise while preserving the sharp edges of the point cloud.

Luo $et\ al.$ introduce ScoreDenoise, a score-based denoising method that models the gradient log of the noise-convolved probability distribution for point cloud patches (Luo and Hu 2021). Chen $et\ al.$(Chen et al. 2019) propose a multi-block denoising approach based on low-rank matrix recovery with graph constraints and later developed RePCD (Chen et al. 2022), a feature-aware recurrent network. Edirimuni $et\ al.$ (de Silva Edirimuni et al. 2023) criticize RePCD for its lack of iterative noise reduction during testing and propose IterativePFN, an iterative point cloud filtering network that explicitly models the iterative filtering process internally. Wei $et\ al.$(Wei et al. 2024) propose PathNet, a path-selective point cloud denoising framework that adapts its approach based on varying levels of noise and the distinct geometric structures of the points.

Spiking Neural Networks

SNNs are regarded as the third generation of neural networks, inspired by brain-like computing processes that use event-driven signals to update neuronal nodes (Cao et al. 2024). Unlike conventional ANNs, spiking neurons operate on discrete-time events rather than continuous values, making SNNs more energy and memory-efficient on embedded platforms (Wu et al. 2019).

One significant challenge with SNNs is the effective training and optimization of network parameters. Currently, there are two primary methods for developing deep SNN models: ANN-to-SNN conversion and direct training. In ANN-to-SNN conversion, ReLU activation layers are replaced with spiking neurons to replicate the behavior of the original ANN. However, these converte SNNs often require substantial inference time and memory, resulting in increased latency and decreased energy efficiency, which undermines the advantages of spiking models (Roy, Jaiswal, and Panda 2019). In contrast, direct training involves designing surrogate gradients for backpropagation or using gradients with respect to membrane potentials to train SNNs from scratch. Models trained directly tend to reduce spiking time latency and are more suitable for practical applications. However, for large-scale tasks, they often do not match the accuracy of conversion-based approaches or ANNs (Roy, Jaiswal, and Panda 2019).

To enhance SNN performance and bridge the gap between ANNs and SNNs, several advancements have been made. Wu $et\ al.$ (Wu et al. 2019) introduce neuron normalization to balance firing rates and preserve important information. The QIF neuron (Brunel and Latham 2003) simulates neuronal electrical activity by extending the standard Integrate-and-Fire (IF) neuron with a quadratic nonlinearity, offering a more accurate representation of the neuron's membrane potential. The KLIF neuron (Jiang and Zhang 2023) is a novel k-based leaky integrate-and-fire (LIF) neuron designed to enhance the learning capabilities of spiking neural networks. Spiking neurons with noise-injected dynamics are considered more biologically realistic. Rao $et\ al.$ (Rao 2004) develope small noise-spiking neural networks to perform probabilistic reasoning, effectively improving network robustness. However, integrating these methods into arbitrary network architectures remains challenging.

Spiking Neural Networks on Point Cloud

Recent research efforts are exploring the application of SNNs in point cloud processing. Lan $et\ al.$ (Lan et al. 2023) propose an efficient unified ANN-SNN conversion method

for point cloud and image classification, significantly reducing time steps for a fast and lossless transformation. Ren *et al.* (Ren et al. 2024) extend PointNet to SNNs and develop Spiking PointNet, while Wu *et al.* (Wu et al. 2024) introduce a point-to-spike residual learning network for point cloud classification. Despite these advances, there are relatively few studies combining SNNs with point cloud data, and most focus on classification tasks. To our knowledge, we are the first to apply SNNs to point cloud denoising.

Methods

We propose noise-injected spiking graph convolutional networks for 3D point cloud denoising striking a good balance between effectiveness and efficiency. We first define noise-injected spiking neurons, which take advantage of non-deterministic, noisy neurodynamic computations derived from the brain to enhance computational robustness. We then integrate noise-injected spiking neurons into traditional graph convolution (Wang et al. 2019) to create noise-injected spiking graph convolution, enhancing 3D point feature learning from spike sequences. Finally, we build 3D spiking denoising networks on the basis of the proposed spiking graph convolution.

Noise-Injected Spiking Neurons

Inspired by the inherent nondeterministic characteristics of neural computations (Ma, Yan, and Tang 2023), we define the noise-injected spiking neuron for spike computation.

Integrate-and-Fire spiking neurons. Bio-inspired spiking neurons are designed to mimic the actual signal processing in the brain. In this paper, we employ the simplest model of spiking neurons, the Integrate-and-Fire model, which is defined as:

$$U_t = V_{t-1} + I_t \tag{1}$$

$$S_t = \Theta(U_t - V_{th}) \tag{2}$$

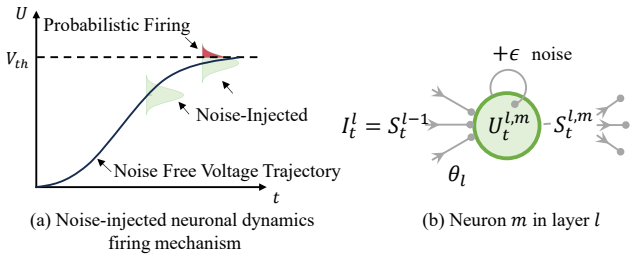
$$V_t = U_t(1 - S_t) + V_{reset} S_t \tag{3}$$

Where U_t denotes the membrane potential at time step t. S_t is the spike output, which occurs when U_t exceeds a threshold V_{th}. I_t is the input current at time step t. $\Theta(\cdot)$ is the Heaviside step function, and V_t is the membrane potential after a spike is triggered. We employ a "hard reset" method (Fang et al. 2021) in Eq. (3), meaning that after a spike ($S_t = 1$), the membrane potential V_t resets to $V_{reset} = 0$.

Noise-injected Integrate-and-Fire spiking neurons. To improve the robustness against challenging disturbances, we inject noise into the IF spiking neuron. Specifically, we add a Gaussian noise term to Eq. (1). The dynamics of the noise-injected IF spiking neuron (NIIF) are defined as:

$$U_t = V_{t-1} + I_t + \epsilon \tag{4}$$

Here, the noise term ϵ is drawn from a Gaussian distribution, $\epsilon \sim N(\mu, \sigma^2)$, where μ is the mean and σ is the standard deviation of the distribution. Additionally, the firing and resetting dynamics of noisy spiking neurons remain unchanged, namely Eq. (2)(3).



(a) Noise-injected neuronal dynamics firing mechanism

(b) Neuron m in layer l

Figure 1: The probabilistic firing mechanism and calculation process of the NIIF neuron.

Fig. 1 illustrates the probabilistic firing mechanism and calculation process of the NIIF neuron. The firing probability is represented by the membrane noise cumulative distribution function, depicted by the shaded red area under the noisy voltage distribution. At the time step t, the membrane potential $U_t^{l,m}$ and the spike output $S_t^{l,m}$ of the m-th neuron in the l-th layer, become random variables due to the injected noise (Ma, Yan, and Tang 2023). The computation for updating synaptic weights denotes as θ_l. With the noise, we obtain the firing probability distribution of NIIF based on the threshold firing mechanism:

$$P[S_t = 1] = F_\epsilon(U_t - V_{th}), \tag{5}$$

where S_t is the spike state, and F_ϵ is the cumulative distribution function of the noise. The difference $U_t - V_{th}$ governs the firing probability. Specifically, it relates to previous literature on escape noise models (Jolivet et al. 2006).

Noise-Injected Spiking Graph Convolution

Inspired by the discriminative representation capability of the dynamic graph convolution from EdgeConv (Wang et al. 2019), we combine the noise-injected spiking neuron with the graph convolution to promote disturbance-aware spiking representation learning on 3D points.

Graph convolution. Graph convolution (Wang et al. 2019) is defined to learn the local geometric structures by iteratively computing edge features between points. For a point cloud $X = \{x_1, x_2, \ldots, x_n\}$, where each $x_i \in \mathbb{R}^3$, we first construct a graph $G = (X, E)$, where X is the vertex set and the edge set is constucted based on the k-nearest neighbor relationship.

At the l-th graph convolutional layer, the edge feature e_{ij}^l between a feature f_i and its neighbor feature f_j is defined as:

$$e_{ij}^l = h_\theta(f_i^{l-1}, f_j^{l-1} - f_i^{l-1}) \tag{6}$$

Here, h_θ is a parametric function, typically modeled as a fully connected layer, which processes the concatenation of the feature f_i^{l-1} from x_i and the relative feature displacement $f_j^{l-1} - f_i^{l-1}$. This helps capture local neighborhood information and directional relationships between points. The edge features are then aggregated for each feature f_i using a symmetric function such as max pooling:

$$f_i^l = \max_{j \in \mathcal{N}(i)} e_{ij}^l \tag{7}$$

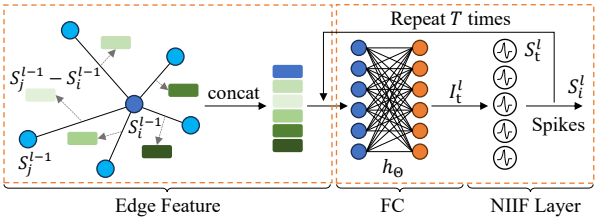


Figure 2: An illustration of noise-injected spiking graph convolution.

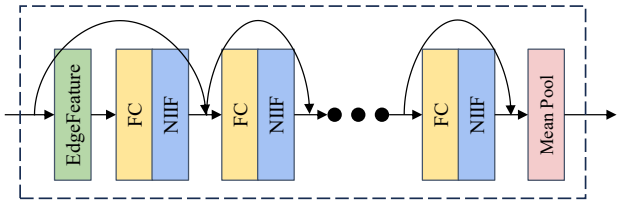


Figure 3: An illustration of dense NI-SGC block.

where $\mathcal{N}(i)$ denotes the neighbors of feature f_i. The aggregation is invariant to the order of neighbors and extracts robust local features. By stacking multiple graph convolution layers, the network refines point features iteratively to capture increasingly complex geometric patterns and spatial hierarchies within the point cloud.

Noise-injected spiking graph convolution (NI-SGC). To leverage graph convolution for capturing local and global geometric features in point clouds while enhancing the dynamics and expressiveness of the network with spiking neurons, we combine the NIIF neuron with the graph convolution to design the noise-injected spiking graph convolution as:

$$\begin{aligned} I_t^l(x_i) &= \{h_\theta(S_t^{l-1}(x_i), S_t^{l-1}(x_j) - S_t^{l-1}(x_i))\}_{j\in\mathcal{N}(i)}, \\ U_t^l(x_i) &= V_{t-1}^l(x_i) + I_t^l(x_i) + \epsilon, \\ S_t^l(x_i) &= \Theta\left(U_t^l(x_i) - V_{th}\right), \\ V_t^l(x_i) &= U_t^l(x_i)\left(1 - S_t^l(x_i)\right) + V_{\text{reset}}\, S_t^l(x_i), \end{aligned} \tag{8}$$

Where $I_t^l(x)$ is the input current inspired by Eq. (6) and (7). The differences lie in two aspects. The first is the spiking feature inputs from the $(l-1)$-th layer. Consequently, the computation of $I_t^l(x)$ can be simplified to AC operations, with weight accumulation occurring only when neighboring neurons generate spikes in the $(l-1)$-th layer. $U_t^l(x)$ denotes the membrane potential of the neuron at 3D position x in the l-th layer at time step t. V_{t-1}^l represents the membrane potential at time $(t-1)$. The Heaviside step function $\Theta(\cdot)$ is used for spiking determination. Fig. 2 illustrates the dynamics of the proposed spiking graph convolution neuron. At the l-th layer, the neuron receives spike inputs from the neighborhood $\mathcal{N}_i$ of the $(l-1)$-th layer. NI-SGC then constructs the edge feature for the spikes S^{l-1} from the $(l-1)$-th layer, followed by processing through FC and NIIF Layers.

Dense NI-SGC block. The spiking graph convolution (NI-SGC) can efficiently extract multi-scale and non-local features for each point, while dense connections can provide richer contextual information, which is much suitable for denoising tasks (Liu et al. 2019). Hence, we design dense NI-SGC block for point cloud denoising by elaborately combining NI-SGC, FC, and NIIF. The r-th NI-SGC block can be formulated as:

$$S_t^{r+1}(x_i) = \text{mean}_{j\in\mathcal{N}(i)}\, H_\theta(S_t^r(x_i), S_t^r(x_j) - S_t^r(x_i)) \tag{9}$$

where $S_t^r = \{S_t^r(x_i)\}_{i=1}^N$ are spiking feature representations in a high-dimensional space, serving as the input to the r-th block. H_θ represents a densely connected FC parameterized by θ, $\mathcal{N}(i)$ denotes the neighborhood of spiking feature $S_t(x_i)$. The other is that we replace the max pooling with the average pooling. As illustrated in Fig. 3, dense connections are employed both within and between spiking graph convolution layers. In each graph convolution layer, H_θ is densely connected, and features are passed to all subsequent layers. These dense connections reduce network parameters and enhance contextual information(Liu et al. 2019).

Noise-Injected Spiking Graph Convolutional Networks

In this section, we design noise-injected spiking graph convolutional networks (NI-SGCN) for 3D point cloud denoising, which is based on an ANN-based denoising architecture, ScoreDenoise (Luo and Hu 2021). The NI-SGCN consists of two main parts: the spiking feature extraction module and the score estimation module.

The spiking feature extraction module aims to learn pointwise spiking features from the input noisy 3D point set $X = \{x_i\}_{i=1}^N \in R^{T*N*3}$, where T represents the spiking time latency, N denotes the number of points, and 3 represents the 3D coordinates of points. The structures of our spiking feature extraction module are shown in Fig. 4, which is constructed by stacking dense NI-SGC blocks and fully connected layers. The learned spiking feature for point x_i is denoted as h_i.

The score estimation module is parameterized by the feature h_i of point x_i, which outputs a score $Sc(x_i)$. The $Sc(x_i)$ represents the gradient from x_i to the underlying surface and is used to determine the direction of optimizing noise. There are two variants: one is the ANN-based score estimation, following ScoreDenoise (Luo and Hu 2021) and the other is a SNN-based score estimation by simply replacing the ReLU in the ANN-based score estimation with NIIF spiking neurons. Consequently, we propose two implementations of NI-SGCN: the first is a hybrid architecture, NI-HSGCN, witch employs the above spiking feature extraction module and the ANN-based score estimation module; the second is NI-PSGCN, a purely SNN-based structure, which fully leverages the enhanced energy efficiency of SNNs.

The final training objective aggregates the objectives for

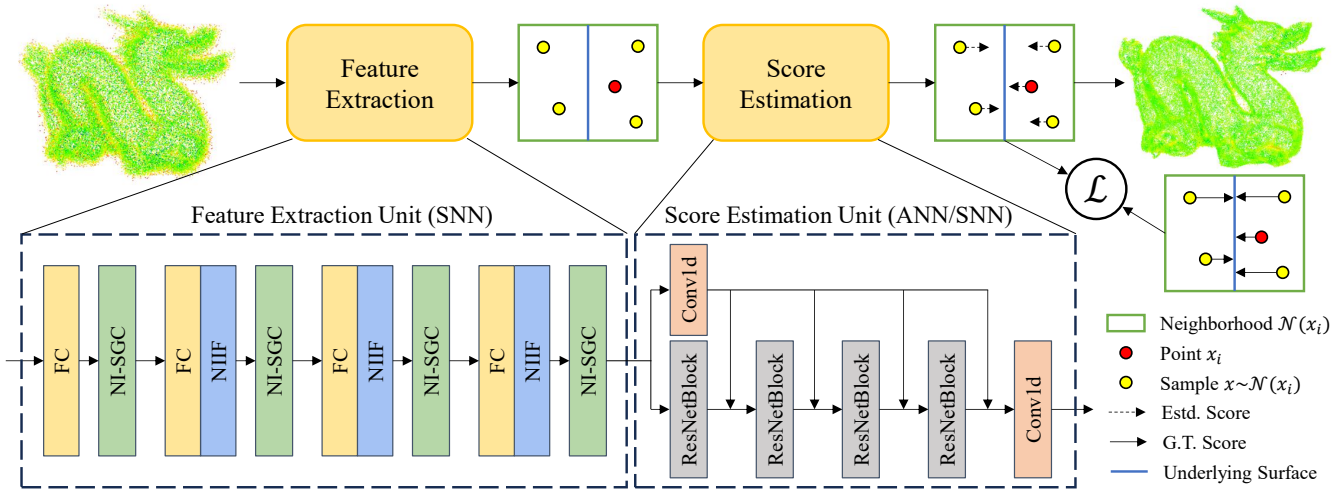


Figure 4: Illustration of the architecture and pipeline of the noise-injected spiking point cloud denoise learning network.

each local score function:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{j\sim\mathcal{N}(i)} \left[\|sc(x_j) - Sc(x_i)\|_2^2 \right] \tag{10}$$

where $\mathcal{N}(i)$ is a distribution concentrated in the neighborhood of x_i. $sc(x_j)$ is the ground truth score. Note that, this objective not only matches the predicted score on the position of x_i but also matches the score on the neighboring areas of x_i. For NIIF neuron, we only add noise during training to improve robustness. The derivative of the Heaviside step function in Eq. (8) equals the Dirac delta function, which makes the training process unstable if used directly for gradient descent. Following (Fang et al. 2023), we use the surrogate function $\Theta'(v) \triangleq \sigma'(v)$ in gradient backpropagation, where $\sigma(v)$ is a smooth, continuous function resembling $\Theta(v)$, specifically the arctan function.

Experiments and Results

Experimental Settings

We evaluate the performance of 3D point cloud denoising using two benchmark datasets, PU-Net (Yu et al. 2018) and PC-Net (Rakotosaona et al. 2020). PU-Net contains 60 distinct shapes, which are divided into 40 for training and 20 for testing. The PC-Net dataset, used exclusively for generalization testing, comprises 10 unique point cloud shapes and their corresponding meshes.

Following the experimental setup in the literature (Luo and Hu 2021), the meshes in both datasets are normalized to a unit sphere. We use Poisson disk sampling to generate point cloud data at a resolution of 10K and 50K points. Then we add Gaussian noise with a standard deviation of 0.5% to 2% of the unit sphere. The model is trained on the PU-Net training set, and its denoising performance is evaluated on both the PU-Net test set and the PC-Net dataset. In line with previous work, Chamfer Distance (CD) and Point-to-Mesh Distance (P2M) are used as evaluation metrics, with both CD and P2M reported in units of 10^{-4}.

module	CD	P2M
NI-HSGCN	**1.798**	**1.147**
NI-PSGCN	1.926	1.211

Table 1: Ablation study on the NI-SGCN architecture designs on the PU-Net dateset.

All experiments are implemented on Intel an i9-13900HX CPU and an NVIDIA RTX 4090 GPU (24GB memory, CUDA 11.8), using PyTorch and SpikingJelly (Fang et al. 2023) for implementation. NI-SGCN is trained with the Adam optimizer using a learning rate of 1×10^{-4}, and the network is trained with a batch size of 32. For all our SNN models, the default time delay T is set to 4, the membrane potential threshold is set to 1, and the standard deviation of injected noise is 0.2.

Ablation Study

We first perform ablation experiments on PU-Net to establish the final architecture of NI-SGCN. All ablation study are performed on the PU-Net dataset, using point clouds with 50K points and 2% Gaussian noise.

Ablation on NI-SGCN architecture designs. Starting from the spiking graph convolution, we build two SNN-based denoising networks. One is a pure spike graph convolutional network structure NI-PSGCN, and the other is a hybrid architecture NI-HSGCN, which integrates some ANN-based learning operations. We use PU-Net to perform denoising ablation experiments on both architectures, with the evaluation results reported in Table 1. The hybrid NI-HSGCN outperforms the purely spiking NI-PSGCN, with a reduction of 0.128 in CD and a reduction of 0.064 in P2M metrics. While NI-PSGCN has achieved good denoising results using a pure SNN, its regression module (score estimation) cannot match the structural performance of an ANN. However, by adopting a hybrid structure, NI-PSGCN lever-

Decoding Scheme	CD	P2M
MAX	2.199	1.495
MEAN	**1.798**	**1.147**

Table 2: Ablation study on the pooling schemes on the PU-Net dataset.

neurons	IF	LIF	NIIF	NILIF	QIF	KLIF
CD	2.057	1.889	**1.798**	1.848	2.176	2.469
P2M	1.362	1.203	**1.147**	1.186	1.465	1.711

Table 3: Ablation study on the usage of different spiking neurons in the network.

ages the complementary strengths of both SNN and ANN, balancing energy efficiency with denoising effectiveness. Therefore, we choose the hybrid NI-HSGCN architecture to build the final denoising network in subsequent experiments.

Ablation on pooling schemes. We perform a series of ablation studies comparing max pooling and mean pooling, with the results shown in Table 2. The findings show that the NI-SGCN network achieves superior denoising performance with mean pooling. Specifically, mean pooling reduces the CD by 0.401 and the P2M by 0.348 compared to max pooling. In both metrics, mean pooling consistently outperforms max pooling. This improvement is attributable to the binary outputs (0 or 1) produced by spiking neurons in SNNs, in contrast to the float values generated by activation functions in ANNs. Max pooling in SNNs captures only the most prominent binary features, potentially overlooking finer details. In contrast, mean pooling converts binary features into float values, preserving more information about the underlying geometry. Consequently, mean pooling offers a more representative and stable aggregation of local geometric features.

Ablation on spiking neurons. We assess the impact of various types of spiking neurons on the performance of NI-SGCN. Specifically, we compare the network's performance using NIIF, Noise-injected Leaky Integrate-and-Fire (NILIF), IF, LIF, QIF (Brunel and Latham 2003) and KLIF (Jiang and Zhang 2023) neurons. As shown in Table 3, NIIF neuron achieve the best performance in both CD and P2M metrics. Noise-injected IF neurons can directly incorporate noise into the integration process, thereby enhancing the robustness and generalization ability of the network. Spiking neurons with noise-perturbed dynamics are believed to be more biologically realistic, and internal noise brings potential benefits by promoting more generalization performance. Furthermore, the simplicity of the IF neuron, when combined with noise injection, allows for more efficient and effective spike-based computation compared to the LIF neurons.

Ablation on time latency. In spiking neural networks, the time delay T is a critical hyperparameter. As illustrated in Table 4, our evaluation indicates that the network sustains strong performance metrics with minimal variation in denoising effectiveness when T ranges from 2 to 8. Notably, the optimal denoising performance, characterized by the lowest CD and P2M metrics, occurs at $T = 4$. While increasing T from 2 to 4 improves denoising performance, further increasing T to 8 results in a decline, likely because excessively large T values introduce redundancy rather than providing valuable information.

Latency	1	2	4	8
CD	5.524	1.951	**1.798**	1.815
P2M	4.419	1.273	**1.147**	1.155

Table 4: Ablation study on the time latency on the PU-Net dataset. NI-SGCN with $T = 4$ presents the highest CD and P2M metrics.

Comparison with State-of-the-art Methods

We compare our method against state-of-the-art deep learning-based denoisers, including PCN (Rakotosaona et al. 2020), ScoreDenoise (Luo and Hu 2021), DMRDenoise (Luo and Hu 2020), and Pointfilter (Zhang et al. 2020). Our evaluations are conducted on the PU-Net and PC-Net datasets, with isotropic Gaussian noise applied at a standard deviation ranging from 1% to 2% of the unit sphere radius.

As shown in Table 5, the proposed NI-HSGCN and NI-PSGCN maintain low accuracy loss compared with the ANN-based ScoreDenoise and Pointfilter. However, our work has the advantage of ultra-low energy consumption as demonstrated in the next section. These results indicate the capability of NI-HSGCN and NI-PSGCN in preserving the point cloud structure while effectively reducing noise. Our work can provide a proper balance between denoising accuracy and efficiency. On the other hand, our networks outperform PCN and DMRDenoise in most cases. The experiments across different datasets and noise levels further highlight the robustness and adaptability of our networks.

Figure 5 visually compares the denoising results achieved by our proposed method with those of competitive baselines, under 2% Gaussian noise and a point cloud resolution of $50K$ points. Each point is color-coded according to its denoising error, as measured by the point-to-grid distance, with green indicating points closer to the underlying surface and red indicating points farther away. The visualization, consistent with Table 5, clearly demonstrates that our NI-HSGCN produces cleaner and more visually appealing outcomes compared to PCN and DMRDenoise. Our work can reduce noise and preserve fine details to some extents.

Theoretical Energy Consumption Calculation

We examine the hardware efficiency of the proposed framework. In contrast, our SNN architecture leverages a transformation that largely bypasses multiplication, retaining it only in the initial layer. This design allows the hardware to take advantage of sparse computation, effectively eliminating addition operations in the absence of spikes. We estimate the primary energy consumption of our network and the ScoreDenoise (Luo and Hu 2021) network, while excluding point cloud downsampling and normalization operations. Both

Points			10K (Sparse)				50K (Dense)			
Noise			1%		2%		1%		2%	
Dataset	Model		CD	P2M	CD	P2M	CD	P2M	CD	P2M
PU	ANN	PCN	3.686	1.599	7.926	4.759	1.103	0.646	1.978	1.370
		ScoreDenoise	**2.611**	**0.863**	**3.684**	**1.416**	0.767	0.448	1.295	0.842
		DMRDenoise	4.712	2.196	5.085	2.523	1.205	0.762	1.443	0.970
		Pointfilter	2.709	0.884	4.508	1.937	**0.723**	**0.389**	**1.175**	**0.709**
	SNN	NI-HSGCN	2.797	0.923	4.437	1.884	0.843	0.461	1.798	1.147
		NI-PSGCN	2.947	1.004	4.622	1.976	0.885	0.485	1.926	1.211
PC	ANN	PCN	3.847	**1.221**	8.752	3.043	1.293	**0.289**	1.913	0.505
		ScoreDenoise	**3.264**	1.663	**5.066**	2.485	1.075	0.543	1.671	1.006
		DMRDenoise	6.602	2.152	7.145	**2.237**	1.566	0.350	2.009	**0.485**
		Pointfilter	3.374	1.945	6.160	3.480	**1.060**	0.522	**1.620**	0.982
	SNN	NI-HSGCN	3.478	1.799	5.756	3.073	1.146	0.571	2.134	1.357
		NI-PSGCN	3.798	1.972	6.035	3.380	1.208	0.626	2.258	1.441

Table 5: Comparison among competitive denoising algorithms. The units of CD and P2M are both 10^{-4}.

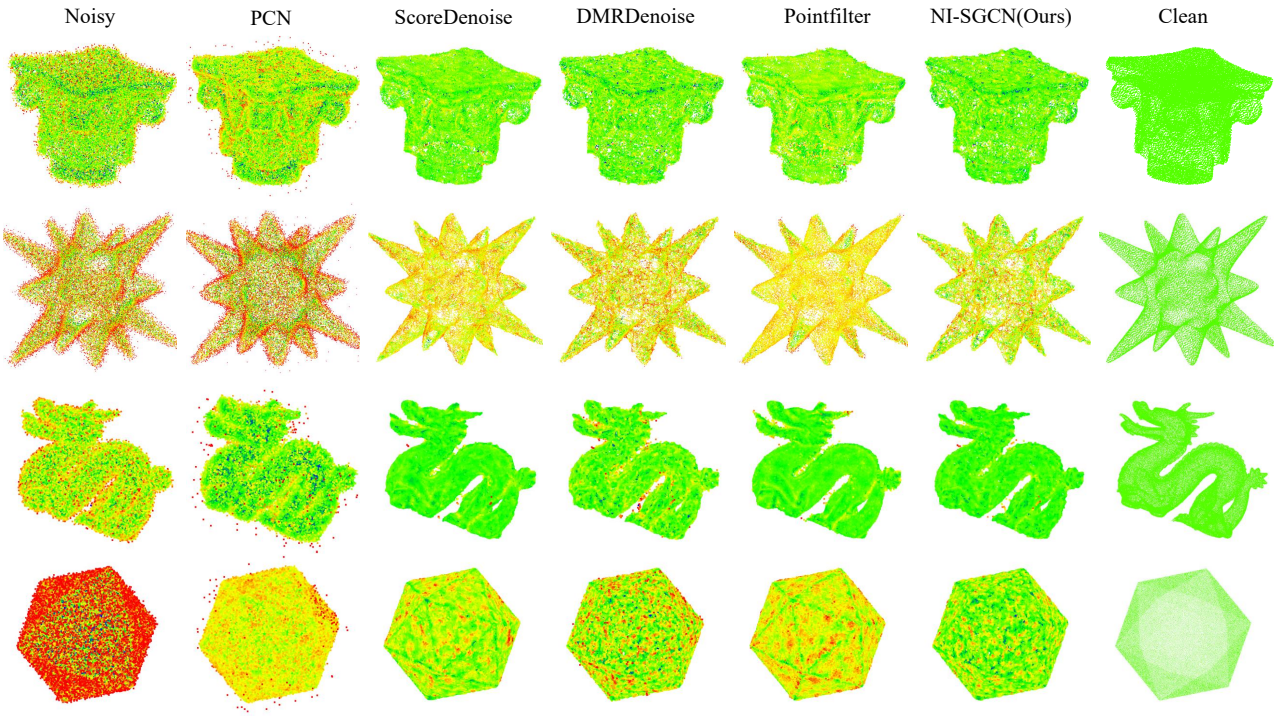


Figure 5: Visual comparison of denoising methods. Points colored reder are farther away from the ground truth surface.

networks are tested with an input of $5,000$ 3D points. Following (Horowitz 2014), by quantifying the computational workload in terms of operations executed, the conventional ANN-based model theoretically consumes $3.01 \times 10^9 pJ$ for per forward pass, In contrast, the SNN-based architecture NI-PSGCN requires only $2.36 \times 10^8 pJ$, representing a 12.75-fold reduction in energy consumption. The hybrid architecture NI-HSGCN consumes $7.65 \times 10^8 pJ$, which is approximately 3.94 times more energy-efficient than the traditional ANN. Due to the sparsity of spikes and the use of alternating current, our network demonstrates exceptional energy efficiency. Our work underscores the potential of SNNs for 3D point cloud denoising and supports the development of energy-efficient 3D data acquisition devices.

Conclusion

In this paper, we propose noise-injected spiking graph convolutional networks for 3D point cloud denoising, achieving an optimal balance between denoising effectiveness and bio-inspired energy efficiency on two benchmark datasets, PU-Net and PC-Net. In future work, we aim to explore more efficient and energy-saving 3D point cloud denoising networks while maintaining the high accuracy demonstrated by state-of-the-art methods.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.52275493, No.92367301, No.92267201, No.92160301, No.52425506, No.62206001).

References

Brunel, N.; and Latham, P. E. 2003. Firing rate of the noisy quadratic integrate-and-fire neuron. *Neural computation*, 15(10): 2281–2306.

Cao, J.; Wang, Z.; Guo, H.; Cheng, H.; Zhang, Q.; and Xu, R. 2024. Spiking denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 4912–4921.

Chen, H.; Wei, M.; Sun, Y.; Xie, X.; and Wang, J. 2019. Multi-patch collaborative point cloud denoising via low-rank recovery with graph constraint. *IEEE transactions on visualization and computer graphics*, 26(11): 3255–3270.

Chen, H.; Wei, Z.; Li, X.; Xu, Y.; Wei, M.; and Wang, J. 2022. Repcd-net: Feature-aware recurrent point cloud denoising network. *International Journal of Computer Vision*, 130(3): 615–629.

Davies, M.; Srinivasa, N.; Lin, T.-H.; Chinya, G.; Cao, Y.; Choday, S. H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. 2018. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro*, 38(1): 82–99.

de Silva Edirimuni, D.; Lu, X.; Shao, Z.; Li, G.; Robles-Kelly, A.; and He, Y. 2023. IterativePFN: True iterative point cloud filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13530–13539.

Fang, W.; Chen, Y.; Ding, J.; Yu, Z.; Masquelier, T.; Chen, D.; Huang, L.; Zhou, H.; Li, G.; and Tian, Y. 2023. Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence. *Science Advances*, 9(40): eadi1480.

Fang, W.; Yu, Z.; Chen, Y.; Masquelier, T.; Huang, T.; and Tian, Y. 2021. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2661–2671.

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.

Jiang, C.; and Zhang, Y. 2023. Klif: An optimized spiking neuron unit for tuning surrogate gradient slope and membrane potential. *arXiv preprint arXiv:2302.09238*.

Jolivet, R.; Rauch, A.; Lüscher, H.-R.; and Gerstner, W. 2006. Predicting spike timing of neocortical pyramidal neurons by simple threshold models. *Journal of computational neuroscience*, 21: 35–49.

Kim, S.; Park, S.; Na, B.; and Yoon, S. 2020. Spiking-yolo: spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 11270–11277.

Lan, Y.; Zhang, Y.; Ma, X.; Qu, Y.; and Fu, Y. 2023. Efficient converted spiking neural network for 3d and 2d classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9211–9220.

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

Luo, S.; and Hu, W. 2020. Differentiable manifold reconstruction for point cloud denoising. In *Proceedings of the 28th ACM international conference on multimedia*, 1330–1338.

Luo, S.; and Hu, W. 2021. Score-based point cloud denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4583–4592.

Ma, G.; Yan, R.; and Tang, H. 2023. Exploiting noise as a resource for computation and learning in spiking neural networks. *Patterns*, 4(10).

Preiner, R.; Mattausch, O.; Arikan, M.; Pajarola, R.; and Wimmer, M. 2014. Continuous projection for fast L1 reconstruction. *ACM Trans. Graph.*, 33(4): 47–1.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Rakotosaona, M.-J.; La Barbera, V.; Guerrero, P.; Mitra, N. J.; and Ovsjanikov, M. 2020. Pointcleannet: Learning to denoise and remove outliers from dense point clouds. In *Computer graphics forum*, volume 39, 185–203. Wiley Online Library.

Rao, R. P. 2004. Bayesian computation in recurrent neural circuits. *Neural computation*, 16(1): 1–38.

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.

Roy, K.; Jaiswal, A.; and Panda, P. 2019. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784): 607–617.

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.

Wei, Z.; Chen, H.; Nan, L.; Wang, J.; Qin, J.; and Wei, M. 2024. PathNet: Path-selective point cloud denoising. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

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, Y.; Deng, L.; Li, G.; Zhu, J.; Xie, Y.; and Shi, L. 2019. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 1311–1318.

Yu, L.; Li, X.; Fu, C.-W.; Cohen-Or, D.; and Heng, P.-A. 2018. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2790–2799.

Zhang, D.; Lu, X.; Qin, H.; and He, Y. 2020. Pointfilter: Point cloud filtering via encoder-decoder modeling. *IEEE Transactions on Visualization and Computer Graphics*, 27(3): 2015–2027.