Routing on Demand: DSNet for Efficient Progressive Point Cloud Denoising

Xiaoqian Cheng[1] Dong Xiao[1] Husen Li[1] Zheng Liu[2] Renjie Chen[1*]
[1]University of Science and Technology of China
[2]China University of Geosciences (Wuhan)
cxq_61@mail.ustc.edu.cn xiaodong@ustc.edu.cn wawalker@mail.ustc.edu.cn
liu.zheng.jojo@gmail.com renjiec@ustc.edu.cn

Abstract

Point cloud denoising is a critical preprocessing step for enhancing the reliability and accuracy of 3D perception systems. Most existing progressive denoising methods rely on fixed iterative pipelines that process all regions uniformly, resulting in redundant computation and over-smoothing of geometric details when handling point clouds with non-uniform noise distributions. To overcome these limitations, we introduce Dynamic Skip Net (DSNet), a novel progressive denoising framework that enables routing on demand by adaptively determining the optimal denoising path for each local patch based on its noise characteristics. DSNet incorporates a noise discriminator that quantifies local noise intensity by analyzing normal similarity, and a reverse monotonic decision function that maps this measure to an appropriate denoising module. Furthermore, we propose a path-selective iteration mechanism that dynamically re-evaluates the restoration state and re-plans the denoising route at each stage, enabling cross-stage skipping to minimize unnecessary computation. Extensive experiments on multiple benchmarks demonstrate that DSNet achieves state-of-the-art performance in noise suppression, geometric fidelity, and computational efficiency. The code are available at https://github.com/cz-61/DSNet.

1. Introduction

Three-dimensional (3D) point clouds serve as a fundamental data representation for real-world geometric structures and are widely applied in autonomous driving, robotic navigation, 3D reconstruction, and virtual or augmented reality. However, point cloud denoising faces fundamental challenges stemming from the complex and non-uniform nature of real-world noise. Sensor-induced Gaussian noise, environmental outliers, and geometric distortions often coexist with spatially varying intensities, making it difficult to design a universal strategy that simultaneously suppresses noise in heavily corrupted regions while preserving fine-grained geometric details in cleaner areas. Thus, point cloud denoising has become a key step to ensure data quality and to improve the performance of downstream tasks.

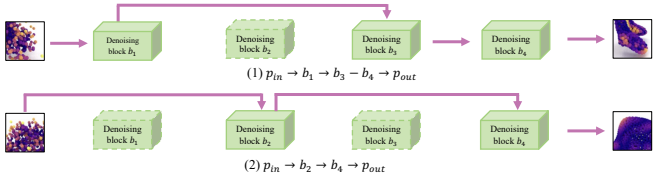

Figure 1. Illustration of the proposed routing on demand mechanism in DSNet. Depending on local noise intensity, each input patch follows a customized denoising route by selectively skipping or passing denoising blocks.

Traditional point cloud denoising methods [1–4, 14, 18, 22, 24, 26, 32, 37] are mostly based on geometric fitting frameworks like least-squares minimization, these works rely on explicit geometric priors but lack the statistical modeling capacity to handle such complexity. Although deep learning methods [25, 28, 30, 33, 34, 40, 43, 52, 53] have demonstrated significant improvements by learning robust feature representations from data, most existing frameworks, from single-stage models to progressive architectures [6, 8, 28, 40] , adopt rigid processing pipelines. These methods apply a uniform denoising strategy to all regions, whether through a single forward pass or a fixed sequence of refinement stages. This "one-size-fits-all" approach fails to account for spatial variations in noise, leading to over-smoothing in cleaner areas, under-denoising in heavily corrupted regions, and redundant computation. This ultimately compromises both geometric fidelity and computational efficiency.

To address these challenges, we propose Dynamic Skip

*Corresponding author: Renjie Chen (renjiec@ustc.edu.cn).
This research was partially funded by the National Natural Science Foundation of China (12494552) and the NSF of Anhui Province of China (2508085MA001).

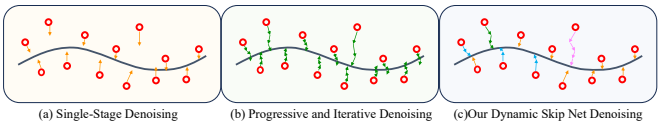

Figure 2. Comparison of denoising paradigms: (a) single-stage, (b) fixed-path progressive, and (c) our dynamic path selection approach.

Net (DSNet), a progressive point cloud denoising framework based on routing on demand. The core objective of DSNet is to adaptively plan the denoising iteration path according to the noise distribution and geometric characteristics of the input point cloud, thereby achieving a better trade-off between denoising accuracy and efficiency. The key innovation lies in enabling the network to perceive local noise intensity and dynamically route each patch through appropriate denoising modules. Specifically, DSNet employs a normal-similarity-based discriminator to quantify geometric degradation and guide adaptive path planning. Through a path-selective iteration mechanism, different patches can follow distinct processing routes—clean regions may skip intensive denoising stages while corrupted areas receive comprehensive refinement. This dynamic paradigm transforms conventional static progressive networks into adaptive systems, significantly enhancing both computational efficiency and geometric reconstruction quality. Figure 1 illustrates the concept of routing on demand selection strategy of our method.

The main contributions of this work are summarized as follows:

- We propose a novel **normal-similarity–driven noise discriminator** that effectively quantifies local geometric degradation, providing reliable guidance for dynamic path planning across regions with varying noise intensities.
- We introduce an innovative **path-selective iteration mechanism** that enables on-demand denoising by adaptively adjusting inference paths based on residual noise assessment, thereby achieving efficient cross-stage skip inference without redundant computation.
- To ensure structural stability throughout the progressive denoising process, we design a **multi-stage training strategy** that incorporates intermediate supervision and geometric consistency constraints to maintain smoothness and structural stability during progressive denoising.
- We conduct comprehensive experiments demonstrating that DSNet achieves state-of-the-art performance on multiple point cloud denoising benchmarks in terms of noise suppression, geometric restoration, and computational efficiency.

2. Related Work

Early point cloud denoising methods relied on geometric priors such as surface fitting [20, 21], bilateral filtering [13, 19, 51], and Laplacian smoothing [10, 38], but struggled to balance noise suppression with detail preservation. With the advancement of deep learning frameworks [27, 31, 33, 34, 43, 48], data-driven methods have demonstrated superior performance. Contemporary approaches can be categorized into single-stage and progressive/iterative methods.

2.1. Single-Stage Denoising Methods

Single-stage methods recover clean point clouds through a single forward pass. Early works focused on displacement prediction [35, 49], later enhanced through multi-scale perception [17], contrastive learning [7], and adaptive encoding strategies [15]. Recent approaches shifted to latent-space modeling via normalizing flows [29, 30, 45] and Transformer-based architectures [25, 41, 53] for unified global-local feature fusion. Despite these advances, single-stage methods adopt fixed one-pass inference, limiting adaptability to non-uniform noise.

2.2. Progressive and Iterative Denoising Methods

Progressive methods decompose denoising into multiple refinement steps. Diffusion-model-based approaches learn data distribution gradients [5, 16, 23, 28, 46, 50] or formulate the task as diffusion bridges [11, 40, 44] and flow velocity fields [9]. Explicit iterative architectures employ recurrent structures [6], weight-sharing modules [8], state-space models [52], and reinforcement learning-based routing [12].

While progressive methods outperform single-stage approaches, most adopt fixed iterative pipelines that uniformly process all regions, failing to account for spatially varying noise distributions. This leads to redundant computation in low-noise areas and insufficient refinement in high-noise regions.

3. Method

3.1. Motivation

Recent progressive denoising methods decompose tasks into refinement stages but enforce fixed pipelines, ignoring non-uniform noise. This causes redundant computation and over-smoothing in low-noise regions, while under-refining high-noise areas, compromising fidelity and efficiency.

This raises a key question: Can networks dynamically allocate computation by planning optimal paths per region? We shift to adaptive inference, using surface normal deviations as a noise proxy. As shown in Figure 2, Our adaptive path selection dynamically patches—intensive for cor-

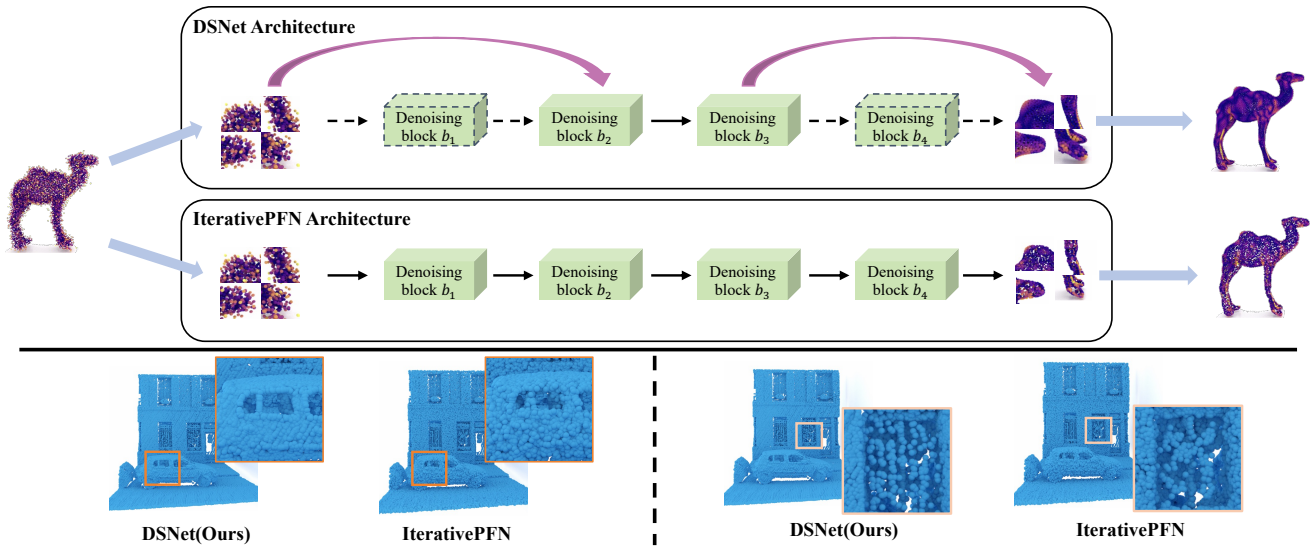

Figure 3. Our pipeline. DSNet employs a dynamic routing mechanism that adaptively selects denoising paths per patch based on local noise levels, unlike fixed-path methods (e.g., IterativePFN) that uniformly refine and often over-smooth. Enhanced detail preservation is evident in the zoomed comparison (bottom).

rupted ones, skipping for clean ones—balancing quality and efficiency.

3.2. Overview

Given a noisy point cloud $\hat{P}$, our objective is to produce a denoised result that approximates the clean point cloud $\overline{P}$ as closely as possible. The proposed DSNet consists of a series of iterative denoising modules, each designed to progressively refine the geometric structure. To enable efficient processing, the input point cloud is first decomposed into overlapping patches.

As illustrated in Figure 3, we assign distinct ground truths for each iteration module inspired by IterativePFN [8], rather than using a uniform target throughout the entire denoising process. We define:

$$\overline{P}_{gt_k} = \overline{P} + \sigma_i \xi, \quad \xi \sim \mathcal{N}(0, I), \tag{1}$$

where σ_i represents the noise standard deviation at iteration i, and is progressively reduced according to:

$$\sigma_i = \frac{\sigma_{i-1}}{\gamma}, \tag{2}$$

with γ denoting the noise decay factor, which we set to $\frac{16}{L}$ to ensure smooth degradation across iterations.

For each input patch, a surface-normal-based noise discriminator is first applied to assess its residual noise level and determine the appropriate entry point for denoising. After each denoising module, the noise level is re-evaluated to dynamically adjust the entry to the next stage. Within each iteration module, we employ a U-Net architecture to progressively refine the patch while adaptively determining the number of encoder–decoder layers based on the estimated noise intensity. Through this adaptive mechanism, DSNet establishes a fine-grained, on-demand denoising scheme tailored to each individual patch, thereby achieving an optimal balance between denoising quality and computational efficiency.

3.3. Network Architecture

In this section, we provide a detailed description of the DSNet architecture, which consists of three key components: a noise discriminator for adaptive path planning, a series of iterative denoising modules, and a dynamic skip mechanism for efficient inference.

Given an input noisy point cloud $\hat{P}$, we first employ the Farthest Point Sampling (FPS) algorithm to extract center points. For each selected center point, a K-Nearest Neighbor (KNN) search identifies the 1000 nearest neighbors to form a local patch. The next step is to evaluate the noise complexity of each patch to determine the appropriate denoising module. To this end, we design a normal-based noise discriminator, which quantifies geometric degradation through surface normal deviations.

Normal-Similarity-based Noise Discriminator. In our adaptive denoising framework, accurately quantifying the noise level and structural complexity of each patch is crucial for dynamic path selection. Through experimental observations, we find that the angular deviation between surface normals of noisy and clean point clouds is highly correlated with the degree of geometric degradation. Motivated by this insight, we propose a novel **Normal-Similarity Factor** ρ,

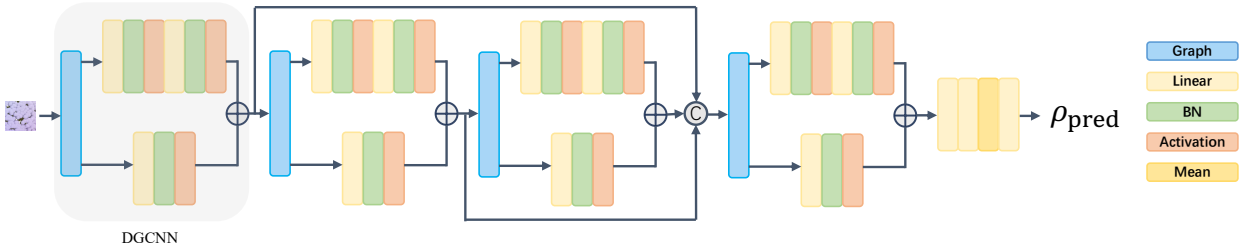

Figure 4. Overview of the noise discriminator network for computing ρ.

which utilizes local normal information to sensitively characterize structural distortions:

$$\rho = \frac{1}{n}\sum_{i=1}^{n} \exp\left(-\frac{\theta_i^4}{\gamma}\right), \tag{3}$$

where n denotes the number of points in the patch, and θ_i represents the angular difference between the clean and noisy normals at point i:

$$\theta_i = \arccos(n_{\text{clean},i} \cdot n_{\text{noisy},i}). \tag{4}$$

Here, γ is a hyperparameter controlling the sensitivity to noise deviations. The quartic term θ_i^4 enlarges penalties for large angular deviations, making $\rho \in (0, 1]$ robust to small perturbations while sensitive to significant structural distortions.

Decision Function $\lambda(\rho)$. To guide each patch to its proper processing module, the continuous similarity score ρ must be mapped to a discrete module index within the range $[N_{\min}, L]$. This mapping function must satisfy several key requirements: (1) *inverse monotonicity*—patches with lower ρ (indicating higher noise) should enter earlier, more aggressive denoising stages, while patches with higher ρ (indicating lower noise) should skip to later, fine-tuning stages; (2) *nonlinear sensitivity*—the mapping should be more discriminative in the high-noise region where precise routing is critical, and less sensitive in the low-noise region where exact module selection is less crucial; and (3) *bounded output*—the result must be a valid integer index within the allowable module range.

To achieve these objectives, we design the decision function as:

$$\lambda(\rho) = \text{clip}\Big(\text{round}\Big(L - (L - N_{\min})\frac{\log(\beta\rho + 1)}{\log(\beta + 1)}\Big), N_{\min},\ L\Big) \tag{5}$$

where L denotes the total number of diffusion steps, $N_{\min}$ represents the earliest allowable entry point, and $\beta > 0$ is a tunable hyperparameter controlling the nonlinearity of the mapping.

The logarithmic term $\frac{\log(\beta\rho+1)}{\log(\beta+1)}$ serves as the core nonlinear transformation. When $\rho \to 0$ (extremely noisy patches), the denominator $\log(\beta\rho + 1) \to 0$, driving the ratio to its maximum, which pushes $\lambda(\rho)$ toward $N_{\min}$, forcing the patch through the complete denoising trajectory. Conversely, when $\rho \to 1$ (clean patches), both numerator and denominator approach $\log(\beta + 1)$, yielding a ratio close to 1, which results in $\lambda(\rho) \approx L$, allowing the patch to bypass most denoising steps.

The hyperparameter β modulates the curvature of this mapping: larger β increases the nonlinearity, making the function more sensitive to changes in ρ within the high-noise region (ρ close to 0), thereby enabling finer-grained routing for challenging patches; smaller β flattens the curve, distributing patches more uniformly across modules. The $\text{round}(\cdot)$ operation discretizes the continuous result to integer indices, while $\text{clip}(\cdot, N_{\min}, L)$ enforces hard boundaries to ensure outputs remain within the valid module range. This formulation thus provides a principled, interpretable, and tunable mechanism for adaptive path selection across varying noise levels.

Network Implementation. The noise discriminator is implemented using EdgeConv layers from DGCNN [43], as shown in Figure 4. Given an input patch $P \in \mathbb{R}^{n\times 3}$, we first construct a k-NN graph and perform hierarchical feature extraction through multiple EdgeConv layers that iteratively update node features via dynamic adjacency. The features from different layers are concatenated and aggregated to produce point-wise representations, which are then globally pooled and processed through MLPs to predict the noise similarity factor ρ.

3.4. Adaptive Mechanism of Iterative Path

In this section, we detail the core innovation of DSNet: a dynamic path planning mechanism that adaptively routes each patch through a tailored sequence of denoising modules based on continuous noise assessment. Unlike conventional cascaded architectures that enforce a rigid sequential flow (e.g., $\text{Net}_1 \to \text{Net}_2 \to \cdots \to \text{Net}_L$), where all patches traverse identical processing stages regardless of their noise characteristics, DSNet replans the denoising trajectory at each iteration by evaluating residual noise levels. This enables a spatially adaptive allocation of computational resources: heavily corrupted regions with complex geometry receive intensive multi-stage refinement, while cleaner re-

gions with simpler structures bypass unnecessary intermediate processing.

During training, each patch undergoes iterative refinement through multiple denoising modules. However, the processing path is not predetermined—instead, it is dynamically determined by the noise discriminator at each iteration. Specifically, for the k-th iteration, the denoising process follows three key steps:

(1) State Evaluation: The current patch P_{k-1} is fed into the noise discriminator to compute its similarity factor ρ_{k-1}, which quantifies the residual geometric degradation relative to the clean surface. This assessment captures both global noise characteristics and local structural distortions through surface normal deviations.

(2) Module Selection: Based on ρ_{k-1}, the decision function $\lambda(\rho_{k-1})$ maps the continuous similarity score to a discrete module index $t \in \{t_{k-1}+1, \ldots, L\}$. This mapping follows the principle that patches with lower similarity (higher noise) are routed to earlier, more aggressive denoising modules, while patches with higher similarity (lower noise) skip directly to later, fine-tuning modules. The selection process is governed by:

$$t_k = \lambda(\rho_{k-1}), \quad t_k \in \{t_{k-1}+1, \ldots, L\}, \tag{6}$$

where the selected module index t_k determines which denoising network Net_{t_k} will process the patch in the current iteration.

(3) Module Execution: The selected module Net_{t_k} then processes P_{k-1} to produce the refined output P_k. Formally:

$$P_k = \text{Net}_{t_k}(P_{k-1}), \quad t_k \in \{t_{k-1}+1, \ldots, L\}. \tag{7}$$

This three-step cycle repeats for a fixed number of iterations K_{total}, with the discriminator continuously reassessing the patch quality after each denoising step. Importantly, the denoising trajectory is not constrained to be sequential. For example, a patch might follow the path:

$$P_{\text{input}} \xrightarrow{\text{Net}_2} P_1 \xrightarrow{\text{Net}_5} P_2 \xrightarrow{\text{Net}_7} \cdots \xrightarrow{\text{Net}_L} P_{\text{output}}, \tag{8}$$

effectively bypassing intermediate modules $\{\text{Net}_3, \text{Net}_4, \text{Net}_6\}$ when the discriminator determines that the patch quality has improved sufficiently to skip those stages. This non-sequential routing distinguishes our method from prior iterative denoising approaches, such as IterativePFN [8], which enforce a fixed sequence of operations at both training and test time.

3.5. U-Net Iteration Module

As illustrated in Figure 5, the U-Net module adopts a hierarchical encoder-decoder architecture for multi-scale feature extraction and refinement.

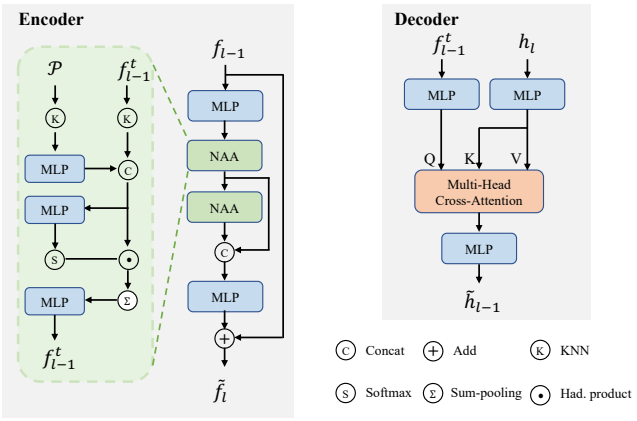

Figure 5. Overview of U-Net architecture. The encoder performs hierarchical feature extraction via neighborhood aggregation (NAA), while the decoder reconstructs features through multi-head cross-attention for cross-level fusion.

Encoder. At each level l, features are refined through two consecutive Neighborhood Attention Aggregation (NAA) operations with MLP-based residual connections:

$$f_l = \text{MLP}\big([\text{NAA}(f_{l-1}^0), \text{NAA}^2(f_{l-1}^0)]\big) + \text{MLP}(f_{l-1}), \tag{9}$$

where $[\cdot,\cdot]$ denotes concatenation. The point set P_l is downsampled via Farthest Point Sampling (FPS).

Decoder. The decoder upsamples features through distance-weighted interpolation followed by multi-head cross-attention (MHCA) module [39] for cross-level fusion:

$$h_{l-1} = \text{MLP}\big([\text{MHCA}(\varphi(f_{l-1}), \varphi(\tilde{h}_l)), f_{l-1}]\big), \tag{10}$$

where $\tilde{h}_l$ is the interpolated feature from the coarser level, capturing both global context and local structural details.

3.6. Loss Function

To effectively train DSNet, we design a composite loss function that integrates multi-stage supervision and geometric structure preservation, ensuring both global consistency and local displacement coherence during progressive denoising.

Progressive Intermediate Ground Truth. At each denoising step k, we define an intermediate target $\overline{P}_{\text{gt}_k}$ with progressively reduced noise, where the final target corresponds to the clean ground truth point cloud.

Single-Step Loss Definition. For each iteration k, the loss function consists of two components:

$$L_k = L_{\text{cd}}(P_k, \overline{P}_{\text{gt}_k}) + L_{\text{disp}}(P_{k-1}, \overline{P}_{\text{gt}_k}, P_k), \tag{11}$$

where L_{cd} measures the Chamfer distance between the predicted and target point clouds, and the displacement loss

L_{disp} is defined as:

$$L_{\text{disp}} = \|(P_k - P_{k-1}) - P_{\text{nearest}}\|_2. \tag{12}$$

Here, P_{nearest} represents the displacement vector from P_{k-1} to its nearest neighbor in P_{gt_k}.

The total loss aggregates all intermediate losses with equal weights:

$$L = \sum_{k=1}^{K_{\text{total}}} L_k, \tag{13}$$

ensuring consistent supervision and stable optimization throughout the coarse-to-fine denoising process.

4. Experiments

4.1. Datasets and Settings

Training dataset. Following previous studies [8, 15], we adopt the PU-Net dataset [47] for training our network. The training set consists of 40 meshes, from which point clouds are sampled using Poisson disk sampling at resolutions of 10K, 30K, and 50K points. This yields a total of 120 training point clouds. To simulate realistic measurement imperfections, Gaussian noise with a standard deviation ranging from 0.05 to 0.2 times the radius of the bounding sphere is added to each sampled point cloud.

Testing datasets. For quantitative evaluation, we test on the PU-Net dataset at 10K and 50K resolutions, yielding 40 noisy point clouds. To assess the generalization ability of our approach, we further evaluate it on the real-scanned Kinect dataset [42], which consists of scans collected by Microsoft Kinect v1 and Kinect v2 devices. Additionally, we evaluate on the Paris-Rue-Madame dataset [36], which contains large-scale outdoor scans of street scenes in the 6th district of Paris obtained via a 3D mobile laser scanning system. These data include complex real-world noise artifacts, making them an excellent benchmark for testing robustness under real scanning conditions. As ground-truth surfaces are unavailable for these real-world scans, we report qualitative results for this dataset.

Implementation Details. Our method is implemented in PyTorch and trained on 8 NVIDIA A100-SXM4-80GB. We employ the Adam optimizer with a learning rate of 1×10^{-4}. All input point clouds are normalized to fit within a unit sphere before training.

The training process begins with an independent pre-training of the noise discriminator to obtain a stable and discriminative noise feature extractor. Subsequently, the pre-trained noise discriminator is employed to guide the training of the DSNet network, forming a joint optimization framework that enhances the model's convergence rate and overall training stability.

4.2. Comparisons on synthetic data

To assess the effectiveness of DSNet, we conduct a comparative evaluation against several recent point cloud denoising models, encompassing both single-stage and progressive frameworks, as summarized in Table 1.

Within the single-stage category, PD-Flow relies on a normalizing-flow-based density estimation approach. It performs reliably under low-resolution and low-noise conditions (e.g., 10K points, 1% noise), yet its denoising performance degrades noticeably as both the resolution and noise level increase (e.g., 50K points, 2.5% noise). Likewise, ASDN adopts an adaptive spatial sampling strategy that enhances local geometry preservation to some degree, but its robustness weakens once the noise level exceeds 2%.

Across all settings, DSNet demonstrates consistently superior outcomes. Regardless of resolution (10K or 50K) or noise level (1% to 2.5%), it achieves the lowest Chamfer Distance (CD) and Point-to-Mesh (P2M) errors. For instance, at 50K points and 2.5% noise, DSNet attains a CD of 0.762 and a P2M of 0.514, significantly outperforming the other methods. These results confirm that DSNet maintains stable denoising performance across diverse conditions while exhibiting strong robustness and effective structural preservation. This performance improvement stems from DSNet's adaptive path selection mechanism, which not only improves computational efficiency but also enhances geometric reconstruction quality. By dynamically routing clean regions to skip unnecessary denoising stages, DSNet effectively prevents over-smoothing and preserves fine-grained geometric details, while corrupted areas receive comprehensive refinement. This on-demand denoising strategy achieves superior balance between noise suppression and detail preservation, consistently outperforming both single-stage and progressive paradigms.

4.3. Comparisons on scanned data

Qualitative comparisons of different denoising methods on the Paris-Rue-Madame and Kinect datasets. As shown in Figure 7, in the first-row comparison of the window region, our method restores the edges of the window frames much more clearly and preserves the overall structural integrity. In the second-row comparison of the car's underside region, Score-Denoise, PD-Flow, IterativePFN, P2P-Bridge, and ASDN can only partially remove the noise, resulting in rough surfaces with noticeable scattered points and blurred boundaries. Although 3DMambaIPF produces relatively smoother results, it still shows isolated noisy points and loses local consistency. In contrast, our DSNet effectively removes most of the noise while maintaining smooth, continuous surfaces and sharp structural boundaries. The resulting point cloud is the most uniform and faithful to the underlying geometry, demonstrating that DSNet achieves the best balance between noise suppression

Method	10K points						50K points					
	1% noise		2% noise		2.5% noise		1% noise		2% noise		2.5% noise	
	CD ↓	P2M ↓	CD	P2M	CD	P2M	CD	P2M	CD	P2M	CD	P2M
Score [28]	2.521	0.753	3.686	1.381	4.412	1.925	0.716	0.400	1.289	0.833	1.952	1.363
PD-Flow [29]	2.134	0.680	3.255	1.328	3.627	1.702	0.652	0.417	1.425	1.056	1.874	1.426
IterativePFN [8]	2.055	0.501	3.043	0.845	3.352	1.045	0.606	0.302	0.802	0.436	1.015	0.588
P2P-Bridge [40]	2.284	0.686	3.202	1.114	3.531	1.386	0.586	0.330	0.902	0.580	1.165	0.803
ASDN [15]	1.871	0.488	2.624	0.789	2.697	0.961	0.512	0.300	0.721	0.437	0.850	0.575
3DMambaIPF [52]	1.989	0.477	2.995	0.803	3.262	0.992	0.589	0.291	0.755	0.405	0.928	0.531
DSNet (ours)	**1.829**	**0.474**	**2.451**	**0.764**	**2.604**	**0.909**	**0.468**	**0.289**	**0.603**	**0.399**	**0.762**	**0.514**

Table 1. Results on PU-Net Dataset. CD and P2M distances are multiplied by 10^4.

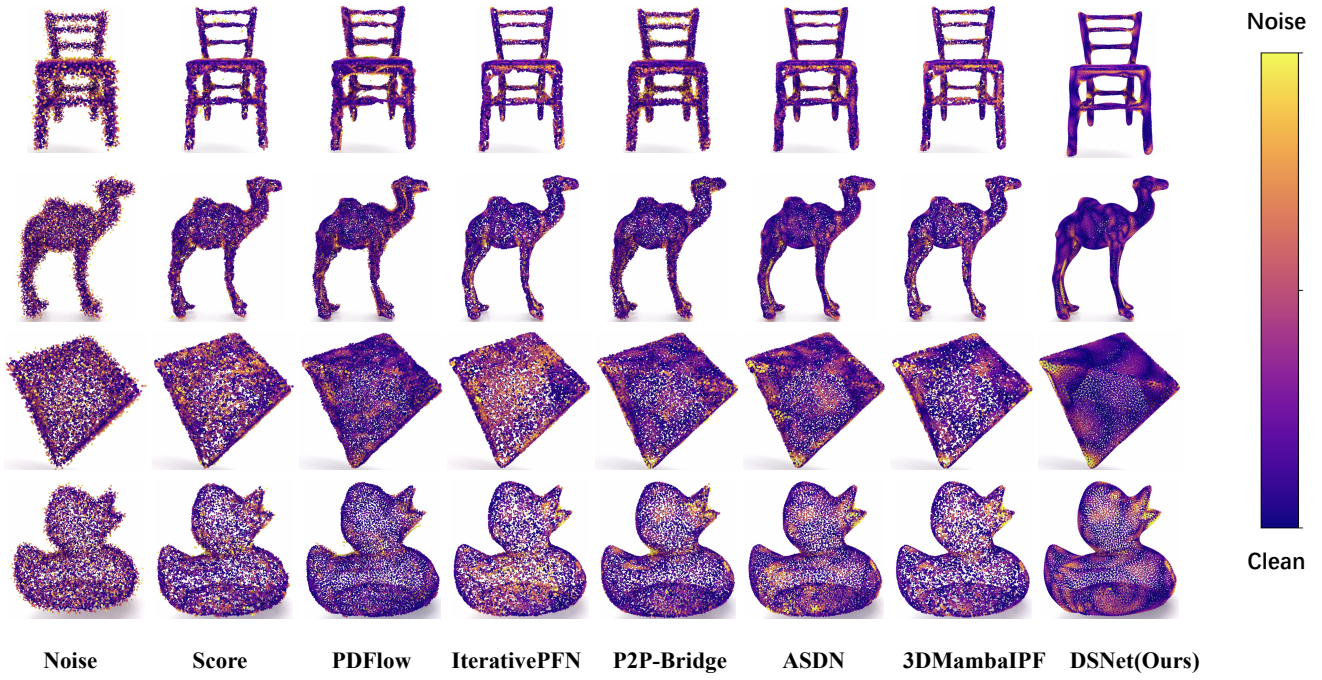

Figure 6. Visual results of point-wise P2M distance for shapes at 10K resolution with 3% Gaussian noise. The color bar on the right indicates the distance field from noisy to clean shapes.

Method	**CD ↓**	**P2M ↓**
Score-Denoise	1.061	0.882
PD-Flow	1.145	0.909
IterativePFN	0.993	0.867
P2P-Bridge	0.974	0.854
ASDN	1.007	0.877
3DMambaIPF	1.008	0.883
DSNet (ours)	1.040	0.885

Table 2. Results on Kinect Dataset. CD and P2M distances are multiplied by 10^4.

and detail preservation. Quantitatively, DSNet performs on par with other state-of-the-art methods on the Kinect, the results are listed in Table 2. Although the improvements are less pronounced compared with synthetic settings, the results remain competitive, demonstrating the general applicability of our approach while leaving space for future enhancement in cross-domain generalization.

5. Ablation Study

To validate the effectiveness of our proposed components, we conduct ablation studies on the number of progressive denoising stages and the dynamic path selection mechanism. Experiments are performed on the PU-Net dataset

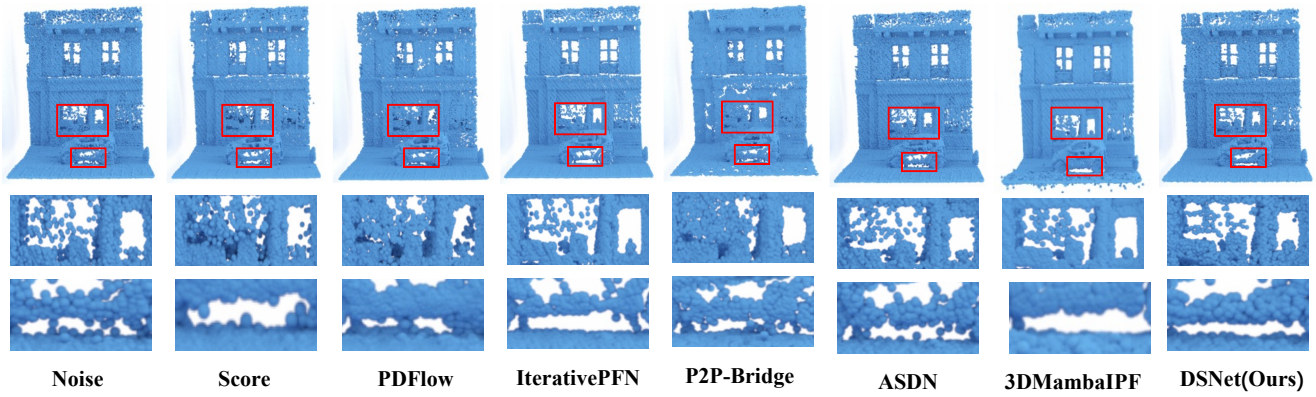

Figure 7. Qualitative Denoising Results on the Real-World Paris-Rue-Madame Dataset.

Method	10k points						50k points					
	1% noise		2% noise		2.5% noise		1% noise		2% noise		2.5% noise	
	CD ↓	P2M ↓	CD ↓	P2M ↓	CD ↓	P2M ↓	CD ↓	P2M ↓	CD ↓	P2M ↓	CD ↓	P2M ↓
DSNet-2	1.919	0.510	2.553	0.787	2.756	0.953	0.496	0.301	0.676	0.428	0.868	0.564
DSNet-Static-2	1.914	0.529	2.573	0.823	2.779	0.979	0.522	0.321	0.768	0.485	0.975	0.631
DSNet-3	1.867	0.489	2.539	0.779	2.723	0.925	0.498	0.298	0.674	0.411	0.845	0.528
DSNet-Static-3	1.904	0.503	2.521	0.778	2.848	1.015	0.511	0.311	0.713	0.448	0.811	0.535
DSNet-4	**1.829**	**0.474**	**2.451**	**0.764**	**2.604**	**0.909**	**0.468**	**0.289**	**0.603**	**0.399**	**0.762**	**0.514**
DSNet-Static-4	1.897	0.483	2.527	0.765	2.824	0.988	0.496	0.299	0.707	0.441	1.017	0.671
DSNet-5	1.847	0.481	2.478	0.771	2.635	0.918	0.473	0.295	0.615	0.406	0.816	0.532
DSNet-Static-5	1.911	0.520	2.586	0.803	2.952	1.058	0.525	0.321	0.751	0.468	1.063	0.701

Table 3. Ablation study on DSNet with different stages and static path variants under various noise levels.

across various noise levels and resolutions.

5.1. Impact of the Number of Denoising Stages

We first analyze the impact of network depth, parameterized by the number of denoising stages (L). As shown in Table 3, we evaluate DSNet with $L \in \{2, 3, 4, 5\}$. Performance consistently improves as L increases from 2 to 4, demonstrating that a deeper architecture enables more effective coarse-to-fine refinement. However, performance slightly degrades at $L = 5$, which we attribute to the risk of over-smoothing or error accumulation. Consequently, we adopt $L = 4$ as the optimal configuration, as it strikes the best balance between denoising capability and model complexity.

5.2. Effectiveness of Dynamic Path Selection

To validate our core contribution, we compare the full model (DSNet-L) against a static variant (DSNet-Static-L) that forces all patches through a fixed sequential path. The results in Table 3 reveal a critical trade-off. At lower noise levels, our dynamic DSNet outperforms the static model. This is because adaptive path selection successfully prevents over-smoothing on cleaner patches by skipping unnecessary stages.

Conversely, under extreme noise conditions (e.g., 3% noise), the static model proves more robust. We hypothesize that severe noise degrades the geometric cues used by our discriminator, making a guaranteed, full-refinement path more reliable. Nonetheless, this confirms that our dynamic mechanism is highly effective for its intended purpose: achieving efficient, on-demand denoising for point clouds with realistic, non-uniform noise.

6. Conclusion

In this paper, we present DSNet, a progressive point cloud denoising framework with a dynamic path selection mechanism adaptively routes each local patch through the denoising pipeline. Built upon a normal-similarity–driven noise discriminator and a path-selective iteration mechanism, DSNet prevents over-smoothing in clean regions while ensuring thorough processing of corrupted areas. Experiments on multiple benchmarks demonstrate DSNet achieves state-of-the-art performance in noise suppression and geometric fidelity.

References

[1] Anders Adamson and Marc Alexa. Point-sampled cell complexes. In *ACM SIGGRAPH 2006 Papers*, pages 671–680. 2006. 1

[2] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David levin, and Claudio T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on visualization and computer graphics*, 9(1):3–15, 2003.

[3] Haim Avron, Andrei Sharf, Chen Greif, and Daniel Cohen-Or. l1-sparse reconstruction of sharp point set surfaces. *ACM Transactions on Graphics (TOG)*, 29(5):1–12, 2010.

[4] Yizhak Ben-Shabat and Stephen Gould. Deepfit: 3d surface fitting via neural network weighted least squares. In *European conference on computer vision*, pages 20–34. Springer, 2020. 1

[5] Haolan Chen, Bi'an Du, Shitong luo, and Wei Hu. Deep point set resampling via gradient fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):2913–2930, 2022. 2

[6] Honghua Chen, Zeyong Wei, Xianzhi li, Yabin Xu, Mingqiang Wei, and Jun Wang. Repcd-net: Feature-aware recurrent point cloud denoising network. *International Journal of Computer Vision*, 130(3):615–629, 2022. 1, 2

[7] Dasith de Silva Edirimuni, Xuequan lu, Gang li, and Antonio Robles-Kelly. Contrastive learning for joint normal estimation and point cloud filtering. *IEEE Transactions on Visualization and Computer Graphics*, 30(8):4527–4541, 2023. 2

[8] Dasith de Silva Edirimuni, Xuequan lu, Zhiwen Shao, Gang li, Antonio Robles-Kelly, and Ying He. Iterativepfn: True iterative point cloud filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13530–13539, 2023. 1, 2, 3, 5, 6, 7

[9] Dasith de Silva Edirimuni, Xuequan lu, Gang li, lei Wei, Antonio Robles-Kelly, and Hongdong li. Straightpcf: Straight point cloud filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20721–20730, 2024. 2

[10] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324, 1999. 2

[11] Yi Du, Zhipeng Zhao, Shaoshu Su, Sharath Golluri, Haoze Zheng, Runmao Yao, and Chen Wang. Superpc: a single diffusion model for point cloud completion, upsampling, denoising, and colorization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16953–16964, 2025. 2

[12] Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017. 2

[13] Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or. Bilateral mesh denoising. In *ACM SIGGRAPH 2003 Papers*, pages 950–953. 2003. 2

[14] Gaël Guennebaud and Markus Gross. Algebraic point set surfaces. In *ACM siggraph 2007 papers*, pages 23–es. 2007. 1

[15] Chuchen Guo, Weijie Zhou, Zheng liu, and Ying He. You should learn to stop denoising on point clouds in advance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3212–3219, 2025. 2, 6, 7

[16] Qianjiang Hu and Wei Hu. Dynamic point cloud denoising via gradient fields. *ACM Transactions on Multimedia Computing, Communications and Applications*, 21(4):1–24, 2025. 2

[17] Anyi Huang, Qian Xie, Zhoutao Wang, Dening lu, Mingqiang Wei, and Jun Wang. Modnet: Multi-offset point cloud denoising network customized for multi-scale patches. In *Computer Graphics Forum*, pages 109–119. Wiley Online library, 2022. 2

[18] Hui Huang, Dan li, Hao Zhang, Uri Ascher, and Daniel Cohen-Or. Consolidation of unorganized point clouds for surface reconstruction. *ACM transactions on graphics (TOG)*, 28(5):1–7, 2009. 1

[19] Thouis R Jones, Frédo Durand, and Mathieu Desbrun. Non-iterative, feature-preserving mesh smoothing. In *ACM SIGGRAPH 2003 Papers*, pages 943–949. 2003. 2

[20] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013. 2

[21] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006. 2

[22] David levin. The approximation power of moving least-squares. *Mathematics of computation*, 67(224):1517–1531, 1998. 1

[23] Zikuan li, Qiaoyun Wu, Jialin Zhang, Kaijun Zhang, and Jun Wang. Noise-injected spiking graph convolution for energy-efficient 3d point cloud denoising. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 18629–18637, 2025. 2

[24] Yaron lipman, Daniel Cohen-Or, David levin, and Hillel Tal-Ezer. Parameterization-free projection for geometry reconstruction. *ACM Transactions on Graphics (ToG)*, 26(3):22–es, 2007. 1

[25] Zheng liu, Weijie Zhou, Chuchen Guo, Qinjun Qiu, and Zhong Xie. Pyramidpcd: A novel pyramid network for point cloud denoising. *Pattern Recognition*, 161:111228, 2025. 1, 2

[26] Xuequan lu, Scott Schaefer, Jun luo, lizhuang Ma, and Ying He. low rank matrix approximation for 3d geometry filtering. *IEEE transactions on visualization and computer graphics*, 28(4):1835–1847, 2020. 1

[27] Shitong Luo and Wei Hu. Differentiable manifold reconstruction for point cloud denoising. In *Proceedings of the 28th ACM international conference on multimedia*, pages 1330–1338, 2020. 2

[28] Shitong luo and Wei Hu. Score-based point cloud denoising. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4583–4592, 2021. 1, 2, 7

[29] Aihua Mao, Zihui Du, Yu-Hui Wen, Jun Xuan, and Yong-Jin liu. Pd-flow: A point cloud denoising framework with normalizing flows. In *European Conference on Computer Vision*, pages 398–415. Springer, 2022. 2, 7

[30] Aihua Mao, Biao Yan, Zijing Ma, and Ying He. Denoising point clouds in latent space via graph convolution and invertible neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5768–5777, 2024. 1, 2

[31] Francesca Pistilli, Giulia Fracastoro, Diego Valsesia, and Enrico Magli. Learning graph-convolutional representations for point cloud denoising. In *European conference on computer vision*, pages 103–118. Springer, 2020. 2

[32] Reinhold Preiner, Oliver Mattausch, Murat Arikan, Renato Pajarola, and Michael Wimmer. Continuous projection for fast l1 reconstruction. *ACM Trans. Graph.*, 33(4):47–1, 2014. 1

[33] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 77–85, 2017. 1, 2

[34] Charles Ruizhongtai Qi, li Yi, Hao Su, and leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1, 2

[35] Marie-Julie Rakotosaona, Vittorio la Barbera, Paul Guerrero, Niloy J Mitra, and Maks Ovsjanikov. Pointcleannet: learning to denoise and remove outliers from dense point clouds. In *Computer graphics forum*, pages 185–203. Wiley Online library, 2020. 2

[36] Andrés Serna, Beatriz Marcotegui, François Goulette, and Jean-Emmanuel Deschaud. Paris-rue-madame database: a 3d mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods. In *4th international conference on pattern recognition, applications and methods ICPRAM 2014*, 2014. 6

[37] Yujing Sun, Scott Schaefer, and Wenping Wang. Denoising point sets via l0 minimization. *Computer Aided Geometric Design*, 35:2–15, 2015. 1

[38] Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358, 1995. 2

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 5

[40] Mathias Vogel, Keisuke Tateno, Marc Pollefeys, Federico Tombari, Marie-Julie Rakotosaona, and Francis Engelmann. P2p-bridge: Diffusion bridges for 3d point cloud denoising. In *European Conference on Computer Vision*, pages 184–201. Springer, 2024. 1, 2, 7

[41] Junbo Wang and Ying li. Transformer-based point cloud denoising network. In *2023 4th International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*, pages 384–388. IEEE, 2023. 2

[42] Peng-Shuai Wang, Yang liu, and Xin Tong. Mesh denoising via cascaded normal regression. *ACM Trans. Graph.*, 35(6): 232–1, 2016. 6

[43] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019. 1, 2, 4

[44] Zhaonan Wang, Manyi li, Shiqing Xin, and Changhe Tu. Adaptive and iterative point cloud denoising with score-based diffusion model. In *Computer Graphics Forum*, page e70149. Wiley Online library, 2025. 2

[45] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4541–4550, 2019. 2

[46] Wenming Yang, Zhouyan He, Yang Song, and Yeling Ma. 3d point cloud denoising method based on global feature guidance. *The Visual Computer*, 40(9):6137–6153, 2024. 2

[47] lequan Yu, Xianzhi li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2790–2799, 2018. 6

[48] Chengwei Zhang, Xueyi Zhang, Mingrui Lao, Tao Jiang, Xinhao Xu, Wenjie Li, Fubo Zhang, and Longyong Chen. Deep learning for point cloud denoising: A survey. *arXiv preprint arXiv:2508.11932*, 2025. 2

[49] Dongbo Zhang, Xuequan lu, Hong Qin, and Ying He. Pointfilter: Point cloud filtering via encoder-decoder modeling. *IEEE Transactions on Visualization and Computer Graphics*, 27(3):2015–2027, 2020. 2

[50] Yaping Zhao, Haitian Zheng, Zhongrui Wang, Jiebo luo, and Edmund Y lam. Point cloud denoising via momentum ascent in gradient fields. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 161–165. IEEE, 2023. 2

[51] Youyi Zheng, Hongbo Fu, Oscar Kin-Chung Au, and Chiew-lan Tai. Bilateral normal filtering for mesh denoising. *IEEE transactions on visualization and computer graphics*, 17 (10):1521–1530, 2010. 2

[52] Qingyuan Zhou, Weidong Yang, Ben Fei, Jingyi Xu, Rui Zhang, Keyi liu, Yeqi luo, and Ying He. 3dmambaipf: A state space model for iterative point cloud filtering via differentiable rendering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10843–10851, 2025. 1, 2, 7

[53] Xusheng Zhu, Shuai Ma, Daixin Chen, li Zhou, and Haibo Tang. MSaD-Net: a mix self-attention networks for 3D point cloud denoising. *IEEE Photonics Journal*, 15(3):1–7, 2023. 1, 2