

# 3DMambaIPF: A State Space Model for Iterative Point Cloud Filtering via Differentiable Rendering

Qingyuan Zhou<sup>1</sup>, Weidong Yang<sup>1\*</sup>, Ben Fei<sup>2\*</sup>, Jingyi Xu<sup>1</sup>, Rui Zhang<sup>1</sup>, Keyi Liu<sup>1</sup>, Yeqi Luo<sup>1</sup>, Ying He<sup>3</sup>

<sup>1</sup>School of Computer Science, Fudan University

<sup>2</sup>Department of Information Engineering, The Chinese University of Hong Kong

<sup>3</sup>College of Computing and Data Science, Nanyang Technological University

{zhouqy23, jyxu22, ruizhang22, 23210240242, 23212010018}@m.fudan.edu.cn; wdyang@fudan.edu.cn; benfei@cuhk.edu.hk; yhe@ntu.edu.sg

## Abstract

Noise is an inevitable aspect of point cloud acquisition, necessitating filtering as a fundamental task within the realm of 3D vision. Existing learning-based filtering methods have shown promising capabilities on commonly used datasets. Nonetheless, the effectiveness of these methods is constrained when dealing with a substantial quantity of point clouds. This limitation primarily stems from their limited denoising capabilities for dense and large-scale point clouds and their inclination to generate noisy outliers after denoising. To deal with this challenge, we introduce **3DMambaIPF**, for the first time, exploiting Selective State Space Models (SSMs) architecture to handle highly-dense and large-scale point clouds, capitalizing on its strengths in selective input processing and large context modeling capabilities. Additionally, we present a robust and fast differentiable rendering loss to constrain the noisy points around the surface. In contrast to previous methodologies, this differentiable rendering loss enhances the visual realism of denoised geometric structures and aligns point cloud boundaries more closely with those observed in real-world objects. Extensive evaluations on commonly used datasets (typically with up to 50K points) demonstrate that 3DMambaIPF achieves state-of-the-art results. Moreover, we showcase the superior scalability and efficiency of 3DMambaIPF on highly dense and large-scale point clouds with up to 500K points compared to off-the-shelf methods.

**Code** — <https://github.com/TsingyuanChou/3DMambaIPF>.

## Introduction

Point clouds, assemblies of numerous three-dimensional points denoting spatial positions, play a crucial role in representing scenes across multimedia and computer vision (CV) domains. Nevertheless, as point clouds from real-world environments are captured by devices such as LiDAR, depth cameras, or 3D scanners, noises are inevitably introduced during the scanning process (Fei et al. 2024b). The presence of these noises undermines the required data consistency and accuracy in the application scenarios of point clouds, making filtering of point clouds a fundamental task within the

realm of 3D vision (Fei et al. 2022, 2023). Additionally, due to the potential large-scale nature of these real-world scans, point clouds can exhibit substantial coordinate ranges and be highly dense, containing a significant number of points.

Traditional point cloud filtering methods rely on expert priors, but they are limited by the accuracy of these priors and are not suitable for complex and sparse point clouds. In recent years, deep learning-based point cloud filtering methods have emerged prominently. These methods commonly employ iterative modules that consist of stacked encoder-decoder pairs. Such architectures leverage the output of the previous module as the input for the subsequent module, thereby demonstrating superior performance. For instance, IterativePFN (de Silva Edirimuni et al. 2023) employs iterative modules and adaptive loss functions to iteratively and progressively refine noisy points toward adaptive ground truth (GT), yielding promising results on both synthetic and real-world datasets.

However, these point cloud filtering methods are only applicable to small-scale point clouds and not suitable for filtering highly dense and large-scale point clouds. Furthermore, previous works only have demonstrated satisfactory performance solely in low-noise environments, while limited to handling details in high-noise environments. This limitation is particularly evident in the generation of unrealistic boundaries, as well as a lack of appropriate constraints. Consequently, the main challenges are twofold: (i) denoising for highly dense and large-scale point clouds with complex geometries, (ii) denoising for high-noise environments.

Recently, State Space Models (SSMs), particularly structured SSMs (S4) (Gu, Goel, and Ré 2021), have shown great promise in Natural Language Processing (NLP) (Lu et al. 2024). Built upon the S4 framework, Mamba (Gu and Dao 2023) exhibits outstanding performance across many tasks in NLP and 2D CV (He et al. 2024; Yang et al. 2024; Zhao et al. 2024; Misra, Gala, and Orvieto 2024). The remarkable achievements of Mamba are a testament to its linear computational complexity and exceptional long-range context learning capabilities on extensive sequences. Moreover, the integration of time-varying parameters and a hardware-aware algorithm within Mamba marks a significant leap forward in the field of sequence modeling.

\*Corresponding authors

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

However, exploiting SSM models for point cloud filtering has yet to be fully explored. Direct application of Mamba may not fully address the challenges involved. Therefore, we propose **3DMambaIPF**, a novel **Iterative Point Clouds Filtering** model utilizing the **Mamba** module with differentiable point rendering techniques. Specifically, 3DMambaIPF is composed of multiple iterations of Mamba-Denoising Modules, each comprising a pair of Mamba encoders and decoders. Initially, the input point cloud is divided into patches, within which a graph structure is generated based on points and their neighbors to extract position features. In the Mamba-based Encoder, the Dynamic EdgeConv module encodes positional features with MLPs, which are then fed into a Mamba module to select and generate position-dependent features from sequential inputs. The Mamba-based Decoder is responsible for upsampling features into point cloud patches. During the training process, a novel rendering loss is introduced leveraging a differential point rendering technique, and the loss is computed by the disparities of rendered images of filtered point cloud and GT. Subsequently, an adaptive GT altering throughout the iterations is introduced, gradually transitioning from an initially noise-added state to the true GT. Following the completion of all iterations, a patch-stitching method is employed to reconstruct the filtered point cloud from the patches. Experimentally, 3DMambaIPF is trained using PU-Net (Yu et al. 2018) dataset and achieves state-of-the-art results compared to off-the-shelf methods. 3DMambaIPF is also able to well generalize to high-noise point clouds from the Stanford 3D Scanning Repository (Curless and Levoy 1996; Krishnamurthy and Levoy 1996) for synthetic objects with complex geometries. Moreover, 3DMambaIPF outperforms baseline methods in terms of visual perceptions on real-world scene datasets. In summary, our main contributions are as follows:

- A novel iterative Mamba-based backbone is designed for point cloud filtering, allowing for the effective modeling of long-sequence point cloud features. This approach enhances the accuracy and speed of highly dense and large-scale point cloud processing.
- A well-designed differentiable point rendering loss is introduced into 3DMambaIPF, supplanting the distance-based loss functions. This differentiable rendering loss endows 3DMambaIPF with the ability to handle edge denoising and makes the denoising results more realistic.
- 3DMambaIPF surpasses existing methods by achieving state-of-the-art results on the commonly used PU-Net dataset, while demonstrating excellent performance even on highly-dense and large-scale datasets with high noise added.

## Related Work

**Deep Learning-based Point Cloud Filtering Methods.** Deep Learning-based point cloud filtering methods are undergoing significant development to address the limitations of traditional approaches. The well-known PCN (Point-CleanNet) (Rakotosaona et al. 2020) model is based on PCP-Net (Guerrero et al. 2018) and focuses on removing outliers and reducing noise in unordered point clouds. However, due

to PCN’s denoising being performed point-wise rather than patch-wise, its efficiency is relatively low. PD-Flow (Mao et al. 2022) learns the distribution of noisy point sets and achieves denoising through noise disentanglement. However, the number of iterations needed for PD-Flow varies depending on the level of noise to achieve optimal results. GPDNet (Pistilli et al. 2020) employs graph-convolutional layers to dynamically build neighborhood graphs and utilizes the similarity of high-dimensional feature representations to construct intricate feature hierarchies. ScoreDenoise (Luo and Hu 2021) leverages the distribution model and score estimation from noisy point clouds. Pointfilter (Zhang et al. 2020) employs an encoder-decoder architecture, where points and their neighbors are taken as input for point-wise learning. It generates a displacement vector to accomplish denoising while preserving sharp features. IterativePFN (de Silva Edirimuni et al. 2023) leverages iterative filtering modules and adaptive GT progressively yet swiftly converges noisy points onto clean surfaces, achieving superior performance. Current point cloud filtering methods are primarily designed for sparse and small-scale datasets (up to 50K points) and struggle to effectively filter dense point clouds (approximately 500K points) with noise far from clean surfaces, particularly in large-scale point clouds. Furthermore, existing methods (Zhang et al. 2020) demonstrate limited effectiveness in filtering the edges of point clouds and capturing fine-grained characteristics. Considering these challenges, our work aims to introduce a model tailored for highly dense and large-scale point clouds characterized by high-density noise and robust filtering capabilities at point cloud edges.

**SSM-based Methods.** SSMs provide a robust mathematical framework rooted in control theory and are increasingly applied in fields such as NLP and CV to model complex temporal dynamics within dynamic systems. This novel modeling approach addresses the challenge of handling long sequences, a difficulty encountered by both Transformers and RNNs. By mitigating the quadratic computational complexity of Transformers and addressing the forgetting issue of RNNs, SSMs prove highly efficient in capturing temporal dependencies within sequential data.

Significant advancements have been made in the Mamba framework (Gu and Dao 2023), which integrates selective SSMs into a simplified architecture. This integration allows Mamba to achieve rapid inference, linear scalability in sequence length, and state-of-the-art performance across multiple modalities, while also enhancing the selectivity of state representation. These advancements collectively demonstrate the ongoing evolution of SSMs and their increasing significance in modeling temporal dynamics across diverse domains. Methods based on Mamba have achieved state-of-the-art results in various fields, including large language models (He et al. 2024), medical image analysis (Liao et al. 2024), and 3D point cloud perception (Liang et al. 2024). Taking Consideration of the advantages of Mamba in modeling long sequences, we introduce Mamba in our work to address highly-dense and large-scale point cloud modeling problems.

**Differentiable Rendering Techniques.** Rendering tech-

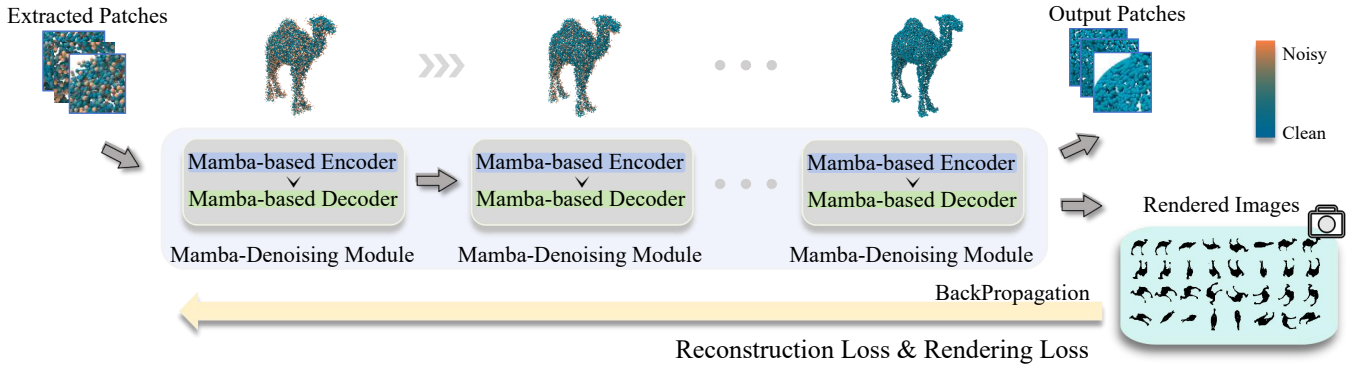


Figure 1: Overview of 3DMambaIPF. An Encoder-Decoder-based model built on Mamba named Mamba-Denoising Module is introduced for iterative filtering. The input noisy point cloud is partitioned into patches and fed into the iterative Mamba-Denoising Module. Upon completion of the iterations, clean point cloud patches are produced as output. To enhance the filtering of noisy points around the surface, a differentiable rendering method is introduced. The rendering loss and reconstruction loss are jointly backpropagated to update the parameters of 3DMambaIPF.

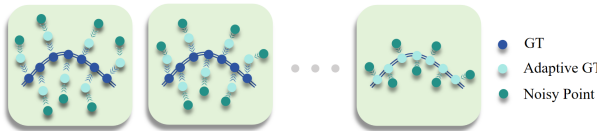


Figure 2: A Gaussian noise with a variable standard deviation is added to the GT to generate the adaptive GT. As the standard deviation gradually decreases, the adaptive GT approaches the GT with each iteration. Eventually, the GT no longer changes in the final iteration.

niques encompass algorithms employed to transform three-dimensional models into two-dimensional images. The differentiable rendering methods are designed for introduction into deep learning, facilitating 3D reconstruction through error backpropagation. Current differentiable rendering methods can be categorized into four types based on their geometric representations: point-based (Yifan et al. 2019; Müller, Weinmann, and Klein 2022; Kerbl et al. 2023; Fei et al. 2024a), triangle mesh-based (Laine et al. 2020; Liu et al. 2019), implicit function-based (Bangaru et al. 2022; Vicini, Speierer, and Jakob 2022), and volume-based (Mildenhall et al. 2021; Ren et al. 2023). In this paper, we propose an efficient differentiable rendering method based on point-based rendering (Insafutdinov and Dosovitskiy 2018) to render both the filtered and GT point clouds during training to generate the view loss, which is then propagated backward to adjust the network parameters.

## Method

### Overview

The overview of 3DMambaIPF is depicted in Figure 1, which utilizes a novel iterative Mamba-Denoising backbone and a differentiable point rendering technique. Mamba-Denoising backbone consists of a cascade of Mamba-based encoders and decoders, with each pair referred to as a Mamba-Denoising Module. After receiving an extracted

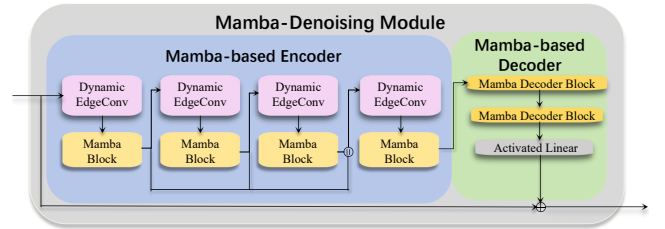


Figure 3: Mamba-Denoising Module for iterative point cloud filtering of 3DMambaIPF. The Mamba-based Encoder comprises Dynamic EdgeConv modules and Mamba Blocks, while the Mamba-based Decoder consists of Mamba Decoder Blocks and an Activated Linear module.

patch of the point cloud as input, a single Mamba-Denoising Module constructs a directed graph from the points within the patch. The directed graph is encoded into token sequences to extract features. The denoised point token sequences are then decoded, eliminating noise, and serve as the input for the next Mamba-Denoising Module.

Specifically, given a 3-dimensional clean point cloud, denoted as  $X_c = \{x_{c1}, x_{c2}, \dots, x_{cn}\} \subseteq \mathbb{R}^3$ , each point in  $X_c$  can be represented by 3-dimensional coordinates  $x_{ci} = (\alpha_i, \beta_i, \gamma_i)$ . To prepare the point cloud as input for 3DMambaIPF, the K-nearest neighbors (KNN) algorithm (Peterson 2009) is used to partition the point cloud into patches, and a directed graph is constructed within each patch. Considering a point cloud patch  $X \subseteq \mathbb{R}^3$  which generated by KNN and a directed graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ , where  $\mathcal{V} = \{x_1, x_2, \dots, x_K\}$  and  $\mathcal{E}$  represents the edges between  $x_i \in \mathcal{V}$  and its  $k$ -nearest neighbors.

Simultaneously, the adaptive GT, inspired by the iterative point cloud filtering method (de Silva Edirimuni et al. 2023), is introduced to gradually pull noisy points closer to the clean surface with each iteration. As shown in Figure 2, GT is adaptive at each iteration, except for the last one. Specifically, to generate the adaptive GT  $X_n \subseteq \mathbb{R}^3$ ,

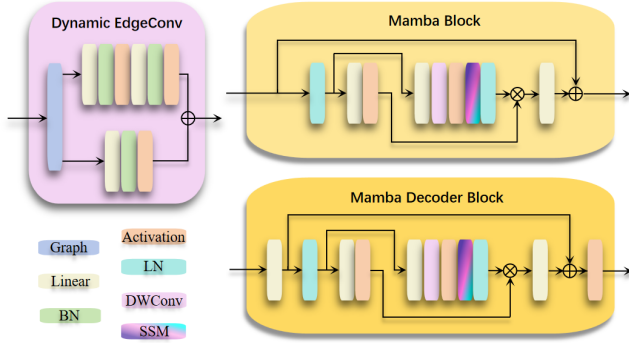


Figure 4: Detailed network architectures for Dynamic EdgeConv, Mamba Block, and Mamba Decoder Block.

perturbed points are formed by adding Gaussian noise to  $X_c$  with a standard deviation  $\sigma$  between 0.5% and 2% of the radius of the bounding sphere. As the number of iterations increases,  $\sigma$  decreases continuously, eventually reaching 0 at the final iteration. During each iteration, we aim to minimize the loss function to bring  $x_i$  closer to its nearest neighbors in adaptive GT.

Subsequently, a differentiable 3D point rendering loss is employed to refine the positions of noisy points, especially around the surface of the point clouds, by comparing the rendered images from filtered point clouds with the adaptive GT.

### Mamba-Denoising Backbone

The Mamba-Denoising Backbone is the core component of 3DMambaIPF for iterative point cloud filtering. As illustrated in Figure 3, the Mamba-Denoising backbone adopts an Encoder-Decoder structure. The encoder consists of four Dynamic EdgeConv modules and four corresponding Mamba Blocks respectively, while the decoder comprises a Mamba Decoder Block and a linear activation layer. The detailed network structure is illustrated in Figure 4.

The Dynamic EdgeConv module facilitates the conversion of input patches into a graph structure, subsequently processed through an MLP network to extract pertinent features. Initially, vertices undergo feature extraction via an MLP, capturing intrinsic characteristics. Subsequently, an additional MLP focuses on extracting positional features from vertices connected by edges to the target vertex, facilitating feature differentiation. Specifically, vertex features are updated as  $h_i^{l+1} = f(h_i^l) + \sum_{i,j} g(\text{concat}(h_i^l, h_j^l) - h_i^l)$ , where  $h^l$  denotes to features at layer  $l$ ,  $f(\cdot)$ ,  $g(\cdot)$  denote one-layer or two-layers MLPs,  $i$  is a vertex on the graph,  $(i, j)$  forms an edge. This process culminates in concatenating the target vertex’s features with the differences of features computed from all connected vertices, effectively deriving nuanced positional features within the patch. Following the feature extraction of the point cloud within the patch using MLPs, the resultant feature sequence is sent to the Mamba Encoder. This encoder selectively processes the input feature sequence and alters it into an output feature sequence. In the Decoder module, the Mamba Decoder Block selec-

tively processes the input feature sequence to reduce dimensionality, gradually reconstructing it into three dimensions. Linear activation is used to determine the output movement distance of a noisy point towards a clean surface after filtering. Utilizing a residual connection, the noisy points within the input patch are adjusted under the guidance of the movement distance, resulting in a denoised patch.

### Differentiable Rendering and Backpropagation

For enhanced filtering of the noisy points around the surface, we propose a differentiable rendering method and integrate it in 3DMambaIPF to explicitly quantify the level of noise and enable backpropagation. To render the point clouds, the 3D coordinates of the raw point clouds are first converted into the standard coordinate frame via a projective transformation aligned with the camera pose. Subsequently, each discretized point is expressed as scaled Gaussian densities, yielding the occupancy map, and making the backpropagation available. Through the introduction of a differentiable ray tracing operator, these occupancies are transformed into ray termination probabilities. Ultimately, the rendered image is generated by projecting the volume onto the plane.

Denoised point cloud and adaptive GT are utilized to produce  $K$  pairs of images with fixed camera poses using point rendering technique introduced above. Our differentiable rendering method employs three projection planes, allowing for the rendering of images with greater detail and coverage of the overall shapes, rather than projecting them onto a single plane.

The rendering loss is defined in Equation 1. It aims to characterize the local distribution of points and penalize noise distribution, encouraging the denoised point cloud to achieve an improved geometric appearance, particularly around the surfaces of the point cloud, throughout the back-propagation process:

$$\mathcal{L}_{Render}(t) = \frac{1}{K} \sum_{i=1}^K |V_i(\mathcal{P}_t) - V_i(\hat{\mathcal{P}})|, \quad (1)$$

where  $\hat{\mathcal{P}}$  denotes the predicted point cloud,  $\mathcal{P}_t$  denotes the adaptive GT at the  $t$ -th iteration,  $K$  is the quantity of rendered views, and  $V_i$  represents a rendered view in a specific camera position.

### Loss Function

Our loss function comprises two components: reconstruction loss and rendering loss. The reconstruction loss aims to minimize the distance between each point in the noisy point cloud and its closest counterpart in the adaptive GT, thereby structurally denoising the point cloud. Specifically, it seeks to bring every point in the noisy point cloud closer to its corresponding (the nearest) point in the adaptive GT.

Due to the possibility of overlapping regions within each patch during patch partitioning, we introduce a patch stitching method (Zhou et al. 2022). This method weights input points based on the proximity of each point  $p_i$  to a reference point  $p_r$  with a Gaussian distribution. Specifically, the weight  $w_i$  for each  $p_i$  is calculated as:

Method	Armadillo		Dragon	
	173K points		438K points	
	CD	P2M	CD	P2M
Noisy	50.19	47.96	38.92	37.51
Pointfilter	<b>6.41</b>	<b>4.82</b>	<b>12.10</b>	<b>12.70</b>
PD-Flow	52.30	50.08	51.48	50.00
ScoreDenoise	29.42	27.41	36.25	34.88
IterativePFN	<u>17.04</u>	<u>14.93</u>	20.43	18.98
3DMambaIPF (Ours)	<u>17.14</u>	<u>15.06</u>	<u>19.79</u>	<u>18.36</u>

Table 1: Results on Stanford 3D Scanning Repository. CD and P2M distances are multiplied by  $10^5$ .

$$w_i = \frac{\exp(-\|p_i - p_r\|^2/2r_s^2)}{\sum_i \exp(-\|p_i - p_r\|^2/2r_s^2)}, \quad (2)$$

where the support radius  $r_s$  is set to  $r/3$ , and  $r$  denotes to the patch radius. Then, the reconstruction loss at the  $t$ -th iteration can be defined as:

$$\mathcal{L}_{Recon}(t) = \sum_{\hat{p}_i \in \hat{\mathcal{P}}} w_i \min_{p_{ti} \in \mathcal{P}_t} \|p_{ti} - \hat{p}_i\|_2^2, \quad (3)$$

where  $\hat{\mathcal{P}}$  denotes the predicted point cloud,  $\mathcal{P}_t$  denotes the adaptive GT at the  $t$ -th iteration, and  $p_i$  represents a point in point cloud  $\mathcal{P}$ .

Loss function of one iteration is a weighted combination of the reconstruction loss and the rendering loss, and the ultimate loss is obtained by aggregating loss contributions across iterations, which are summed up as:

$$\mathcal{L} = \sum_{t=1}^T (\mathcal{L}_{Recon}(t) + \alpha \mathcal{L}_{Render}(t)), \quad (4)$$

where  $T$  is number of iteration times, and weight  $\alpha$  is 0.01 in experiments. The detailed pseudocode for training 3DMambaIPF is provided in Appendix.

## Experiments

### Overview of Comparative Baselines

The filtering capabilities of comparative baselines are conducted with DMRDenoise (Luo and Hu 2020), Pointfilter (Zhang et al. 2020), PD-Flow (Mao et al. 2022), ScoreDenoise (Luo and Hu 2021) and IterativePFN (de Silva Edirimuni et al. 2023). IterativePFN and 3DMambaIPF work on highly-dense and large-scale datasets. Pointfilter tends to obtain over-smoothed surfaces that deviate from the GT and fail to keep the original structure after filtering and struggles to work on highly-dense point clouds. On the other hand, the point cloud filtering task is not addressed as a **multi-modal** issue in previous works. All the baselines focus solely on either the movement of individual noise points or the generation of clean points, which leads to cluttered visual results, particularly around the surface of the point clouds.

### Datasets and Experimental Settings

Details about the PU-Net (Yu et al. 2018) dataset can be found in Appendix. Due to the limited number of points in PU-Net dataset, which contains a maximum of 50K points, point clouds from the Stanford 3D Scanning Repository (Curless and Levoy 1996; Krishnamurthy and Levoy 1996) are utilized to evaluate the filtering performance of large-scale synthetic objects. The preprocessing of datasets is detailed in the Appendix.

The experiments are conducted on NVIDIA A100 GPUs, and 3DMambaIPF is implemented with PyTorch 1.13.1 and CUDA 11.7. The Mamba modules are applied with 6 layers, and the total parameters of Mamba-Denoising Modules is 0.81M. Following previous works (Luo and Hu 2021; de Silva Edirimuni et al. 2023), the quantitative indicators CD and P2M are adopted for evaluation and comparison. More details can be found in Appendix.

### Comparisons on Highly-dense and Large-scale Datasets

The filtering performance of highly dense and large-scale point clouds is evaluated on synthetic models from the Stanford 3D Scanning Repository. The quantitative and visual results are shown in Table 1 and Figure 5. Due to the presence of dense and distant noise relative to the surface, some baselines struggle to effectively filter the point clouds. While Pointfilter achieves the best quantitative results, it lacks significant geometric details and accurate surface structures. It is found that the incorporation of the generative model into PD-Flow results in varied denoising outcomes across different experiments on the same point cloud. Despite potential variations in outcomes, PD-Flow generally falls short in effectively denoising highly-dense point clouds, particularly in severely noisy areas where it struggles to achieve significant noise reduction. ScoreDenoise, employing a patch partitioning method, exhibits gaps in results when processing highly-dense and large-scale point clouds. Although IterativePFN tackles this issue with patch-stitching and achieves competitive quantitative results, its performance is limited when dealing with point cloud surfaces and densely cluttered areas. As a single-modal filtering method, it struggles with the visual quality of point cloud surfaces, leading to gaps on the object surface that do not align with GT. On the contrary, 3DMambaIPF achieves the best performance when dealing with highly dense and large-scale point clouds. Additionally, the utilization of rendering loss results in significantly improved filtering effects and better preservation of geometric details, particularly around the surfaces of the point clouds.

### Comparisons on PU-Net Dataset

Table 2 presents the results of 3DMambaIPF compared to other baseline methods on PU-Net dataset. Figure 6 illustrates the visualization comparison on PU-Net dataset. Each visualized object contains 50K points with a Gaussian deviation of 2.5% of the bounding sphere radius. 3DMambaIPF outperforms all baseline methods across different resolutions and various levels of noise. Moreover, visual results

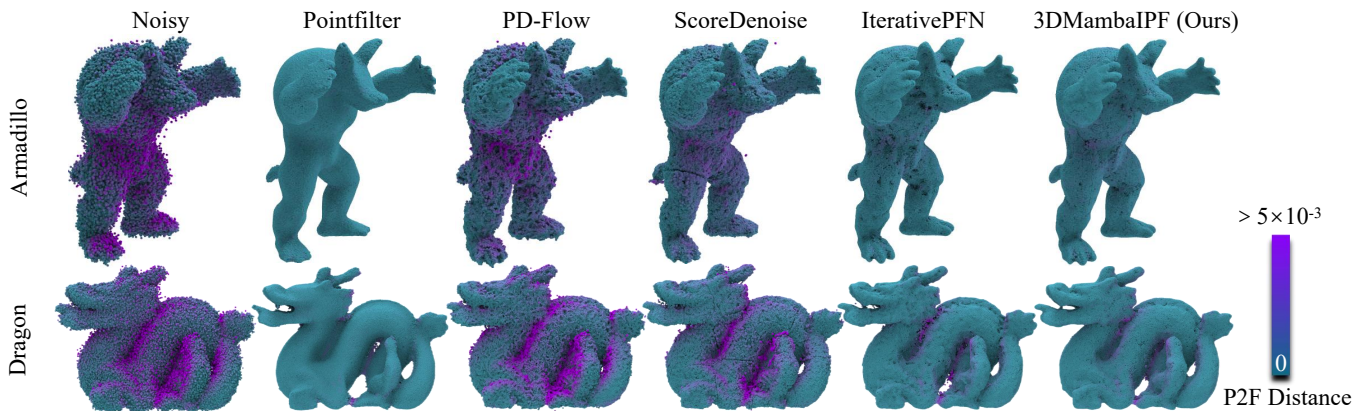


Figure 5: Visualization Comparisons on Stanford 3D Scanning Repository with 5% Gaussian deviation. Coloration of each point is determined by its point-wise P2F distance, with points exhibiting low P2F values (clean) depicted in green, while those with high P2F values (noisy) are portrayed in purple.

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
Noisy	36.90	16.03	79.39	47.72	105.02	70.03	18.69	12.82	50.48	41.36	72.49	62.03
PCN	36.86	15.99	79.26	47.59	104.86	69.87	11.03	6.46	19.78	13.70	32.03	24.86
GPDNet	23.10	7.14	42.84	18.55	58.37	30.66	10.49	6.35	32.88	25.03	50.85	41.34
DMRDenoise	47.12	21.96	50.85	25.23	52.77	26.69	12.05	7.62	14.43	9.70	16.96	11.90
Pointfilter	24.61	7.30	35.34	11.55	40.99	15.05	7.58	4.32	9.07	5.07	10.99	6.29
PD-Flow	21.26	6.74	32.46	13.24	36.27	17.02	6.51	4.16	12.70	9.21	18.74	14.26
ScoreDenoise	25.22	7.54	36.83	13.80	42.32	19.04	7.16	4.00	12.89	8.33	14.45	9.58
IterativePFN	20.55	5.01	30.43	8.43	33.53	10.46	6.05	3.02	8.03	4.36	10.15	5.88
<b>3DMambaIPF (Ours)</b>	<b>19.89</b>	<b>4.77</b>	<b>29.95</b>	<b>8.03</b>	<b>32.62</b>	<b>9.92</b>	<b>5.89</b>	<b>2.91</b>	<b>7.55</b>	<b>4.05</b>	<b>9.28</b>	<b>5.31</b>

Table 2: Results on PU-Net Dataset. CD and P2M distances are multiplied by  $10^5$ .

have shown that 3DMambaIPF exhibits superior filtering performance, particularly with high accuracy in boundary denoising. Specifically, while Pointfilter demonstrates competitive results in terms of CD and P2M metrics, it exhibits geometrically unrealistic deviations in boundary denoising for complex objects (e.g. casting). Compared to other baselines, IterativePFN excels both in terms of metrics and visual presentation, with its only drawback being slightly inferior denoising effects on point cloud boundaries. Conversely, 3DMambaIPF ensures the authenticity of geometric structures and exhibits superior denoising results for boundary sections, while also outperforming in metrics.

## Ablation Study

### Ablation on Modules

Ablation experiments are firstly conducted on the influencing factors of 3DMambaIPF, focusing on two aspects: i) the quantity of rendered images in point rendering, ii) the number of Mamba layers. Table 3 validates the enhancement of the filtering capability of 3DMambaIPF across various levels of noise with the introduction of view loss, and proceeds to examine how varying numbers of rendered views

affect filtering effectiveness. However, employing an excessive number of views will decrease the training speed, and detailed runtime metrics will be validated in appendix. Subsequently, ablation experiments are carried out on the number of Mamba layers. Table 3 illustrates that the denoising capability is at its peak with 6 Mamba layers.

### Runtime of Differentiable Rendering

An additional ablation for the runtime of rendered views is conducted as shown in Table 4.  $\text{Runtime}_{DR}$  represents the total time required to render both the GT and the predicted point clouds.  $\text{Runtime}_{Step}$  denotes the total time for one training step. Due to our devised network-free differentiable rendering, the runtime of differentiable rendering only utilizes 11.8%-16.5% of the total time for one training step. Except for this efficiency, differentiable rendering demonstrates notable improvements of 12.0% and 13.9% in CD and P2M metrics, respectively, as the number of views increases. This trade-off between runtime and performance is cost-effective.

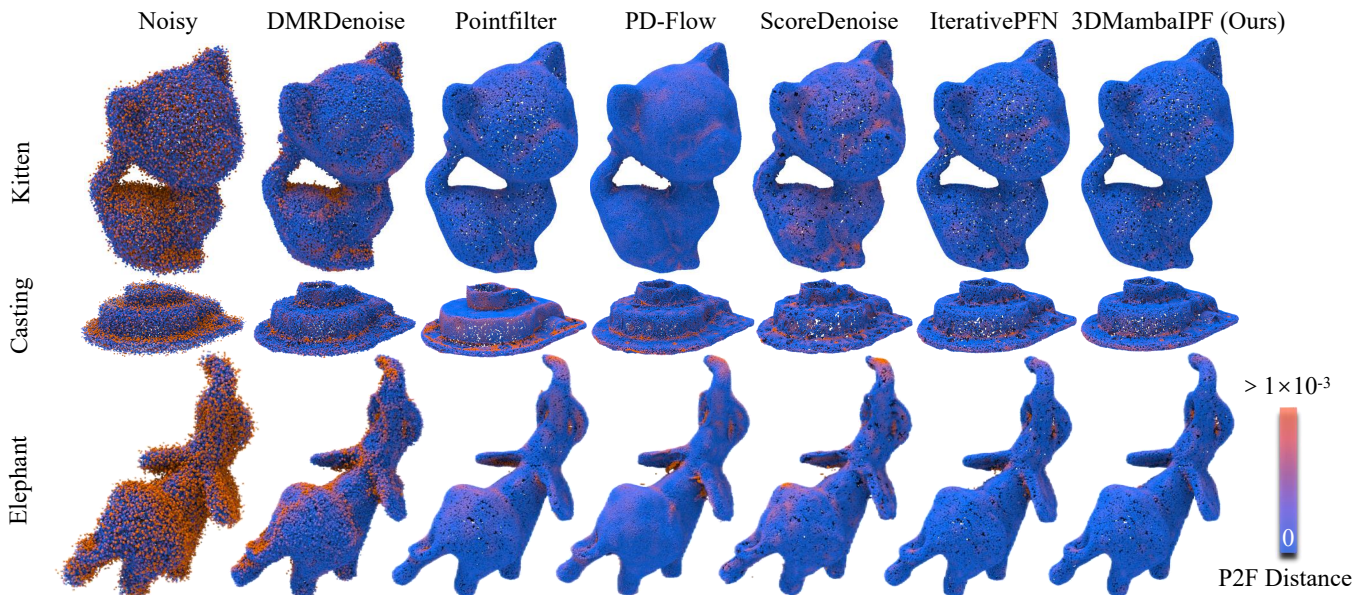


Figure 6: Visualization Comparisons on PU-Net Dataset of 50K points with 2.5% Gaussian deviation. The coloration of each point is determined by its point-wise P2F distance, with points exhibiting low P2F values (clean) depicted in blue, while those with high P2F values (noisy) are portrayed in red.

Rendering Views	Mamba Layers	50K points					
		1% noise		2% noise		2.5% noise	
		CD	P2M	CD	P2M	CD	P2M
/	/	6.05	3.02	8.03	4.36	10.15	5.88
/	6	5.94	2.92	7.70	4.14	9.41	5.37
8	6	6.04	2.99	8.06	4.36	10.39	6.05
16	6	6.21	3.14	7.87	4.29	9.63	5.60
24	6	5.96	2.97	7.81	4.26	9.56	5.53
32	6	<b>5.89</b>	<b>2.91</b>	<b>7.55</b>	<b>4.05</b>	<b>9.28</b>	<b>5.31</b>
32	1	8.15	4.71	10.82	6.53	13.23	8.20
32	3	6.12	3.09	8.05	4.43	9.58	5.54
32	6	<b>5.89</b>	<b>2.91</b>	<b>7.55</b>	<b>4.05</b>	<b>9.28</b>	<b>5.31</b>
32	9	6.15	3.08	8.01	4.38	9.89	5.74
32	12	6.12	3.07	8.09	4.42	9.85	5.67

Table 3: Ablation results of Mamba layers and rendered views on PU-Net 50K dataset. CD and P2M distances are multiplied by  $10^5$ .

### Comparisons 3D Mamba with Other Backbones

To validate the effectiveness of our designed 3D Mamba, ablation studies (Table 5) comparing the Mamba modules with Transformers and MLPs are conducted. For fairness, the number of modules, layers and embedding sizes are consistent with 3DMambaIPF. For intuitive results, differentiable rendering loss is not implemented in this ablation. 3DMambaIPF fulfills the best denoising performance while ensuring fast training speed as well as the smallest model size.

### Conclusion

In this study, we introduce 3DMambaIPF, a novel iterative Mamba-based point cloud filtering model that utilizes a differentiable rendering technique for the first time in this area.

Views	Metrics			
	CD	P2M	Runtime <sub>DR</sub> (s)	Runtime <sub>Step</sub> (s)
8	10.39	6.05	0.08	0.74
16	9.63	5.60	0.11	0.82
24	9.56	5.53	0.14	0.97
32	9.28	5.31	0.17	1.03

Table 4: Runtime analysis on differentiable rendering. CD and P2M metrics are multiplied by  $10^5$ , and tested on PU-Net Dataset of 50K points with 2.5% Gaussian deviation.

Methods	Metrics				
	CD	P2M	GFLOPs	FPS	MParams
with Transformers	14.02	8.56	178.76	2.31	158.00
with MLPs	30.17	21.71	20.78	<b>4.39</b>	4.56
with Mambas (Ours)	<b>9.42</b>	<b>5.37</b>	<b>16.92</b>	<u>3.62</u>	<b>0.81</b>

Table 5: Comparisons on Mamba with other backbones. CD and P2M metrics are multiplied by  $10^5$ , and tested on PU-Net Dataset of 50K points with 2.5% Gaussian deviation.

3DMambaIPF not only achieves state-of-the-art results on a commonly-used dataset but also demonstrates superior performance on a highly dense and large-scale dataset, indicating its capability to address denoising challenges in extensive scenes, as validated through ablation experiments. Although 3DMambaIPF may encounter reverse-amplification issues on denoising a smooth and clean surface, its performance remains optimal. In future work, we will explore different datasets to address the challenges of filtering real-world point clouds from large scenes.

## Acknowledgments

This project was partially supported by the National Natural Science Foundation of China (U2033209); Hong Kong Innovation and Technology Fund (PiH/213/24GS); the Ministry of Education, Singapore, under its Academic Research Fund Grants (MOE-T2EP20220-0005 & RT19/22) and an OPPO gift fund.

## References

- Bangaru, S. P.; Gharbi, M.; Luan, F.; Li, T.-M.; Sunkavalli, K.; Hasan, M.; Bi, S.; Xu, Z.; Bernstein, G.; and Durand, F. 2022. Differentiable rendering of neural sdfs through reparameterization. In *SIGGRAPH Asia 2022 Conference Papers*, 1–9.
- Curless, B.; and Levoy, M. 1996. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, 303–312.
- 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.
- Fei, B.; Xu, J.; Zhang, R.; Zhou, Q.; Yang, W.; and He, Y. 2024a. 3D Gaussian Splatting as New Era: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, 1–20.
- Fei, B.; Yang, W.; Chen, W.-M.; Li, Z.; Li, Y.; Ma, T.; Hu, X.; and Ma, L. 2022. Comprehensive review of deep learning-based 3d point cloud completion processing and analysis. *IEEE Transactions on Intelligent Transportation Systems*, 23(12): 22862–22883.
- Fei, B.; Yang, W.; Liu, L.; Luo, T.; Zhang, R.; Li, Y.; and He, Y. 2023. Self-supervised learning for pre-training 3d point clouds: A survey. *arXiv preprint arXiv:2305.04691*.
- Fei, B.; Zhang, R.; Yang, W.; Li, Z.; and Chen, W.-M. 2024b. Progressive Growth for Point Cloud Completion by Surface-Projection Optimization. *IEEE Transactions on Intelligent Vehicles*.
- Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.
- Gu, A.; Goel, K.; and Ré, C. 2021. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*.
- Guerrero, P.; Kleiman, Y.; Ovsjanikov, M.; and Mitra, N. J. 2018. Pcpnet learning local shape properties from raw point clouds. In *Computer Graphics Forum*, volume 37, 75–85. Wiley Online Library.
- He, W.; Han, K.; Tang, Y.; Wang, C.; Yang, Y.; Guo, T.; and Wang, Y. 2024. DenseMamba: State Space Models with Dense Hidden Connection for Efficient Large Language Models. *arXiv preprint arXiv:2403.00818*.
- Insafutdinov, E.; and Dosovitskiy, A. 2018. Unsupervised learning of shape and pose with differentiable point clouds. *Advances in Neural Information Processing Systems*, 31.
- Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4): 1–14.
- Krishnamurthy, V.; and Levoy, M. 1996. Fitting smooth surfaces to dense polygon meshes. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, 313–324.
- Laine, S.; Hellsten, J.; Karras, T.; Seol, Y.; Lehtinen, J.; and Aila, T. 2020. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics (ToG)*, 39(6): 1–14.
- Liang, D.; Zhou, X.; Wang, X.; Zhu, X.; Xu, W.; Zou, Z.; Ye, X.; and Bai, X. 2024. PointMamba: A Simple State Space Model for Point Cloud Analysis. *arXiv preprint arXiv:2402.10739*.
- Liao, W.; Zhu, Y.; Wang, X.; Pan, C.; Wang, Y.; and Ma, L. 2024. LightM-UNet: Mamba Assists in Lightweight UNet for Medical Image Segmentation. *arXiv preprint arXiv:2403.05246*.
- Liu, S.; Li, T.; Chen, W.; and Li, H. 2019. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7708–7717.
- Lu, C.; Schroecker, Y.; Gu, A.; Parisotto, E.; Foerster, J.; Singh, S.; and Behbahani, F. 2024. Structured state space models for in-context reinforcement learning. *Advances in Neural Information Processing Systems*, 36.
- 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.
- Mao, A.; Du, Z.; Wen, Y.-H.; Xuan, J.; and Liu, Y.-J. 2022. Pd-flow: A point cloud denoising framework with normalizing flows. In *European Conference on Computer Vision*, 398–415. Springer.
- Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.
- Misra, D.; Gala, J.; and Orvieto, A. 2024. On the low-shot transferability of [V]-Mamba. *arXiv preprint arXiv:2403.10696*.
- Müller, J. U.; Weinmann, M.; and Klein, R. 2022. Unbiased Gradient Estimation for Differentiable Surface Splatting via Poisson Sampling. In *European Conference on Computer Vision*, 281–299. Springer.
- Peterson, L. E. 2009. K-nearest neighbor. *Scholarpedia*, 4(2): 1883.
- Pistilli, F.; Fracastoro, G.; Valsesia, D.; and Magli, E. 2020. Learning graph-convolutional representations for point cloud denoising. In *European Conference on Computer Vision*, 103–118. Springer.

- 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.
- Ren, Y.; Zhang, T.; Pollefeys, M.; Süssstrunk, S.; and Wang, F. 2023. Volrecon: Volume rendering of signed ray distance functions for generalizable multi-view reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16685–16695.
- Vicini, D.; Speierer, S.; and Jakob, W. 2022. Differentiable signed distance function rendering. *ACM Transactions on Graphics (ToG)*, 41(4): 1–18.
- Yang, Z.; Mitra, A.; Kwon, S.; and Yu, H. 2024. ClinicalMamba: A Generative Clinical Language Model on Longitudinal Clinical Notes. *arXiv preprint arXiv:2403.05795*.
- Yifan, W.; Serena, F.; Wu, S.; Öztireli, C.; and Sorkine-Hornung, O. 2019. Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics (ToG)*, 38(6): 1–14.
- 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. Point-filter: Point cloud filtering via encoder-decoder modeling. *IEEE Transactions on Visualization and Computer Graphics*, 27(3): 2015–2027.
- Zhao, H.; Zhang, M.; Zhao, W.; Ding, P.; Huang, S.; and Wang, D. 2024. Cobra: Extending Mamba to Multi-Modal Large Language Model for Efficient Inference. *arXiv preprint arXiv:2403.14520*.
- Zhou, J.; Jin, W.; Wang, M.; Liu, X.; Li, Z.; and Liu, Z. 2022. Fast and accurate normal estimation for point clouds via patch stitching. *Computer-Aided Design*, 142: 103121.