

PD-Refiner: An Underlying Surface Inheritance Refiner with Adaptive Edge-Aware Supervision for Point Cloud Denoising

Chengwei Zhang*

National Key Laboratory of
Microwave Imaging, Aerospace
Information Research Institute,
Chinese Academy of Sciences
Beijing, China

Xueyi Zhang*

Laboratory for Big Data and Decision,
National University of Defense
Technology
Changsha, Hunan, China

Xianghu Yue

Department of Electrical and
Computer and Engineering, National
University of Singapore
Singapore, Singapore

Mingrui Lao

National Key Laboratory of
Information Systems Engineering,
National University of Defense
Technology
Changsha, Hunan, China

Tao Jiang

National Key Laboratory of
Microwave Imaging, Aerospace
Information Research Institute,
Chinese Academy of Sciences
Beijing, China

Jiawei Wang

School of Software Engineering,
Tongji University
Shanghai, China

Fubo Zhang

National Key Laboratory of
Microwave Imaging, Aerospace
Information Research Institute,
Chinese Academy of Sciences
Beijing, China

Longyong Chen[†]

National Key Laboratory of
Microwave Imaging, Aerospace
Information Research Institute,
Chinese Academy of Sciences
Beijing, China

Abstract

Point clouds from real-world scenarios inevitably contain complex noise, significantly impairing the accuracy of downstream tasks. To tackle this challenge, cascading encoder-decoder architecture has become a conventional technical route to iterative denoise. However, circularly feeding the output of denoiser as its input again involves the re-extraction of underlying surface, leading to unstable denoising process and over-smoothed geometric details. To address these issues, we propose a novel denoising paradigm dubbed PD-Refiner that employs a single encoder to model the underlying surface. Then, we leverage several lightweight hierarchical Underlying Surface Inheritance Refiners (USIRs) to inherit and strengthen it, thereby avoiding the re-extraction from the intermediate point cloud. Furthermore, we design adaptive edge-aware supervision to improve the edge awareness of the USIRs, allowing for the adjustment of the denoising preferences from global structure to local details. The results demonstrate that our method not only achieves state-of-the-art performance in terms of denoising stability and efficacy, but also enhances edge clarity and point cloud uniformity.

*These authors contributed equally to this work.

[†]Corresponding author: chenly@aircas.ac.cn

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MM '24, October 28-November 1, 2024, Melbourne, VIC, Australia

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0686-8/24/10...\$15.00
<https://doi.org/10.1145/3664647.3681540>

CCS Concepts

• **Computing methodologies** → **Point-based models**.

Keywords

Point Cloud Denoising, Point Cloud Processing, Geometry Detail Recovery, Edge-Aware Supervision

ACM Reference Format:

Chengwei Zhang, Xueyi Zhang, Xianghu Yue, Mingrui Lao, Tao Jiang, Jiawei Wang, Fubo Zhang, and Longyong Chen. 2024. PD-Refiner: An Underlying Surface Inheritance Refiner with Adaptive Edge-Aware Supervision for Point Cloud Denoising. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM '24)*, October 28-November 1, 2024, Melbourne, VIC, Australia. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3664647.3681540>

1 Introduction

As a fundamental representation of 3D geometry, point cloud can be acquired through various methods, including LiDAR [18, 24, 59], millimeter wave radar [36, 42, 44, 51], and synthetic aperture radar [37, 62, 63]. It plays a pivotal role in the realm of 3D vision, finding applications in diverse areas, such as object detection [7, 48, 49], classification [6, 16, 20], segmentation [22, 66, 68], and surface reconstruction [32, 33, 47]. It can also integrate with other modalities [15, 25, 60, 64, 70] and apply across various domains, including autonomous driving [4, 18] and urban modeling [12, 35, 54]. Nevertheless, owing to sensor limitations and environmental factors, collected point clouds often contain intricate noise, which will significantly impacts downstream tasks' accuracy. Consequently, the development of point cloud denoising algorithms to mitigate noise is a critical research area.

Current methods [5, 26, 28, 40] typically perform denoising using single or cascaded structurally identical denoisers in both the training or testing phases. As illustrated in Fig. 1(a), single-step denoiser like GPDNet [40] employs a pair of encoder and decoder for denoising. On the one hand, the encoder’s role is to learn the relationship between each point and its neighbors in a noisy point cloud, thereby extracting the underlying surface representations. On the other hand, the decoder aims to regress the position displacement vector based on its representation to update point position. To further remove residual noise, iterative denoising methods (e.g., Score [28], PSR [2], IPFN [5]), as in Fig. 1(b), involve iterating single-step methods during testing or training phases. The intermediate point cloud with residual noise from the previous denoiser will serve as the input to the next denoiser for further denoising.

Although previous methods remove point cloud noise to some extent, there are certain limitations in these methods, obstructing the further improvements of denoising capabilities. **Firstly, re-extracting representations from intermediate point clouds may destabilize the denoising process.** In the high noise and low density point clouds, it is challenging to accurately estimate the underlying surfaces with a single denoiser, resulting in inevitable residual noise and edge over-smoothness. Subsequent iterations intensify these issues, as the re-extracted underlying surfaces lead to points converging towards a biased surface, ultimately causing an unstable denoising process and loss of geometric detail. **Secondly, existing methods typically lack accurate and adaptive supervision that adjusts the intensity of edge awareness according to the denoising stages.** In the initial denoising process, it is common for denoisers to prioritize non-edge regions due to their simpler geometry and lower noise sensitivity, while edge regions are challenging to restore with limited prior knowledge. Then, noise reduction in non-edge areas makes edge regions more discernible, enabling the recovery of their geometric details. However, without adaptive intensity of edge awareness, there might be an inappropriate focus on the difficult-to-recover edge regions in the early stages. Such undesired behaviour hampers the network’s stability and convergence, making it less effective at accurately preserving geometric details during the denoising process.

This paper introduces PD-Refiner, as in Fig. 1(c), to address the aforementioned limitations of previous methods. PD-Refiner aims to improve denoising stability by inheriting and strengthening the underlying surface representation from the prior denoising step, thus avoiding the re-extraction of representations from the intermediate point cloud. Specifically, PD-Refiner uses a single encoder to extract representation from the original noisy point cloud. Then, it employs hierarchical USIRs to inherit and strengthen the point-wise representation by synchronously considering the moving trend of the point and the multi-scale neighborhood modeling. This approach ensures that USIRs at different stages utilize a stable and increasingly refined underlying surface estimation for updating point positions. Consequently, this stable denoising process alleviates the denoisers’ learning burden of aligning points to the underlying surface, which, in turn, allows more learning capacity to be devoted to improving the quality of the point cloud, such as enhancing uniformity and restoring edges. Additionally, to more precisely guide the USIRs in edge restoration, we design a novel adaptive edge-aware supervision strategy named Adaptive Shape

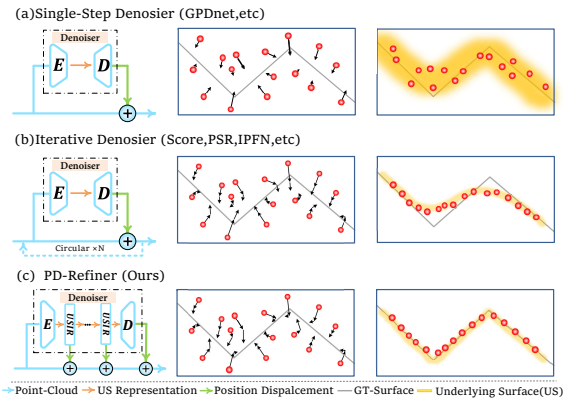


Figure 1: Comparisons of PD-Refiner with previous single-step and iterative denoising methods.

Preserving Loss (ASPL) to gradually recover geometric details. ASPL adjusts the intensity of edge awareness within the optimization targets of each USIR layer based on the denoising step number. It guides the shallow USIRs to prioritize denoising the global non-edge point (plane) and gradually provides stronger supervision signals to the edge points (local geometric detail) in deeper USIRs. Feature extraction is performed only once in the entire denoising process. Additionally, benefiting from the single-layer point convolution, the USIR architecture is more lightweight compared to traditional encoders, thereby requiring fewer computing resources than conventional stacked encoder-decoder denoising methods. The main contributions of this paper are:

- We propose PD-Refiner, a novel yet practical point cloud denoising framework that boosts denoising stability by inheriting and strengthening the underlying surface representation without re-extracting it from intermediate point clouds. This methodology ensures a more stable denoising process by maintaining a consistent representation across denoising steps.
- We establish a stable denoising process that promotes the network to focus more on restoring geometric details. To this end, we introduce the Adaptive Shape Preserving Loss (ASPL), an adaptive supervision strategy that precisely adjusts the edge-awareness intensity of each USIR layer, ensuring detailed and accurate edge restoration.
- Extensive experiments demonstrate that PD-Refiner achieves superior performance compared with previous SOTA iterative methods in both testing or training phases with demanding fewer computational resources. It is noteworthy that, it also benefits the promoting of denoising quality, including the local geometric details and the uniformity.

2 Related Work

Deep learning driven point cloud denoising methods have demonstrated remarkable denoising capabilities in various noise modalities [8–11, 14, 21, 23, 34, 65]. The rapid progress of point cloud analysis has led to the development of various neural network architectures and operators tailored to processing point cloud data such as PointNet [43], DGCNN [56], PointTransformer [67], etc. Deep learning driven point cloud denoising methods can be broadly categorized into displacement-based and generation-based approaches.

Displacement-based methods like GPDNet [40, 41] utilizes graph-convolutional layers [50] as encoder to elegantly addresses the permutation invariance problem commonly encountered in learning-based point cloud denoising methods. Noise2noise [30] proposes to learn Signed Distance Functions (SDF) [29] through noise-to-noise mapping and update the point position.

Iterative denoising plays a crucial role in high noise point cloud. Conventional iterative denoising methods train single denoiser and conduct testing phase multiple times. PathNet [57] designs dynamic network that adjust the denoising path according to geometry and noise level of each point. RePCD [3] proposes to use recurrent neural networks (RNNs) to model the iterative denoising process. IPFN [5] achieves iterative denoising by stacking multiple IterationModules during training. These methods [3, 5] directly add the position displacement to the coordinates of the point to denoise. There are also methods to introduce step size to control the movement of points in iterative denoising. They model the gradients of point cloud distribution to reduce noise through gradient ascent. Score [28] introduces a gradient ascent method to map 3D distribution to a 2D manifold and utilize Langevin sampling [1, 27] to iteratively recover low-noise data from high-noise input. PSR [2] introduced a gradient-based point set resampling that continuously model the probability distribution density of the noisy point cloud, and resample this distribution to obtain the denoised point cloud. MAG [69] introduced a momentum term in the gradient update process to deal with fluctuate gradient field. *Although iterative denoising has achieved a certain effect, these methods generate the intermediate point cloud and need the re-extraction of underlying surfaces, causing the unstable denoising process. Our method falls into displacement-based method, but unlike the previous approach, we model the representation of the point cloud for once, instead of re-extracting from the output point cloud from the previous step.*

Generation-based methods focus on removing high-noise points and generating low-noise points. DMR [26] estimates the point-wise noise level as a measure of point-wise quality. They select the part with the highest quality and generate new points to restore the original point count. Random screen [55] proposes integrating points into higher-dimensional sub-patches to reduce the number of points and denoise, then decoupling them to recover the original number of points. SSPCN [19] suggests extracting features from the down-sampled points for feature compensation, alleviating the inconsistency between point features and their actual locations. *While this method has advantages in dealing with outliers, the downsampling process results in the loss of geometric details. On the contrary, our method could maintain more detailed geometric information without the points sampling.*

3 Methods

3.1 Preliminary

Following previous state-of-the-art methods [5, 26, 28], we start with a clean point cloud $PCD_{clean} \in \mathbb{R}^{M \times 3}$, which is normalized to the unit sphere and M is the point number of the object. An additive 3D Gaussian noise $\xi \sim N(0, \sigma)$ is added to the clean point cloud to get a noisy point cloud $PCD_{noisy} \in \mathbb{R}^{M \times 3}$, where $PCD_{noisy} = PCD_{clean} + \xi$. The noise intensity is controlled by the variance σ . To create paired training data, we randomly select a reference point

x_r from PCD_{noisy} , and choose the nearest N points, where N is the patch size, from PCD_{noisy} and PCD_{clean} to get P_n as the input noisy point cloud and P_c as the ground truth clean point cloud.

Displacement-based point cloud denoising networks commonly utilize an encoder, to extract point-wise underlying surface representations based on the noisy points 3D coordinates. This process can be formulated as:

$$F_n = Encoder(P_n, \theta^E), \quad (1)$$

where $F_n \in \mathbb{R}^{N \times C}$ is the extracted point-wise underlying surface representation and C is the feature dimension. The encoder is built upon Dynamic EdgeConv layers [5, 50]. The update of point-wise features is formulated as:

$$h_i^{s+1} = (f_\phi(h_i^s) \parallel \sum_{j:(i,j) \in \epsilon} g_\theta(h_j^s \parallel h_j^s - h_i^s)), \quad (2)$$

where h_i^s and h_j^s represent the feature of the center point and its neighborhood points in layer s , ϵ is the edges, $f_\phi(\cdot)$ and $g_\theta(\cdot)$ are parameterized by MLPs, and (\parallel) denotes concatenate.

The i -th point-wise representation encodes information of the underlying surface formed by the neighborhood of i -th points and the relationship between i -th point and the underlying surface. A decoder, composed of MLPs, is utilized to map the relationship in the representation into position displacement vector, updating the coordinates accordingly to bring the point closer to the underlying surface. This process can be formulated as:

$$p^{disp} = Decoder(F_n, \theta^D), \quad (3)$$

where $p^{disp} \in \mathbb{R}^{N \times 3}$ is the predicted position displacement vector. The final denoised point cloud is obtained as $P_d = P_n + p^{disp}$. The training objective of the denoising network is to minimize the Chamfer Distance (CD) between the denoised point cloud P_d and the clean point cloud P_c .

Iterative denoising is a commonly-used process to improve denoising performance, particularly for high-noise point clouds. It involves stacking multiple denoising networks formulated as:

$$F_l = Encoder(P_l^m, \theta_l^E) \\ p_{l+1}^m = P_l^m + Decoder(F_l, \theta_l^D), \quad (4)$$

where τ is the iteration number and $0 \leq l < \tau$, P_l^m is the intermediate point clouds, $P_0^m = P_n$ representing the input noisy point cloud, and $P_d = P_\tau^m$ is the output denoised point cloud. $F_0 = F_n$ is the extracted original underlying surface representation from P_n . In testing phase iterative denoising [2, 28, 69], they use the same network where $\theta_i^u = \theta_j^u$, $u \in \{E, D\}$, $0 \leq i, j < \tau$. However, taking the intermediate point cloud with different residual noise as input of the same network increases the burden of learning point cloud distribution. To alleviate this problem, IPFN [5] propose iterative denoising in training phase to stage the iterative denoising process. Nevertheless, these iterative denoising methods need re-extracting representations from intermediate point clouds, resulting in the unstable denoising process. To address these limitations of iterative denoising, we introduce PD-Refiner as shown in Fig. 2, consisting of the Underlying Surface Inheritance Refiner (USIR), Shape Variance Weighting Network (SVWN) and adaptive edge-aware supervision strategy named Adaptive Shape Preserving Loss (ASPL).

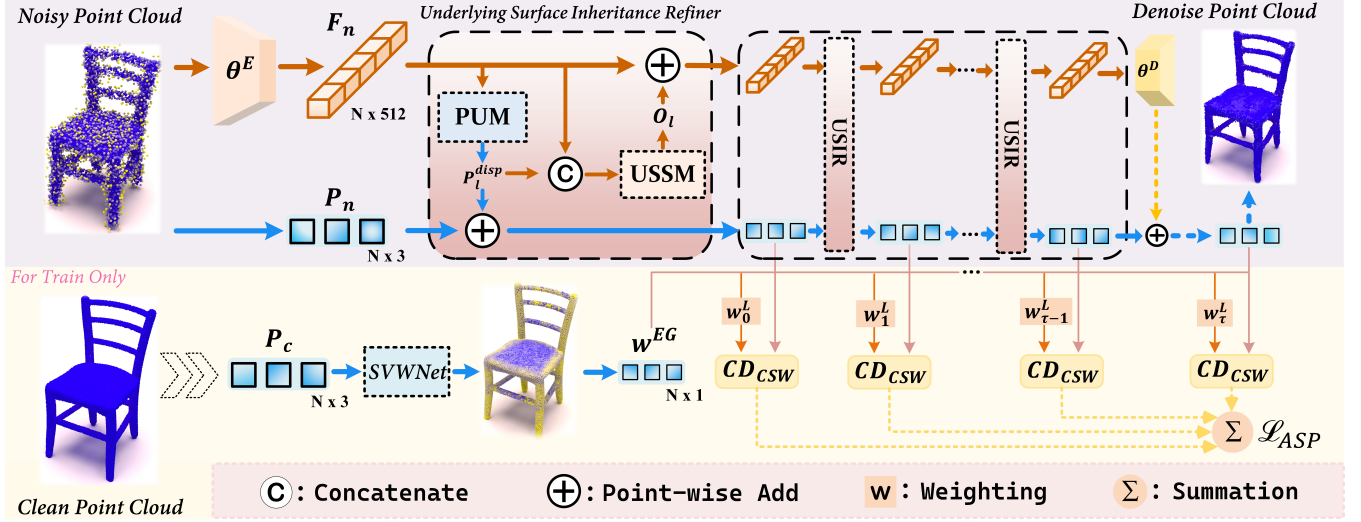


Figure 2: Overview of the proposed PD-Refiner. Our PD-Refiner employs a single encoder along with a hierarchical Underlying Surface Inheritance Refiner (USIR) to strengthen the inherited underlying surface representation, ensuring a stable and accurate denoising process. We incorporate an edge-aware supervision strategy, called Adaptive Shape Preserving Loss (ASPL), to restore geometric details in the denoised point cloud.

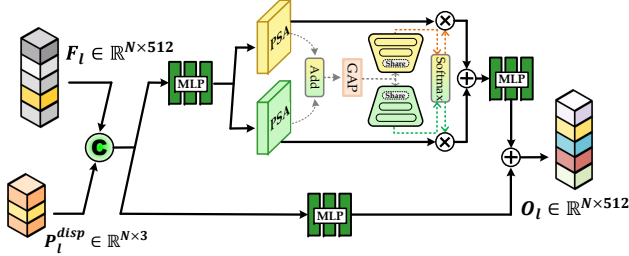


Figure 3: Details of the USSM. The point-wise representation and position displacement vector is synchronously considering for multi-scale modeling.

3.2 Underlying Surface Inheritance Refiner

Previous iterative denoising consists of a stack of encoder-decoder pairs, where each pair represents a denoiser. Unlike the network structure of previous iterative methods, PD-Refiner employs a single pair of encoder and decoder. We insert hierarchical USIRs between them to inherit and strengthen the underlying surface representation while update the point position.

First, we utilize a MLPs-based Position Update Module (PUM) to compute a positional displacement vector P_l^{disp} using F_l as input, enabling point position updates formed as $P_{l+1}^m = P_l^m + P_l^{disp}$, where l signifies the layer number, and F_l is the input representation of the l -th USIR. Then, we propose the Underlying Surface Strengthen Module (USSM), as shown in Fig. 3., that leveraging multi-scale neighborhood modeling to strengthen the estimation of the underlying surface. Benefited from the commonality of underlying surfaces described by neighboring points, this method combines multiple estimates and further achieves more robust and reliable results, thereby enhancing both the overall stability and accuracy of

the denoising process. Simultaneously, the positional displacement vector contains complementary information about point movement trend, also offering valuable guidance for estimating the underlying surface’s location.

The input of USSM is formed as $G_l = (P_l^{disp} || F_l)$, where $G_l \in \mathbb{R}^{N \times (512+3)}$. Multi-scale neighborhood representation modeling is enabled by leveraging the selective kernel operation [17, 38, 39], specialized for dynamically selecting and aggregating convolutional kernels of various receptive field. The local geometry structure and noise distribution characteristics of point clouds present different scale requirements for underlying surface modeling. USSM adaptively fuses multi-scale underlying surface information learned from point moving trends and neighborhood representations to strengthen the underlying surface representation and formulated as $O_l = USSM(G_l)$ and $F_{l+1} = F_l + O_l$.

At the beginning of USSM, a MLP layer is used to adjust the feature dimension and get $G_l^a \in \mathbb{R}^{N \times 512}$. Multi-scale feature is extracted by Point Self-Attention (PSA) [52] layer to get multi-scale information formulated as:

$$H_l^a = \{PSA(G_l^a, k_1), \dots, PSA(G_l^a, k_n)\}, \quad (5)$$

where $k_o, 1 \leq o \leq n$ is the receptive fields. Considering the balance of computation and precision, we set $\{k_o\}_{o=1}^n = \{8, 16, 32\}$. These multi-scale features are added together and get the joint representation by global average pooling (GAP) through all points. Then, we utilize a MLP layer to squeeze the feature dimension to obtain the compressed mixed-scale feature $S_l^a \in \mathbb{R}^{N \times 128}$. The feature dimension is restored by multiple MLPs and use SoftMax to calculate the corresponding weights of each feature at different scale. We perform weighted summations of the multi-scale feature, and feed it into the MLP to get $O_l^a \in \mathbb{R}^{N \times 512}$. Residual connection is achieved by using an MLPs to get the residual feature $O_l^b \in \mathbb{R}^{N \times 512}$. The multi-scale feature and the residual feature is added as $O_l = O_l^a + O_l^b$.

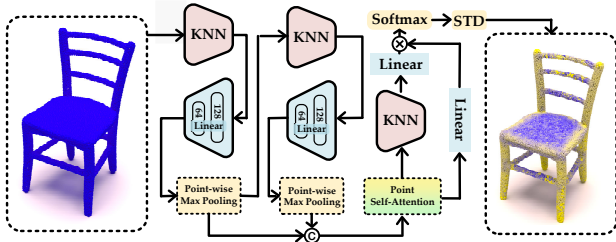


Figure 4: Details of SVWN. High-weighted edge and low-weighted plane points are marked in yellow and blue.

3.3 Shape Variance Weighting Network

Implementing ASPL requires knowing whether each point belongs to the edge region. We design a Shape Variance Weighting Network (SVWN), as in Fig. 4, and introduces attention-based edge point learning strategy [58] that leveraging statistical characteristics to extract edge weight from the clean point cloud patch P_c . SVWN calculates the variance of the attention-based correlation map within the neighborhood points feature. This calculation captures the degree of geometric structure change. It represents the probability that a point belongs to the edge region.

To be specific, SVWN construct dynamic graph structures [56] according to the feature similarity among points by KNN. MLP layers is used for point relationship learning. We use two dynamic graph convolution layers to encourage the network to organize the graph semantically and expands the receptive field of local neighborhood to better capture edge feature. The intermediate features from the pooling layer are concatenated and subsequently passed through a PSA layer to get the edge feature $F^{Edge} \in R^{N \times 128}$. KNN generates the edge feature difference F_{diff}^{Edge} between the central point and its neighboring points. These features, F^{Edge} and F_{diff}^{Edge} , respectively serve as the inputs for two MLPs, producing $F^{Query} \in R^{N \times 1 \times 128}$ and $F^{Key} \in R^{N \times 128 \times 32}$. The variance of attention-based correlation map is formed as $w^{Edge} = STD(Softmax(F^{Query} \times F^{Key}))$. A higher variance indicates a more pronounced change in the local geometry structure, signifying its probability of proximity to the edge. Since the edge of input patch does not belong to the real edge of the object, Gaussian weights is added to retain more accurate edge annotation. The final edge weight is formulated as:

$$w_i^{EG} = w_i^{Edge} * \frac{\exp(-\|x_i - x_r\|_2^2 / r_s^2)}{\sum_i \exp(-\|x_i - x_r\|_2^2 / r_s^2)}, \quad (6)$$

where $r_s = r/3$ and r is the patch radius. The parameters of SVWN is updated through back propagation together with USIRs, without additional GT geometrical loss supervision like local curvatures.

3.4 Adaptive Shape Preserving Loss

To improve PD-Refiner’s structural recovery, we propose Adaptive Shape Preserving Loss (ASPL), which adjusts the intensity of edge awareness within the optimization targets of each USIR layer based on the denoising step number to guide points precisely converge to the correct surface.

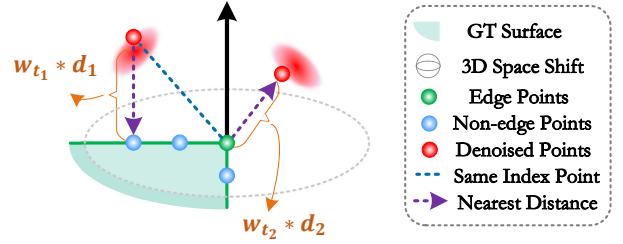


Figure 5: Illustration of adding edge weights to different loss items. If add weights to both items w_{t1} and w_{t2} , the supervision signal of the edge region might leak into the non-edge region.

The Chamfer Distance (CD) loss is wildly used for supervised point cloud denoising, which can be formulated as:

$$CD(P_d, P_c) = \frac{1}{N} \left(\sum_{a \in P_d} \min_{b \in P_c} \|a - b\|^2 + \sum_{b \in P_c} \min_{a \in P_d} \|a - b\|^2 \right), \quad (7)$$

where $\|a - b\|^2$ represents the Euclidean distance between the two points. Standard CD loss consists of two terms. The first term calculates the average nearest distance from each point in the denoised point cloud P_d to the clean point cloud P_c , and the second term does the same for each point in P_c to P_d . Within each term, every point’s nearest distance contributes equally to the overall loss.

The edge weights extracted by SVWN can provide index-level edge weight annotation to P_c . Previous method [3] add weights to both terms (denoised-clean and clean-denoised) of CD loss according to point index. Considering the edge points in P_c may shift to non-edge region due to added noise and denoising process, even though they have the same index, as shown in Fig. 5, the high weight to the edge region may leak to the non-edge region thus providing imprecise supervision signal of the edge region. To overcome this limitation, we propose an improved CD loss named Clean-denoised Shape Weighted CD formulated as:

$$CD_{csw}(P_d, P_c, w^L) = \frac{1}{N} \left(\sum_{a \in P_d} \min_{b \in P_c} \|a - b\|^2 + \sum_{b \in P_c} \min_{a \in P_d} (1 + w_i^{EG} * w^L) * \|a - b\|^2 \right), \quad (8)$$

where only the loss term of the nearest distance from clean to denoised point cloud is added the edge weights. Since P_c is not affected by noise addition or denoising, we can provide a more accurate supervision signal to edge region, and w^L is used to control the intensity of edge awareness according to denoising stages.

To gradually increase the overall focus on the edge regions of USIR, we set different layer weights for the aggregation of losses at each layer. The layer weight is increasing with the number of USIR. The final ASPL is formulated as:

$$\mathcal{L}_{ASP} = \sum_{l=1}^{\tau} CD_{csw}(P_{m,l}, P_c, w_l^L), \quad (9)$$

where $w_l^L = \frac{\exp(l)}{\exp(\tau)}$ represents the layer weight.

#Points		10K Points						50K Points					
Noise Level		1% Noise		2% Noise		3% Noise		1% Noise		2% Noise		3% Noise	
Method	Venue	CD↓	P2M↓	CD↓	P2M↓	CD↓	P2M↓	CD↓	P2M↓	CD↓	P2M↓	CD↓	P2M↓
PCN [45]	CGF 19	35.15	11.48	74.67	39.65	130.67	87.37	10.49	3.46	14.47	6.08	22.89	12.85
GDPNet [40]	ECCV 20	37.80	13.37	80.07	44.26	134.82	91.14	19.13	10.37	50.21	37.36	97.05	79.98
DMR [26]	ACM MM 20	44.82	17.22	49.82	21.15	58.92	28.46	11.62	4.69	15.66	8.00	24.32	15.28
Score [28]	ICCV 21	25.21	4.63	36.86	10.74	47.08	19.42	7.16	1.50	12.88	5.66	19.28	10.41
PD-Flow [31]	ECCV 22	21.26	3.81	32.46	10.10	44.47	19.99	6.51	1.64	11.73	5.81	19.14	12.10
PSR [2]	TPAMI 22	23.53	3.06	33.50	7.34	40.75	12.42	6.49	0.76	9.97	2.96	<u>13.44</u>	5.31
Noise2Noise [30]	ICML 23	10.60	2.41	29.25	10.10	42.21	18.47	3.77	1.55	10.29	4.84	16.54	9.72
IPFN [5]	CVPR 23	20.56	2.18	30.43	5.55	42.41	13.76	6.05	0.59	8.03	<u>1.82</u>	19.71	10.12
PathNet [57]	TPAMI 24	26.72	5.84	39.73	12.99	45.24	24.04	7.16	1.24	11.40	4.10	18.75	9.52
Ours _{tiny}		18.27	<u>1.82</u>	<u>25.87</u>	<u>4.89</u>	<u>33.74</u>	<u>10.45</u>	4.93	<u>0.49</u>	<u>7.15</u>	1.85	14.58	7.22
Ours		<u>17.54</u>	1.66	24.44	4.49	30.77	9.13	<u>4.66</u>	0.45	6.53	1.64	12.28	<u>5.75</u>

Table 1: Denoising comparison with SOTA methods in PU-Net dataset under Gaussian noise. CD is multiplied by 10^5 and P2M is multiplied with 10^5 . Data in bold and underline represent the best and second best result among all methods.

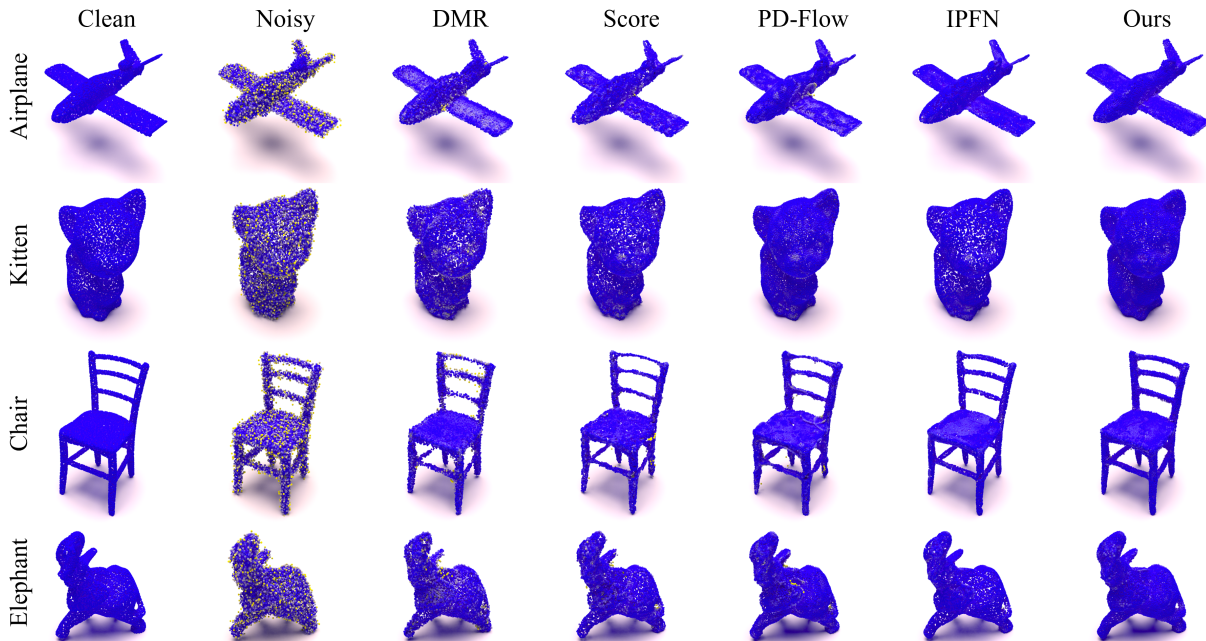


Figure 6: Visual result of point-wise P2M distance for 10K resolution with Gaussian noise of 2% of the bounding sphere radius.

4 Experiment

4.1 Dataset and Implementation Details

Dataset: We follow the training strategy of the previous methods that selects 40 meshes from the training set of PU-Net [61] for the network training. The training sample uses Poisson disk sampling with a resolution of 10k, 30k, 50k to obtain the point cloud, resulting in a total of 120 different training samples. We add Gaussian noise of 0.5% to 2% of bounding sphere’s radius variance for each sample. The patch size N of the training sample pair is set to 1000. The test set consist of 20 samples from the PU-Net test set with resolutions of 10k and 50k and noise intensity of 1%, 2% and 3%. CD and P2M are used to measure all method performance.

Furthermore, we conduct qualitative study on the real-world dataset Paris-Rue-Madame [46], which comprises scans obtained

through the utilization of the Mobile Laser Scanning system L3D2. Notably, this dataset exhibits real-world imperfections introduced by the inherit limitations of the scanning technology, making it an ideal benchmark for assessing the algorithm’s robustness in the face of such artifacts. Next, we consider the Kinect v1 and Kinect v2 datasets [53], consisting of 71 and 72 real-world scans acquired using Microsoft Kinect v1 and Kinect v2 cameras.

Implementation details: We use the network structure PyTorch implementation of IPFN [5] and set the IterationModule number to 1 as our baseline. The proposed PD-Refiner is trained and tested on NVIDIA 3090 GPUs using PyTorch 1.9.0 with CUDA 11.1. We train the network for 100 epochs, with the Adam [13] optimizer. We used the cosine learning rate scheduler to smoothly decrease the learning rate from an initial value of 1×10^{-4} to a final value of 1×10^{-7} . Batch size is set to 16.

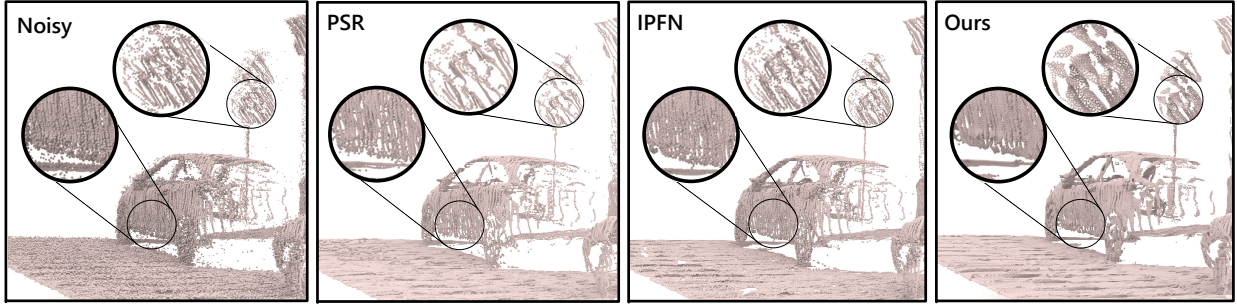


Figure 7: Visual result of the real-world RueMadame dataset, showcasing the superior point uniformity and shape preservation ability of our method.

#Points		10K Points					
Noise		1% noise		2% noise		3% noise	
Type	Method	CD↓	P2M↓	CD↓	P2M↓	CD↓	P2M↓
Lap	PCN	46.16	19.40	110.82	72.18	209.81	159.22
	Score	29.15	6.74	46.01	17.99	63.32	32.71
	PSR	26.63	4.50	37.90	10.67	<u>51.10</u>	<u>20.17</u>
	IPFN	<u>24.25</u>	<u>3.28</u>	<u>34.66</u>	<u>8.68</u>	68.60	34.60
	Ours	20.62	2.66	28.96	7.94	42.48	17.93
Dis	PCN	11.77	3.07	28.70	8.71	40.28	16.74
	Score	12.49	2.51	21.77	4.16	26.53	6.53
	PSR	10.21	1.63	19.21	2.68	22.74	4.31
	IPFN	<u>6.76</u>	<u>0.91</u>	<u>16.60</u>	<u>1.88</u>	<u>20.29</u>	<u>3.21</u>
	Ours	6.13	0.75	15.34	1.50	19.49	2.56
Ani	PCN	34.32	11.29	73.93	39.40	129.52	86.54
	Score	24.70	4.56	36.82	10.84	47.76	20.00
	PSR	23.05	3.08	33.45	7.58	<u>41.52</u>	<u>13.50</u>
	IPFN	<u>20.25</u>	<u>2.24</u>	<u>30.59</u>	<u>5.89</u>	50.31	20.14
	Ours	17.41	1.76	24.70	4.80	33.43	11.28
Uni	PCN	12.05	3.37	33.78	10.18	50.44	19.95
	Score	12.77	2.48	24.67	4.18	30.79	6.54
	PSR	10.56	1.64	23.48	2.75	29.16	4.43
	IPFN	<u>6.78</u>	<u>0.91</u>	<u>20.38</u>	<u>2.03</u>	<u>26.90</u>	<u>3.53</u>
	Ours	6.12	0.76	17.11	1.52	21.65	2.62

Table 2: Denoising comparison with SOTA methods in other noise distribution.

4.2 Comparisons of the state-of-the-art

Tab. 1. shows the evaluation metrics comparison including CD and P2M under different noise level and point density. We run PathNet in PU-Net for 1x, 2x, and 4x testing phase denoising, under 1%, 2%, and 3% noise. Other methods were tested with default settings. We also offer a tiny version with $C = 256$ and $\tau = 4$. Our method significantly outperforms competitive methods in most settings. In particular, our method has proven to be more efficient, especially when encountering more complex, low-density point clouds with high noise levels, where the noise level far exceeds that encountered during training. This robustly demonstrates that PD-Refiner ensures a more stable denoising process and enhanced robustness.

The comparison of the visual results of the denoised point cloud from PU-Net obtained by the SOTA methods and PD-Refiner is shown in Fig. 6. The first two columns present the clean and noisy point clouds, serving as benchmarks for evaluating the effectiveness of various denoising methods. DMR shows significant loss

Method	V1		V2	
	CD↓	P2M↓	CD↓	P2M↓
Noisy	15.36	8.21	25.21	14.60
PSR	15.18	7.77	23.25	<u>12.74</u>
PD-Flow	14.25	7.68	22.56	12.93
Score	14.25	6.94	23.46	12.36
IPFN	14.92	7.99	<u>22.27</u>	12.89
Ours	<u>14.34</u>	<u>7.61</u>	22.26	12.89

Table 3: Denoising comparison with SOTA methods in Kinect V1 and V2 dataset.

of geometric details and fails to remove residual noise effectively. Score performs better in noise reduction but struggles with surface smoothness. PD-Flow ensures a more uniform point distribution but suffers from irregular surface undulations and misalignment. IPFN achieves commendable noise reduction but lacks uniformity in point cloud distribution. Compared with previous methods, PD-Refiner excels in both residual noise elimination and preservation of geometric details. This superiority is attributed to an innovative underlying surface inheritance and strengthening framework, coupled with an adaptive shape preserving loss. This denoising process not only stabilizes but also improves the structural integrity and uniformity of the point cloud distribution.

Furthermore, we also test under various noise modalities including anisotropic Gaussian, discrete, Laplace and uniform noise. As in Tab. 2, PD-Refiner can achieve the state-of-the-art performance in all noise modalities and noise levels. In the real-world RueMadame dataset, as in Fig. 7, our method achieves better preservation of geometric details and point uniformity. The quantitative comparison of Kinect V1 and V2 dataset, as in Tab. 3, also demonstrate the generalizability of PD-Refiner in real-world scenarios.

4.3 Ablation study

We conduct an ablation study to verify the effectiveness of each component, including: 1) the influence of the number of USIRs used in PD-Refiner; 2) the inputs and methods for strengthening the underlying surface representation in USIR; 3) stage adaptability and edge supervision strategies in ASPL; and 4) the computational resource requirements across different configurations.

The influence of the number of USIR used on PD-Refiner is studied in Tab. 4. We incorporate different numbers of USIRs into the baseline model and conduct multiple testing phase denoising. The results of 1x testing phase denoising indicate that adding more

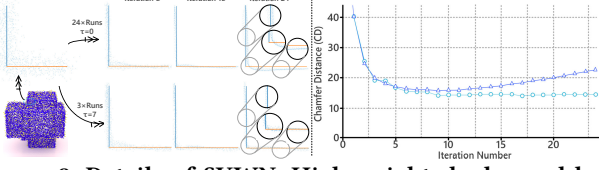


Figure 8: Details of SVWN. High-weighted edge and low-weighted plane points are marked in yellow and blue.

#Points		10K Points					
Noise Level		1% noise		2% noise		3% noise	
Runs	τ	CD↓	P2M↓	CD↓	P2M↓	CD↓	P2M↓
1×	0	19.91	2.68	29.05	6.60	43.74	17.30
	1	18.70	1.94	27.02	5.37	36.07	11.69
	3	18.00	1.73	25.57	4.77	32.84	9.86
	7	<u>17.54</u>	<u>1.66</u>	<u>24.44</u>	<u>4.49</u>	30.77	<u>9.13</u>
2×	3	17.72	1.70	24.51	4.54	<u>30.74</u>	9.33
	7	17.53	1.64	23.92	4.34	29.77	8.89

Table 4: Study of the influence of the number of USIR used on PD-Refiner. τ is the USIR number and we run multiple testing phase denoising.

#Points			10K Points					
Noise Level			1% noise		2% noise		3% noise	
F_l	P_l^{disp}	USSM	CD↓	P2M↓	CD↓	P2M↓	CD↓	P2M↓
✓	✓		19.39	2.39	28.37	6.23	43.23	16.95
	✓	✓	18.56	1.89	26.77	5.12	35.56	11.54
✓		✓	<u>17.86</u>	<u>1.70</u>	<u>25.04</u>	<u>4.64</u>	<u>31.93</u>	<u>9.54</u>
✓	✓	✓	<u>17.54</u>	<u>1.66</u>	<u>24.44</u>	<u>4.49</u>	<u>30.77</u>	<u>9.13</u>

Table 5: Study of the inputs and methods for strengthening underlying surface in USIR. If USSM is not used, MLP will be used instead.

USIRs can significantly improve the denoising performance, particularly in point clouds with high noise levels. This trend suggests that USIRs can stabilize the underlying surface modeling process in noisy conditions. Run 2× testing phase denoising can achieve a small accuracy improvement, but need more computational resources. Notably, the 1× testing phase denoising with $\tau = 7$ can outperform the 2× testing phase denoising with $\tau = 3$, even though they undergo the same number of point position updates, as the latter re-extracts the underlying surface using the intermediate point cloud, while the former maintains a stable estimation of the underlying surface. The visualization in Fig. 7 shows that frequently re-extracting the underlying surfaces causes the points to converge towards a biased surface.

The inputs and methods for strengthening underlying surface in USIR is studied in Tab. 5. Without multi-scale neighborhood modeling for underlying surface representation strengthening using USSM, PD-Refiner degrades similarly to the fixed path version of PathNet, and the precision almost drops to that of the baseline model, demonstrating the significance of underlying surface representation strengthening. Both F_l and P_l^{disp} can promote the underlying surface estimation, but the effect of P_l^{disp} is relatively minor due to its limited description of the underlying surface’s location. When combining F_l and P_l^{disp} , their complementary information provides a more effective depiction of the denoising process.

#Points	10K Points					
Noise Level	1% noise		2% noise		3% noise	
Loss type	CD↓	P2M↓	CD↓	P2M↓	CD↓	P2M↓
L2norm	18.97	1.91	26.75	4.91	35.06	10.89
CD	18.23	1.79	25.41	4.75	31.88	9.62
CD _{both-terms}	17.79	1.72	24.66	4.59	31.56	9.48
CD _{csw}	<u>17.73</u>	<u>1.69</u>	<u>24.59</u>	<u>4.56</u>	<u>31.33</u>	<u>9.42</u>
CD _{csw} + Adaptive	17.54	1.66	24.44	4.49	30.77	9.13

Table 6: Study of stage adaptability and edge supervision strategies in ASPL. We use different loss type for supervision. $CD_{both-terms}$ is to add weight on both terms of CD loss.

Method	PD-Flow	Score	PSR	PathNet	IPFN	Ours _{tiny}	Ours
Param (M)	0.460	0.187	0.267	10.860	3.196	1.946	15.108
FLOPs (G)	4.508	21.281	102.284	2067.594	69.650	8.860	61.098
Time (ms)	72.2	152.7	220.6	2239.1	165.1	48.0	110.5

Table 7: Compression of number of parameters (M), FLOPs (G) and inference time (ms).

The stage adaptability and edge supervision strategies in ASPL is studied in Tab. 6. Compared to the standard Chamfer Distance (CD) loss, the edge-weighted CD loss achieves better performance. This accuracy is further improved when precise guidance is applied using the proposed clean-denoised shape-weighted CD loss. By providing hierarchical optimization targets for refiners in each layer, our method take the first place on denoising performance. We also use the $L2_{norm}$ loss proposed by IPFN and Score for training, achieving significant improvements compared to IPFN. This indicates that the underlying surface inheritance and strengthening paradigm is more efficient than stacking IterationModules.

The training and testing computational resource requirements is studied in Tab. 7. In processing 1K points patch with single 3090 GPU, we achieved higher efficiency. We also offer a tiny version with $C = 256$ and $\tau = 4$, which significantly improves efficiency with minimal accuracy loss. PathNet [57] needs to denoise each point in the point cloud independently, resulting in higher computational resource requirements. When training with 2× 3090 GPUs, PD-Refiner with $\tau = 8$ requires 34.57 GiB of training memory and 0.55s per iteration, whereas IPFN [5] with "4 it." requires 35.62 GiB and 0.52s per iteration. This results demonstrate PD-Refiner’s relatively lower computational resource consumption.

5 Conclusion

In this paper, we propose a novel yet practical PD-Refiner for efficient point cloud denoising. Unlike previous iterative denoising methods, which replying on re-extract representations from intermediate point clouds, PD-Refiner inherits the underlying surface representation directly and leveraging USIRs to strengthen it by multi-scale neighborhood modeling. We also propose ASPL to provide USIRs with adaptive and more precise edges-aware supervision. Qualitative and quantitative results show that our method achieves excellent performance on both synthetic and real point cloud denoising benchmark.

Since we have found that the underlying surface representation inheritance and strengthening paradigm can make the denoising process more stable, we plan to extend our model to process data with more complex noise modalities, such as those from synthetic aperture radar and millimeter wave radar point clouds.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (Distributed Polarization 3D Imaging Radar System Technology, No. 2022YFB3901601).

References

- [1] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snively, and Bharath Hariharan. 2020. Learning gradient fields for shape generation. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 364–381.
- [2] Haolan Chen, Shitong Luo, Wei Hu, et al. 2022. Deep point set resampling via gradient fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 3 (2022), 2913–2930.
- [3] Honghua Chen, Zeyong Wei, Xianzhi Li, Yabin Xu, Mingqiang Wei, and Jun Wang. 2022. Repcd-net: Feature-aware recurrent point cloud denoising network. *International Journal of Computer Vision* 130, 3 (2022), 615–629.
- [4] Yaodong Cui, Ren Chen, Wenbo Chu, Long Chen, Daxin Tian, Ying Li, and Dongpu Cao. 2021. Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Transactions on Intelligent Transportation Systems* 23, 2 (2021), 722–739.
- [5] Dasith de Silva Edirimuni, Xuequan Lu, Zhiwen Shao, Gang Li, Antonio Robles-Kelly, and Ying He. 2023. IterativePFN: True iterative point cloud filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13530–13539.
- [6] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. 2021. Re-visiting point cloud shape classification with a simple and effective baseline. In *International Conference on Machine Learning*. PMLR, 3809–3820.
- [7] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. 2020. Structure aware single-stage 3d object detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11873–11882.
- [8] Robin Heinzler, Florian Piewak, Philipp Schindler, and Wilhelm Stork. 2020. Cnn-based lidar point cloud de-noising in adverse weather. *IEEE Robotics and Automation Letters* 5, 2 (2020), 2514–2521.
- [9] Qianjiang Hu and Wei Hu. 2022. Dynamic Point Cloud Denoising via Gradient Fields. *arXiv preprint arXiv:2204.08755* (2022).
- [10] Wei Hu, Qianjiang Hu, Zehua Wang, and Xiang Gao. 2021. Dynamic point cloud denoising via manifold-to-manifold distance. *IEEE Transactions on Image Processing* 30 (2021), 6168–6183.
- [11] Maryam Jameela and Gunho Sohn. 2024. NSANet: Noise Seeking Attention Network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* (2024).
- [12] Dušan Jovanović, Stevan Milovanov, Igor Ruskovski, Miro Govedarica, Dubravka Sladić, Aleksandra Radulović, and Vladimir Pajić. 2020. Building virtual 3D city model for smart cities applications: A case study on campus area of the university of novi sad. *ISPRS International Journal of Geo-Information* 9, 8 (2020), 476.
- [13] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [14] Maxime Kirgo, Guillaume Terrasse, Guillaume Thibault, and Maks Ovsjanikov. 2023. ReVISOR: ResUNets with visibility and intensity for structured outlier removal. *ISPRS Journal of Photogrammetry and Remote Sensing* 202 (2023), 184–204.
- [15] Georg Krispel, Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. 2020. Fuseseg: Lidar point cloud segmentation fusing multi-modal data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1874–1883.
- [16] Ruihui Li, Xianzhi Li, Pheng-Ann Heng, and Chi-Wing Fu. 2020. Pointaugment: an auto-augmentation framework for point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6378–6387.
- [17] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. 2019. Selective kernel networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 510–519.
- [18] Ying Li, Lingfei Ma, Zilong Zhong, Fei Liu, Michael A Chapman, Dongpu Cao, and Jonathan Li. 2020. Deep learning for lidar point clouds in autonomous driving: A review. *IEEE Transactions on Neural Networks and Learning Systems* 32, 8 (2020), 3412–3432.
- [19] Ying Li and Huankun Sheng. 2023. A single-stage point cloud cleaning network for outlier removal and denoising. *Pattern Recognition* 138 (2023), 109366.
- [20] Hongbin Liu, Jinyuan Jia, and Neil Zhenqiang Gong. 2021. Pointguard: Provably robust 3d point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6186–6195.
- [21] Junyu Liu, Jianfeng Ren, Hongliang Sun, and Xudong Jiang. 2023. Face Recognition on Point Cloud with Cgan-Top for Denoising. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.
- [22] Lizhao Liu, Zhuangwei Zhuang, Shangxin Huang, Xunlong Xiao, Tianhang Xiang, Cen Chen, Jingdong Wang, and Mingkui Tan. 2023. Cpcm: Contextual point cloud modeling for weakly-supervised point cloud semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 18413–18422.
- [23] Jiawei Long, Hongmei Zhang, and Jianhu Zhao. 2023. A comprehensive deep learning-based outlier removal method for multibeam bathymetric point cloud. *IEEE Transactions on Geoscience and Remote Sensing* 61 (2023), 1–22.
- [24] Fan Lu, Guang Chen, Yinlong Liu, Lijun Zhang, Sanqing Qu, Shu Liu, and Rongqi Gu. 2021. Hregnet: A hierarchical network for large-scale outdoor lidar point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 16014–16023.
- [25] Yuhang Lu, Qi Jiang, Runnan Chen, Yuenan Hou, Xinge Zhu, and Yuexin Ma. 2023. See more and know more: Zero-shot point cloud segmentation via multi-modal visual data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 21674–21684.
- [26] Shitong Luo and Wei Hu. 2020. Differentiable manifold reconstruction for point cloud denoising. In *Proceedings of the 28th ACM International Conference on Multimedia*. 1330–1338.
- [27] Shitong Luo and Wei Hu. 2021. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2837–2845.
- [28] Shitong Luo and Wei Hu. 2021. Score-based point cloud denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4583–4592.
- [29] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. 2020. Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. *arXiv preprint arXiv:2011.13495* (2020).
- [30] Baorui Ma, Yu-Shen Liu, and Zhizhong Han. 2023. Learning Signed Distance Functions from Noisy 3D Point Clouds via Noise to Noise Mapping. In *International Conference on Machine Learning (ICML)*.
- [31] Aihua Mao, Zihui Du, Yu-Hui Wen, Jun Xuan, and Yong-Jin Liu. 2022. Pd-flow: A point cloud denoising framework with normalizing flows. In *European Conference on Computer Vision*. Springer, 398–415.
- [32] Christian Mostegel, Rudolf Prettenhaler, Friedrich Fraundorfer, and Horst Bischof. 2017. Scalable surface reconstruction from point clouds with extreme scale and density diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 904–913.
- [33] Liangliang Nan and Peter Wonka. 2017. Polyfit: Polygonal surface reconstruction from point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*. 2353–2361.
- [34] Keisuke Nonaka, Ryosuke Watanabe, Haruhisa Kato, Tatsuya Kobayashi, Eduardo Pavez, and Antonio Ortega. 2022. Graph-Based Point Cloud Denoising Using Shape-Aware Consistency For Free-Viewpoint Video. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4663–4667.
- [35] Gilles-Antoine Nys, Florent Poux, and Roland Billen. 2020. CityJSON building generation from airborne LiDAR 3D point clouds. *ISPRS International Journal of Geo-Information* 9, 9 (2020), 521.
- [36] Sameera Palipana, Dariush Salami, Luis A Leiva, and Stephan Sigg. 2021. Pan-tomime: Mid-air gesture recognition with sparse millimeter-wave radar point clouds. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 1 (2021), 1–27.
- [37] Stephan Palm and Uwe Stilla. 2020. 3-D point cloud generation from airborne single-pass and single-channel circular SAR data. *IEEE Transactions on Geoscience and Remote Sensing* 59, 10 (2020), 8398–8417.
- [38] Liang Pan, Xinyi Chen, Zhongang Cai, Junzhe Zhang, Haiyu Zhao, Shuai Yi, and Ziwei Liu. 2021. Variational relational point completion network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8524–8533.
- [39] Liang Pan, Xinyi Chen, Zhongang Cai, Junzhe Zhang, Haiyu Zhao, Shuai Yi, and Ziwei Liu. 2023. Variational relational point completion network for robust 3D classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).
- [40] Francesca Pistilli, Giulia Fracastoro, Diego Valsesia, and Enrico Magli. 2020. Learning graph-convolutional representations for point cloud denoising. In *European Conference on Computer Vision*. Springer, 103–118.
- [41] Francesca Pistilli, Giulia Fracastoro, Diego Valsesia, and Enrico Magli. 2020. Learning robust graph-convolutional representations for point cloud denoising. *IEEE Journal of Selected Topics in Signal Processing* 15, 2 (2020), 402–414.
- [42] Akarsh Prabhakara, Tao Jin, Arnav Das, Gantavya Bhatt, Lilly Kumari, Elahe Soltanaghai, Jeff Bilmes, Swarun Kumar, and Anthony Rowe. 2023. High resolution point clouds from mmwave radar. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 4135–4142.
- [43] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. [n. d.]. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 652–660.
- [44] Kun Qian, Zhaoyuan He, and Xinyu Zhang. 2020. 3D point cloud generation with millimeter-wave radar. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 4 (2020), 1–23.

- [45] Marie-Julie Rakotosaona, Vittorio La Barbera, Paul Guerrero, Niloy J Mitra, and Maks Ovsjanikov. 2020. Pointcleannet: Learning to denoise and remove outliers from dense point clouds. In *Computer graphics forum*, Vol. 39. Wiley Online Library, 185–203.
- [46] Andrés Serna, Beatriz Marcotegui, François Goulette, and Jean-Emmanuel Deschaud. 2014. 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*.
- [47] Rajat Sharma, Tobias Schwandt, Christian Kunert, Steffen Urban, and Wolfgang Broll. 2021. Point cloud upsampling and normal estimation using deep learning for robust surface reconstruction. *arXiv preprint arXiv:2102.13391* (2021).
- [48] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. 2020. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 8 (2020), 2647–2664.
- [49] Weijing Shi and Raj Rajkumar. 2020. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1711–1719.
- [50] Martin Simonovsky and Nikos Komodakis. 2017. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3693–3702.
- [51] Akash Deep Singh, Sandeep Singh Sandha, Luis Garcia, and Mani Srivastava. 2019. Radhar: Human activity recognition from point clouds generated through a millimeter-wave radar. In *Proceedings of the 3rd ACM Workshop on Millimeter-wave Networks and Sensing Systems*. 51–56.
- [52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30 (2017).
- [53] Peng-Shuai Wang, Yang Liu, and Xin Tong. 2016. Mesh denoising via cascaded normal regression. *ACM Trans. Graph.* 35, 6 (2016), 232–1.
- [54] Ruisheng Wang, Shangfeng Huang, and Hongxin Yang. 2023. Building3D: A Urban-Scale Dataset and Benchmarks for Learning Roof Structures from Point Clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 20076–20086.
- [55] Weijia Wang, Wei Pan, Xiao Liu, Kui Su, Bernard Rolfe, and Xuequan Lu. 2023. Random screening-based feature aggregation for point cloud denoising. *Computers & Graphics* 116 (2023), 64–72.
- [56] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. 2019. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)* 38, 5 (2019), 1–12.
- [57] Zeyong Wei, Honghua Chen, Liangliang Nan, Jun Wang, Jing Qin, and Mingqiang Wei. 2024. PathNet: Path-selective point cloud denoising. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).
- [58] Chengzhi Wu, Junwei Zheng, Julius Pfrommer, and Jürgen Beyerer. 2023. Attention-based point cloud edge sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5333–5343.
- [59] Jianyun Xu, Ruixiang Zhang, Jian Dou, Yushi Zhu, Jie Sun, and Shiliang Pu. 2021. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 16024–16033.
- [60] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. 2021. Multimodal virtual point 3d detection. *Advances in Neural Information Processing Systems* 34 (2021), 16494–16507.
- [61] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. 2018. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2790–2799.
- [62] Bangjie Zhang, Gang Xu, Hanwen Yu, Hui Wang, Hao Pei, and Wei Hong. 2023. Array 3-D SAR tomography using robust gridless compressed sensing. *IEEE Transactions on Geoscience and Remote Sensing* 61 (2023), 1–13.
- [63] Hanqing Zhang, Yun Lin, Fei Teng, and Wen Hong. 2022. A probabilistic approach for stereo 3D point cloud reconstruction from airborne single-channel multi-aspect SAR image sequences. *Remote Sensing* 14, 22 (2022), 5715.
- [64] Wenwei Zhang, Zhe Wang, and Chen Change Loy. 2020. Exploring data augmentation for multi-modality 3d object detection. *arXiv preprint arXiv:2012.12741* (2020).
- [65] Xue Zhang, Gene Cheung, Jiahao Pang, Yash Sanghvi, Abhiram Gnanasambandam, and Stanley H Chan. 2022. Graph-Based Depth Denoising & Dequantization for Point Cloud Enhancement. *IEEE Transactions on Image Processing* 31 (2022), 6863–6878.
- [66] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. 2020. Polarnet: An improved grid representation for on-line lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9601–9610.
- [67] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. 2021. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 16259–16268.
- [68] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. 2021. Few-shot 3d point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8873–8882.
- [69] Yaping Zhao, Haitian Zheng, Zhongrui Wang, Jiebo Luo, and Edmund Y Lam. 2023. Point cloud denoising via momentum ascent in gradient fields. In *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE, 161–165.
- [70] Wu Zheng, Mingxuan Hong, Li Jiang, and Chi-Wing Fu. 2022. Boosting 3d object detection by simulating multimodality on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13638–13647.