COUPLED SUBSETS FOR SELF-SUPERVISED POINT CLOUD DENOISING FROM A SINGLE SCAN

Anonymous authors
Paper under double-blind review

ABSTRACT

Noisy-only learning allows point cloud denoisers to be trained without clean targets. When only a single scan of each shape is available, a natural way to obtain Noise2Noise-style supervision is to split the scan into two subsets and enforce consistency between their denoised outputs. This strategy, however, implicitly assumes that the two subsets share the same clean samples. We observe that this assumption generally fails for point clouds: subsets drawn from one scan contain different surface samples that differ in spatial coverage and local density. We term this discrepancy the coupling gap and show that a set-level consistency loss cannot be reduced below it without moving the outputs away from the clean geometry. We address the coupling gap at two levels. At the data level, a coupled paired-subset sampler builds the subsets from local point pairs, which bounds the coupling gap. At the representation level, an invertible network factorizes the latent space into a subset-invariant structure stream and a subset-specific nuisance stream, and reconstructs denoised coordinates from the structure stream alone; unlike prior factorizations, it is learned without clean targets. Training combines an Earth Mover's Distance between the denoised subsets, an entropic optimal-transport consistency between structure latents, and an input-constrained displacement regularizer. Experiments on synthetic benchmarks and real scans show that our method achieves the best overall performance among noisy-only denoisers and remains competitive with supervised denoisers. *Code will be made publicly available.*

1 INTRODUCTION

Point cloud denoising aims to recover clean surface geometry from noisy observations. It is fundamental to 3D scanning pipelines, as sensor noise distorts local geometry and degrades downstream tasks such as surface reconstruction. Learning-based denoisers have made substantial progress, from PointNet-style displacement regression Rakotosaona et al. (2020); Zhang et al. (2020) to score estimation, iterative refinement, normalizing flows, and diffusion Luo & Hu (2021); Mao et al. (2022); Zhao et al. (2023); Vogel et al. (2024); Zhou et al. (2025b). Most of these methods, however, require clean point clouds for supervision. Such clean counterparts are rarely available for real scans, and obtaining them is costly and difficult to scale. This motivates learning directly from noisy observations.

Noisy-only learning has a well-established foundation in image denoising Lehtinen et al. (2018); Huang et al. (2021). For point clouds, Total Denoising Hermosilla et al. (2019) learns from a single noisy scan using local neighborhood supervision. Noise2NoiseMapping Ma et al. (2023) learns an implicit field with an EMD-based noise-to-noise objective, benefiting from multiple noisy observations of the same shape, a setting also adopted by U-CAN Zhou et al. (2025a). Noise2Score3D Wei et al. (2025) instead learns the score of the noisy distribution and denoises via Tweedie's formula. Despite these advances, learning a generalizable denoiser from only one noisy scan per shape, without clean targets or an explicit noise model, remains challenging.

A natural approach is to split a single scan into two non-overlapping subsets and enforce consistency between their denoised outputs Huang et al. (2021). This creates two distinct noisy views of the same underlying surface without requiring repeated scans or synthetic noise. However, the two subsets contain different surface samples. Even if both are perfectly denoised, their point sets need not

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

coincide. We refer to this inherent sampling discrepancy as the *coupling gap*. Consequently, a set-level consistency loss penalizes not only residual noise but also differences between clean samples, potentially encouraging geometrically correct points to move merely to improve agreement. We formalize this discrepancy and its effect on consistency supervision in Sec. 3.2. The key challenge is therefore to exploit the shared geometry between subsets without forcing their different surface samples to coincide.

To address this challenge, we propose a coupled sampling and representation learning framework. At the data level, a coupled paired-subset sampler assigns the endpoints of local point pairs to different subsets, which bounds the coupling gap by the distance within pairs. At the representation level, since sampling alone cannot eliminate the gap, we adopt an invertible backbone that factorizes the latent representation into a shared structure stream and a compact nuisance stream. Consistency is encouraged between the structure codes, while the nuisance stream accommodates subset-specific variations. Denoised coordinates are decoded with the nuisance code set to zero. Unlike previous structure-noise factorizations learned with clean supervision Mao et al. (2022; 2024), our framework learns from noisy observations alone. Training combines EMD consistency between denoised subsets, entropic optimal-transport consistency between structure codes, and a point-wise displacement regularizer that keeps points from moving merely to match the other subset.

Unlike Total Denoising, our supervision is constructed from subsets covering the same surface rather than local neighborhood targets. Unlike Noise2NoiseMapping, our feed-forward network generalizes to unseen shapes without per-shape optimization. Unlike Noise2Score3D, our method does not require an explicit noise model. Experiments on synthetic benchmarks and real scans show that our method achieves the best overall performance among noisy-only denoisers and remains competitive with supervised ones.

Our contributions are summarized as follows:

- We characterize the coupling gap between subsets of a single noisy scan and show how sampling discrepancies introduce bias into set-level consistency supervision.
- We propose a coupled paired-subset sampler whose pairs bound this gap, and a structure-nuisance invertible factorization that learns shared geometry from noisy observations alone.
- Experiments on synthetic benchmarks and real scans demonstrate the effectiveness of the proposed framework, with ablations validating the contributions of coupled sampling and structure-nuisance factorization.

2 Related Work

2.1 Supervised Point Cloud Denoising

Point cloud denoising aims to recover clean geometry from noisy point observations, and supervised methods learn this process using clean point clouds as training supervision. Early attempts generated cleaner point clouds by adapting 2D CNN-style processing to the projected representations, such as PointProNet Roveri et al. (2018), while later works shifted toward directly predicting per-point displacements. This line evolved from PointNet/PointNet++-style encoders Qi et al. (2017a;b) and low-level geometric networks Guerrero et al. (2018); Yu et al. (2018a) to dedicated denoisers such as PointCleanNet Rakotosaona et al. (2020) and Pointfilter Zhang et al. (2020), with subsequent improvements from multi-scale modeling Huang et al. (2022), contrastive learning de Silva Edirimuni et al. (2023a), routing strategies Wei et al. (2024), and lightweight architectures Sheng & Li (2024). Graph-based pipelines, inspired by DGCNN Wang et al. (2019), enhance local structure modeling through graph-convolutional feature learning Pistilli et al. (2020),and have been further extended by normal-aware filtering Liu et al. (2023), iterative refinement de Silva Edirimuni et al. (2023b), path-based denoising de Silva Edirimuni et al. (2024), progressive multi-stage refinement Chen et al. (2024), adaptive stopping mechanisms Guo et al. (2025), and efficient dense-input filtering with global sequence modeling Zhou et al. (2025b). Recurrent, attention-based, and Transformer-based denoisers further improve feature aggregation and non-local dependency modeling Chen et al. (2022); Zhu et al. (2023); Wang & Li (2023); Ran et al. (2025). More recently, score- and diffusion-inspired methods Luo & Hu (2021); Chen et al. (2023); Zhao et al. (2023); Yang et al. (2024); Vogel et al. (2024), building on diffusion models Song et al. (2020); Ho et al. (2020), estimate distribu-

tions or gradients to guide noisy points toward underlying surfaces, while invertible mapping-based methods learn transformations between noisy and clean point distributions Mao et al. (2022; 2024). Despite these advances, supervised methods rely on paired clean targets, which are expensive to acquire for real scans.

2.2 Noisy-Only and Unsupervised Point Cloud Denoising

Noisy-only denoising removes the need for noisy-clean pairs and instead exploits geometric objectives, self-supervision, or probabilistic structure. In image denoising, Noise2Noise Lehtinen et al. (2018) trains with pairs of noisy observations, and Neighbor2Neighbor Huang et al. (2021), Noise2Void Krull et al. (2019), and Noise2Self Batson & Royer (2019) learn from single images through sub-sampling or masked prediction, relying on a regular pixel grid that point clouds lack. Early work such as Total Denoising Hermosilla et al. (2019) learns from a single noisy point cloud by regressing each point toward samples from its local neighborhood under a spatial prior. Self-supervised learning can also be realized by synthesizing noisy variants, e.g., Noise4Denoise Wang et al. (2024) trains displacement prediction from paired noisy inputs under a designed corruption scheme. In the probabilistic direction, score-based methods extend unsupervised score learning to 3D and recover clean estimates via posterior-mean principles (e.g., Noise2Score3D Wei et al. (2025), building on Noise2Score Kim & Ye (2021) and Tweedie's formula Efron (2011)). Most relevant to our work are consistency-based methods that typically learn from multiple noisy observations by enforcing agreement between them using permutation-invariant objectives Ma et al. (2023); Zhou et al. (2025a). Like Total Denoising, our work uses a single observation, but supervision is extracted from two spatially coupled subsets of one corrupted point cloud. To make this signal reliable, we combine coupled subset construction, which bounds the coupling gap between the subsets, with a dual-stream invertible factorization that separates shared surface structure from subset-specific nuisance variation.

2.3 Consistency Learning and Invertible Representations

Consistency learning provides supervision by enforcing agreement between corrupted observations. For point clouds, it must address permutation ambiguity and missing correspondences, often via EMD-based objectives Ma et al. (2023); Zhou et al. (2025a). Meanwhile, normalizing flows and invertible neural networks (INNs) provide structured latent representations through bijective mappings and have been widely studied for density estimation and generation Kingma & Dhariwal (2018); Grathwohl et al. (2018); Yang et al. (2019). In 3D restoration, invertible designs have been explored mostly under supervised or likelihood-driven settings, including latent disentanglement for denoising (PD-Flow) Mao et al. (2022), continuous flow for upsampling (ND-PUFlow) Hu et al. (2023), and improved latent restoration with stronger geometric encoders (PD-LTS) Mao et al. (2024). Existing noisy-only denoisers mainly impose consistency in the observation space, whereas invertible restoration models are typically trained with clean supervision or likelihood-based objectives. Our method leverages a flow-inspired invertible backbone for self-supervised denoising. In addition to matching the denoised subsets in coordinate space, it separates subset-invariant geometry from subset-specific nuisance in the latent space and reconstructs denoised point coordinates from the structure representation.

3 Methodology

3.1 Overview

Let $\mathbf{X} = \{x_i\}_{i=1}^{N} \subset \mathbb{R}^3$ denote a noisy patch sampled around an unknown clean surface $\mathcal{S}$. The corresponding clean point cloud is unavailable during training. Unlike images, point clouds are unordered, and noise directly perturbs their spatial coordinates. As a result, there is no canonical index correspondence across different noisy acquisitions.

We study a single-observation noisy-only setting, where supervision must be extracted from $\mathbf{X}$ itself. We construct two disjoint but spatially coupled subsets, $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$, from the same noisy patch. Since both subsets sample the same underlying surface, they should share stable geometric structure. Meanwhile, their sampling differences, local density variations, and noise perturbations are treated

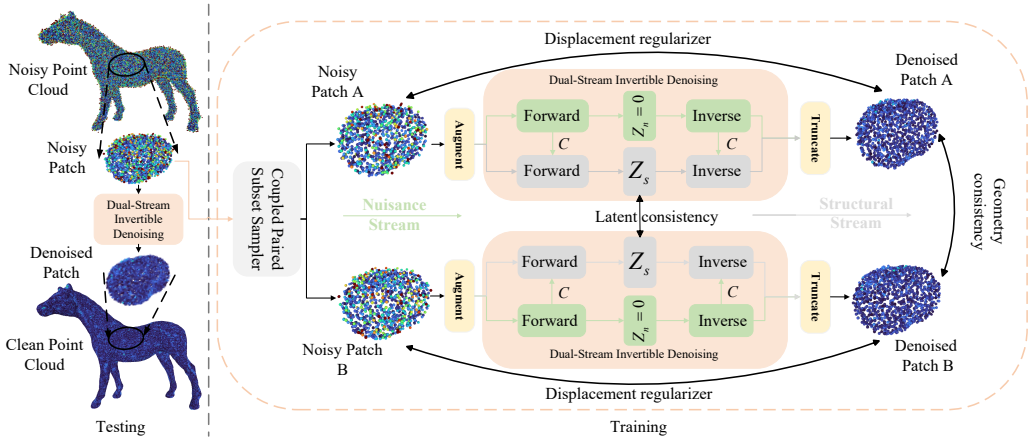

Figure 1: Overview of our method.

as subset-specific nuisance. The sampling differences also limit how far a consistency loss between the two subsets can be reduced without distorting the geometry, which we analyze in Sec. 3.2.

Based on this view, we learn a denoising operator D_θ that maps a noisy patch to a cleaned estimate:

$$\hat{\mathbf{Y}} = D_\theta(\mathbf{X}). \tag{1}$$

Internally, D_θ is implemented by a dual-stream invertible backbone F_θ. The backbone factorizes the latent representation into a structure code $\mathbf{Z}_s$ and a nuisance code $\mathbf{Z}_n$. Denoising is performed by mapping the structure representation back to Euclidean point coordinates while suppressing nuisance variation.

The overall pipeline has three steps. First, a coupled paired-subset sampler constructs $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ with reduced discrepancy in spatial coverage and local density. Second, the shared dual-stream invertible backbone maps each subset into structure and nuisance representations. Third, training is guided by a composite objective with three terms: geometry consistency in Euclidean coordinate space, latent-structure consistency, and displacement regularization. The first two terms provide complementary consistency constraints, while the displacement regularizer stabilizes training by preventing excessive point movement. Figure 1 illustrates the overall framework.

3.2 Coupling Gap between Paired Subsets

Let $\mathbf{X}^{*(1)}$ and $\mathbf{X}^{*(2)}$ be the unknown clean counterparts of the two subsets. Because the subsets are disjoint, they generally differ in spatial coverage and local density, so $\mathbf{X}^{*(1)} \neq \mathbf{X}^{*(2)}$. For point sets of equal size M, let $\mathcal{D}_{\text{EMD}}(\mathbf{A}, \mathbf{B}) = \left(\min_\pi \frac{1}{M}\sum_i \|a_i - b_{\pi(i)}\|_2^2\right)^{1/2}$ denote the Earth Mover's Distance with squared Euclidean cost, i.e., the 2-Wasserstein distance between the empirical measures, which is a metric. We quantify the discrepancy by the *coupling gap* $\Delta = \mathcal{D}_{\text{EMD}}(\mathbf{X}^{*(1)}, \mathbf{X}^{*(2)})$.

Proposition 1. *For any outputs $\hat{\mathbf{Y}}^{(1)}, \hat{\mathbf{Y}}^{(2)}$, the triangle inequality gives*

$$\sum_{m=1}^{2} \mathcal{D}_{\text{EMD}}\big(\hat{\mathbf{Y}}^{(m)}, \mathbf{X}^{*(m)}\big) \geq \Delta - \mathcal{D}_{\text{EMD}}\big(\hat{\mathbf{Y}}^{(1)}, \hat{\mathbf{Y}}^{(2)}\big). \tag{2}$$

An ideal denoiser, $\hat{\mathbf{Y}}^{(m)} = \mathbf{X}^{*(m)}$, attains $\mathcal{D}_{\text{EMD}}(\hat{\mathbf{Y}}^{(1)}, \hat{\mathbf{Y}}^{(2)}) = \Delta$. Since the consistency loss increases monotonically with this distance (Sec. 3.5), reducing it further moves the outputs away from the clean geometry, i.e., the loss rewards fitting the sampling difference. A random split leaves Δ uncontrolled, whereas our sampler bounds it (Sec. 3.3). The clean counterparts are used only for this analysis.

3.3 Coupled Paired-Subset Construction

The first step is to construct two subsets that can provide a reliable self-supervised signal from a single noisy patch. A naive random split may produce subsets with different spatial support and

local density, i.e., a large coupling gap, so the consistency loss can be dominated by sampling discrepancy rather than denoising. FPS-based splitting improves the global coverage of one subset, but the remaining points, which form the other subset, may still have mismatched local support. We therefore design a coupled paired-subset sampler that forms local point pairs and assigns one endpoint of each pair to each subset. This produces two disjoint but spatially coupled subsets with more balanced local coverage. Figure 2 compares random, FPS-based, and our coupled sampling strategy.

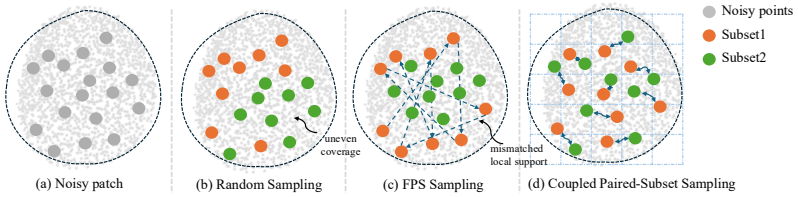

Figure 2: Random sampling often produces uneven spatial coverage, while FPS sampling leaves the remaining subset with mismatched local support. Our coupled paired-subset sampler explicitly constructs paired points spatially, resulting in balanced local support across the two subsets.

Given a noisy patch $\mathbf{X} = \{x_i\}_{i=1}^{N}$, our aim is to generate two subsets $\mathbf{X}^{(1)}, \mathbf{X}^{(2)} \in \mathbb{R}^{M\times 3}$ and a disjoint pair list

$$\begin{aligned} (\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{P}) &= \mathcal{T}(\mathbf{X}), \\ \mathbf{P} = \{(a_j, b_j)\}_{j=1}^{M}, \qquad M &= \lfloor \rho N \rfloor, \;\; \rho \le \tfrac{1}{2}, \end{aligned} \tag{3}$$

where each original index appears in at most one pair. Each pair contributes exactly one endpoint to each subset, hence $\mathbf{X}^{(1)} \cap \mathbf{X}^{(2)} = \emptyset$ and $|\mathbf{X}^{(1)}| = |\mathbf{X}^{(2)}| = M$.

Importantly, $\mathbf{P}$ is only used to couple subset formation and is not used as a point-wise clean correspondence during training.

We implement $\mathcal{T}$ with a two-stage pairing strategy: (i) *voxel-stratified pairing* balances spatial occupancy by forming disjoint pairs within each voxel, where at most r pairs are selected from each voxel; (ii) *greedy kNN completion* fills the remaining pairs by selecting, in order of increasing length, the edges of the k_fNN graph of the patch whose endpoints are both still unused, where k_f controls the neighborhood size in the completion stage. In our implementation, we use a fixed voxel size $h = 0.03$ in the unit-sphere normalized space, set $r = 1$, and use $k_f = 32$. If the target number of pairs is still not reached, we randomly pair the remaining unused indices to ensure a fixed output size. Finally, a fair coin flip assigns the two endpoints of each pair to different subsets, producing two disjoint and spatially coupled subsets suitable for stable self-supervision.

Remark. Because each pair contributes one endpoint to each subset, $\mathbf{P}$ defines a one-to-one matching between $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$. Since $\mathcal{D}_{\text{EMD}}$ is the minimum over all such matchings, the coupling gap is bounded by the root-mean-square clean distance within pairs:

$$\Delta \;\le\; \Big(\frac{1}{M}\sum_{j=1}^{M} \big\|x^*_{a_j} - x^*_{b_j}\big\|_2^2\Big)^{1/2}. \tag{4}$$

Voxel-stratified pairing keeps the paired points within the same voxel, and greedy kNN completion selects the shortest available edges, so both stages keep this bound small; a random split imposes no such constraint.

3.4 Dual-Stream Invertible Denoising

After constructing two spatially coupled subsets, we aim to separate geometry-preserving information from observation-specific perturbations. To this end, we introduce a dual-stream invertible factorization tailored to single-observation self-supervised denoising. The model decomposes the latent representation into a structure stream and a nuisance stream. The structure stream is encouraged to encode information shared by the two coupled subsets, whereas the nuisance stream absorbs

subset-specific noise and sampling variation. Different from prior single-stream flows Mao et al. (2022), which identify the noise channels with the help of clean targets, our key design is to explicitly route stable geometric information and input-dependent nuisance variation through separate streams, trained without clean targets, and to denoise by reconstructing point coordinates from the structure representation.

Specifically, each subset is augmented with local geometric features following PD-LTS Mao et al. (2024), giving $\mathbf{U}^{(m)}$, and mapped by an invertible network to

$$[\mathbf{Z}_s^{(m)}, \mathbf{Z}_n^{(m)}] = F_\theta\big(\mathbf{U}^{(m)}\big), \tag{5}$$

where the compact nuisance code $\mathbf{Z}_n^{(m)}$ ($C_n = 6$ channels) conditions the structure stream through a pooled global code. Denoising keeps the structure code and replaces the nuisance code by a canonical value $\bar{\mathbf{Z}}_n = \mathbf{0}$:

$$\hat{\mathbf{Y}}^{(m)} = \Pi_{xyz}\Big(F_\theta^{-1}\big([\mathbf{Z}_s^{(m)}, \bar{\mathbf{Z}}_n]\big)\Big) \in \mathbb{R}^{M\times 3}, \tag{6}$$

where $\Pi_{xyz}(\cdot)$ selects the coordinate channels. At test time, a single noisy input patch is sufficient; paired-subset construction is used only during training. The feature augmentation, stream structure, and conditioning are detailed in the supplementary material.

3.5 Training Objectives

Given $(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{P}) = \mathcal{T}(\mathbf{X})$, we obtain the denoised subsets $(\hat{\mathbf{Y}}^{(1)}, \hat{\mathbf{Y}}^{(2)})$ through the dual-stream denoiser. Rather than relying only on coordinate-space matching, we impose consistency at two complementary levels: the denoised point coordinates and the latent structure representation. Specifically, a geometry loss aligns the two denoised subsets in Euclidean coordinate space, while a latent-structure loss aligns their structure codes in the latent space. We further design a displacement regularizer to prevent excessive geometric drift by anchoring each output to its input. We optimize:

$$\mathcal{L} = \lambda_{\text{geom}}\mathcal{L}_{\text{geom}} + \lambda_{\text{disp}}\mathcal{L}_{\text{disp}} + \lambda_{\text{lat}}\mathcal{L}_{\text{lat}}. \tag{7}$$

(i) Geometry consistency. We enforce cross-subset consistency of denoised coordinates using:

$$\mathcal{L}_{\text{geom}} = M\,\mathcal{D}_{\text{EMD}}^2\big(\hat{\mathbf{Y}}^{(1)}, \hat{\mathbf{Y}}^{(2)}\big), \tag{8}$$

i.e., the sum of squared distances under the optimal assignment; this is the consistency loss analyzed in Proposition 1.

(ii) Displacement regularizer. To anchor each denoised subset to its own input, we penalize per-point displacement within each subset:

$$\mathcal{L}_{\text{disp}} = \frac{1}{2M}\sum_{m=1}^{2}\sum_{i=1}^{M}\phi\big(\|\hat{y}_i^{(m)} - x_i^{(m)}\|_2\big), \tag{9}$$

where $\phi(\cdot)$ is the Charbonnier penalty. This constrains each denoised subset to stay close to its own noisy input.

(iii) Latent structure consistency. We regularize the structure stream by encouraging the two subsets to have similar structure codes in a permutation-invariant manner. We use a Sinkhorn distance in the structure-code space:

$$\mathcal{L}_{\text{lat}} = \mathcal{D}_{\text{SK}}^{\varepsilon}\big(\mathbf{Z}_s^{(1)}, \mathbf{Z}_s^{(2)}\big). \tag{10}$$

Here $\mathcal{D}_{\text{SK}}^{\varepsilon}$ is computed with uniform marginals and entropic regularization ε, using the pairwise cost $D_{ij} = \|\mathbf{z}_{s,i}^{(1)} - \mathbf{z}_{s,j}^{(2)}\|_1$.

4 Experiments

4.1 Experimental Setting

Datasets and metrics. Following the standard protocol Luo & Hu (2021), we train on the PU-Net training set Yu et al. (2018b) and evaluate on the PU-Net and PC-Net Rakotosaona et al. (2020) test

Table 1: Comparison with noisy-only denoisers on PU-Net and PC-Net under Gaussian noise. CD and P2M are scaled by 10^4. Best and second-best results are in **bold** and underlined. †: results taken from the original paper, as the code is not publicly available.

	10K points						50K points					
	1%		2%		3%		1%		2%		3%	
Method	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**
PU-Net												
TotalDenoising	3.068	0.925	6.241	2.947	11.452	7.117	0.790	0.418	3.030	2.226	6.593	5.349
DMRDenoise-U	5.234	2.280	6.225	2.935	8.842	5.065	1.579	0.927	3.330	2.356	7.076	5.715
ScoreDenoise-U	3.558	1.069	5.231	2.020	7.249	3.523	1.046	0.467	2.216	1.304	4.413	3.140
Noise2Score3D	3.011	1.241	4.191	1.818	5.173	2.555	0.833	0.462	1.532	0.971	2.350	1.639
U-CAN†	2.497	1.105	3.234	**1.255**	**3.666**	**1.842**	0.835	0.609	0.975	0.675	2.479	1.863
Ours	**2.296**	**0.920**	**3.179**	1.402	3.798	1.848	**0.606**	**0.369**	**0.843**	**0.562**	**2.019**	**1.492**
PC-Net												
TotalDenoising	3.918	1.771	7.308	4.178	12.764	8.941	1.010	0.482	2.492	1.693	6.450	5.275
DMRDenoise-U	6.482	3.676	7.716	4.610	10.428	6.952	1.708	0.980	3.444	2.482	7.000	5.763
ScoreDenoise-U	4.324	1.944	6.148	3.173	8.320	4.970	1.258	0.584	2.428	1.513	4.574	3.415
Noise2Score3D	3.552	1.784	5.096	2.652	6.335	3.637	1.027	0.504	1.710	1.064	2.668	1.891
Ours	**3.029**	**1.481**	**4.146**	**2.117**	**4.958**	**2.745**	**0.792**	**0.386**	**1.244**	**0.747**	**2.005**	**1.430**

sets at 10K and 50K points, with Gaussian noise whose standard deviation is 1%, 2%, or 3% of the bounding-sphere radius. To test generalization beyond Gaussian noise, we additionally evaluate four non-Gaussian noise types on PU-Net at 50K points. For real scans, we use Kinect v1/v2 Wang et al. (2016), which provides reference meshes for quantitative evaluation, and RueMadame Serna et al. (2014), which has no ground truth and is evaluated qualitatively. We report Chamfer Distance (CD) and Point-to-Mesh distance (P2M), both scaled by 10^4, and their definitions are given in the supplementary material. For each 2048-point noisy patch, the coupled sampler (Sec. 3.3) constructs two disjoint 1024-point subsets $(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$, and no clean point cloud or clean correspondence enters the training objective.

Baselines. Our main comparison is with noisy-only denoisers that, like ours, train a network once and apply it to unseen shapes: Total Denoising Hermosilla et al. (2019), the unsupervised variants of DMRDenoise and ScoreDenoise (DMRDenoise-U and ScoreDenoise-U) Luo & Hu (2020; 2021), Noise2Score3D Wei et al. (2025), and U-CAN Zhou et al. (2025a). Since the official code of U-CAN is unavailable, its results (marked with †) are taken from the original paper. The supplementary material further compares with five supervised denoisers Rakotosaona et al. (2020); Zhang et al. (2020); Luo & Hu (2020; 2021); Chen et al. (2023), and with Noise2NoiseMapping Ma et al. (2023), which optimizes an implicit surface for each test shape instead of training a feed-forward network.

4.2 Comparison with Noisy-Only Methods

Table 1 reports the results under Gaussian noise. On PU-Net, our method obtains the lowest CD and P2M in all 50K settings and at 10K with 1% noise, as well as the lowest CD at 10K with 2% noise. In the remaining three entries, it ranks second after U-CAN, with a difference of at most 0.15 in CD and P2M. On PC-Net, for which U-CAN does not report results, our method achieves the lowest CD and P2M in all settings. As shown in Fig. 3, our method exhibits fewer high-error regions on both the Camel(2%) and Elephant(3%) shapes compared with the competing methods. For reference, the supplementary material compares with supervised denoisers trained with clean targets (Table 4). Without clean supervision, our method obtains the lowest CD in five of the six PU-Net settings, whereas supervised methods retain lower P2M in four of them. In a single-observation comparison with Noise2NoiseMapping (Table 6; PU-Net, 10K, 2% noise), our method attains lower CD (3.179 vs. 3.923) and P2M (1.402 vs. 1.737) and denoises a shape in about 1 s instead of 506 s, since Noise2NoiseMapping optimizes an implicit field separately for each test shape.

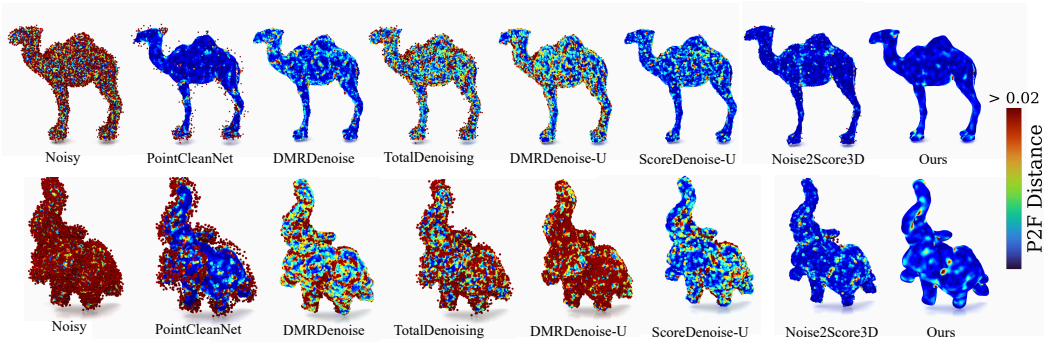

Figure 3: Qualitative comparison on Camel (2% Gaussian noise) and Elephant (3% Gaussian noise). Colors encode point-wise error from low (blue) to high (red).

Table 2: Generalization to unseen noise types and real scans. Left: PU-Net (50K) under four non-Gaussian noise types. Right: real Kinect v1/v2 scans with reference meshes Wang et al. (2016). All models are trained with Gaussian noise. CD and P2M are scaled by 10^4. Best and second-best results are in **bold** and underlined.

	Non-Gaussian noise (PU-Net, 50K)								Real scans			
	Cov		Laplacian		Discrete		UniformBall		Kinect v1		Kinect v2	
Method	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**
TotalDenoising	2.483	1.741	0.777	0.463	0.686	0.520	0.775	0.427	1.549	0.867	2.027	1.095
DMRDenoise-U	3.360	2.391	1.515	0.889	1.456	0.865	1.509	0.873	2.682	1.892	4.045	2.415
ScoreDenoise-U	2.238	1.329	0.872	0.382	0.642	0.303	1.007	0.442	1.717	0.933	2.671	1.163
Noise2Score3D	1.650	1.064	0.749	0.436	0.487	0.349	0.777	0.419	1.430	0.915	2.230	1.313
Ours	**1.276**	**0.883**	**0.535**	**0.332**	**0.350**	**0.281**	**0.565**	**0.340**	**1.326**	**0.807**	**2.015**	**1.041**

4.3 Generalization to Other Noise Types and Real Scans

Table 2 evaluates the models on four other noise distributions and data not seen during training. Our method achieves the lowest CD and P2M in all twelve entries. Under the four non-Gaussian noise types, it reduces CD by 23-29% relative to the second-best method, indicating strong robustness to noise distributions beyond those used in training. As shown in Fig. 4, our method better preserves the architectural structures and local details on the real RueMadame_2 scan while effectively removing noise.

4.4 Ablation and Analysis

Table 3 analyzes our method on PU-Net (10K). Each variant changes one component of the full model, so that differences can be attributed to that component.

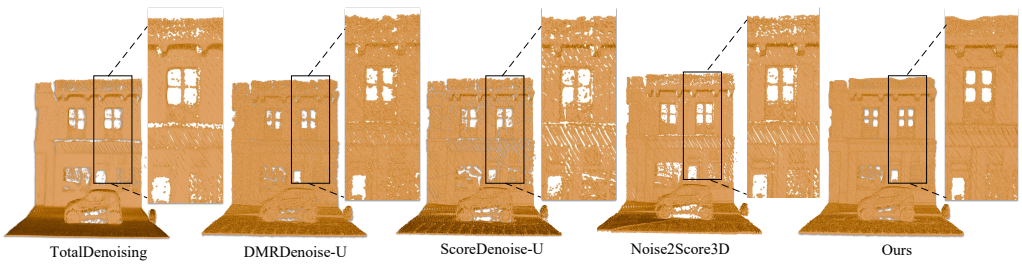

Figure 4: Qualitative comparison on the real RueMadame_2 scan, with zoom-ins.

Table 3: Ablation on PU-Net (10K). The full model uses the coupled sampler, the dual-stream INN, and $\mathcal{L}_{\text{geom}} + \mathcal{L}_{\text{lat}} + \mathcal{L}_{\text{disp}}$; each variant below changes one component. Disp., Geom., and Latent denote $\mathcal{L}_{\text{disp}}$, $\mathcal{L}_{\text{geom}}$, and $\mathcal{L}_{\text{lat}}$. CD and P2M are scaled by 10^4. The lowest and second-lowest results are in **bold** and underlined.

	1%		2%		3%	
Variant	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**
Full model (Ours)	<u>2.296</u>	<u>0.920</u>	**3.179**	**1.402**	**3.798**	**1.848**
Paired-subset Samplers						
(a) Random	3.228	1.842	4.152	2.478	4.583	2.689
(b) FPS	2.727	1.324	3.530	1.772	4.064	2.085
Network Backbones						
(a) Regular (DGCNN)	2.368	0.981	3.394	1.465	4.530	2.252
(b) Single-INN	2.523	1.150	<u>3.236</u>	<u>1.459</u>	<u>3.842</u>	<u>1.862</u>
Loss Components						
(a) Disp.	3.690	1.603	7.939	4.772	13.382	9.600
(b) Disp. + Geom.	2.392	1.006	3.342	1.538	4.035	2.047
(c) EMD + Geom. + Latent	**2.001**	**0.579**	3.692	1.622	7.209	4.447
(d) One-sided + Geom. + Latent	5.375	3.722	5.723	3.808	6.263	4.066

Coupling gap and sampler. The denoising error follows the same order at all noise levels: replacing our sampler with FPS or random splitting increases CD at 1% noise from 2.296 to 2.727 and 3.228, respectively. This indicates that reducing the coupling gap of the training pairs improves the self-supervised signal.

Structure-nuisance factorization. Replacing the dual-stream INN with a non-invertible DGCNN-based encoder-decoder Wang et al. (2019) or a single-stream INN Mao et al. (2022), trained with the same sampler and objective, increases the error at all noise levels. The single-stream INN is close to ours at 2% and 3% noise but falls behind even the non-invertible network at 1% noise, whereas the factorized design is best in all settings.

Loss components. Using the displacement regularizer $\mathcal{L}_{\text{disp}}$ alone yields the highest errors under 2% and 3% noise, since it provides no cross-subset signal. Adding the geometry consistency $\mathcal{L}_{\text{geom}}$ (w/o $\mathcal{L}_{\text{lat}}$) accounts for most of the improvement. Adding the latent consistency $\mathcal{L}_{\text{lat}}$ further reduces CD by 0.096, 0.163, and 0.237 at 1%, 2%, and 3% noise; the gain grows with the noise level, consistent with its role of separating noise from structure. We also compare two alternatives to $\mathcal{L}_{\text{disp}}$. Replacing it with the set-level EMD constraint between input and output used in U-CAN Zhou et al. (2025a) gives the lowest error at 1% noise but degrades sharply at 2% and 3% noise (CD 7.209 vs. 3.798 at 3%). A one-sided variant, which regularizes only one of the two subsets, is worse at all noise levels. We therefore adopt the symmetric point-wise $\mathcal{L}_{\text{disp}}$, which is the most stable across noise levels.

5 Conclusion

We studied self-supervised point cloud denoising from a single noisy scan and identified the coupling gap between sub-sampled subsets as a source of bias: because two subsets of one scan differ in spatial coverage and local density, a set-level consistency loss cannot be reduced below this gap without moving the outputs away from the clean geometry. We addressed this gap at two levels. At the data level, a coupled paired-subset sampler bounds it by the distance within local pairs; at the representation level, a dual-stream invertible network, trained without clean targets, separates subset-invariant structure from subset-specific nuisance. Experiments show that the coupled sampler outperforms random and FPS splitting, that the dual-stream backbone outperforms single-stream and non-invertible alternatives, and that the resulting method achieves the best overall performance among noisy-only denoisers on synthetic benchmarks and real scans while remaining competitive with supervised denoisers.

AI USE STATEMENT

Generative AI tools were used for language polishing. All AI-assisted outputs were reviewed and verified by the authors. The authors take full responsibility for the content of this work.

REFERENCES

Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *International conference on machine learning*, pp. 524–533. PMLR, 2019.

Haolan Chen, Bi'an Du, Shitong Luo, and Wei Hu. Deep point set resampling via gradient fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):2913–2930, 2023.

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

Wu Chen, Hehe Fan, Qiuping Jiang, Chao Huang, and Yi Yang. Progressive point cloud denoising with cross-stage cross-coder adaptive edge graph convolution network. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 6578–6587, 2024.

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

Dasith de Silva Edirimuni, Xuequan Lu, Zhiwen Shao, Gang Li, Antonio Robles-Kelly, and Ying He. Iterativepfn: True iterative point cloud filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13530–13539, 2023b.

Dasith de Silva Edirimuni, Xuequan Lu, Gang Li, Lei Wei, Antonio Robles-Kelly, and Hongdong Li. Straightpcf: Straight point cloud filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20721–20730, 2024.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy Jyoti Mitra. Pcpnet learning local shape properties from raw point clouds. *Computer Graphics Forum*, 37, 2018.

Chuchen Guo, Weijie Zhou, Zheng Liu, and Ying He. You should learn to stop denoising on point clouds in advance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 3212–3219, 2025.

Pedro Hermosilla, Tobias Ritschel, and Timo Ropinski. Total denoising: Unsupervised learning of 3d point cloud cleaning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 52–60, 2019.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Xin Hu, Xin Wei, and Jian Sun. A noising-denoising framework for point cloud upsampling via normalizing flows. *Pattern Recognition*, 140:109569, 2023.

Anyi Huang, Qian Xie, Zhoutao Wang, Dening Lu, Mingqiang Wei, and Jun Wang. Modnet: Multi-offset point cloud denoising network customized for multi-scale patches. In *Computer Graphics Forum*, volume 41, pp. 109–119. Wiley Online Library, 2022.

Tao Huang, Songjiang Li, Xu Jia, Huchuan Lu, and Jianzhuang Liu. Neighbor2neighbor: Self-supervised denoising from single noisy images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14781–14790, 2021.

Kwanyoung Kim and Jong Chul Ye. Noise2score: tweedie's approach to self-supervised image denoising without clean images. *Advances in Neural Information Processing Systems*, 34:864–874, 2021.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2129–2137, 2019.

Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: Learning image restoration without clean data. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2965–2974. PMLR, 10–15 Jul 2018.

Zheng Liu, Yaowu Zhao, Sijing Zhan, Yuanyuan Liu, Renjie Chen, and Ying He. Pcdnf: Revisiting learning-based point cloud denoising via joint normal filtering. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

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

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

Baorui Ma, Yu-Shen Liu, and Zhizhong Han. Learning signed distance functions from noisy 3d point clouds via noise to noise mapping. In *International Conference on Machine Learning (ICML)*, 2023.

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

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

Francesca Pistilli, Giulia Fracastoro, Diego Valsesia, and Enrico Magli. Learning robust graph-convolutional representations for point cloud denoising. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):402–414, 2020.

C. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, 2017a.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017b.

Marie-Julie Rakotosaona, Vittorio La Barbera, Paul Guerrero, Niloy J Mitra, and Maks Ovsjanikov. Pointcleannet: Learning to denoise and remove outliers from dense point clouds. In *Computer graphics forum*, volume 39, pp. 185–203. Wiley Online Library, 2020.

Chunqing Ran, Xiaobo Zhang, Shuo Han, Hao Yu, and Shengli Wang. Tpdnet: A point cloud data denoising method for offshore drilling platforms and its application. *Measurement*, 241:115671, 2025.

Riccardo Roveri, A. Cengiz Öztireli, Ioana Pandele, and Markus H. Gross. Pointpronets: Consolidation of point clouds with convolutional neural networks. *Computer Graphics Forum*, 37, 2018.

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

Huankun Sheng and Ying Li. Denoising point clouds with fewer learnable parameters. *Computer-Aided Design*, 172:103708, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

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

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

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

Weijia Wang, Xiao Liu, Hailing Zhou, Lei Wei, Zhigang Deng, Manzur Murshed, and Xuequan Lu. Noise4denoise: Leveraging noise for unsupervised point cloud denoising. *Computational Visual Media*, 10(4):659–669, 2024.

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

Xiangbin Wei, Yuanfeng Wang, Ao Xu, Lingyu Zhu, Dongyong Sun, Keren Li, Yang Li, and Qi Qin. Noise2score3d: Tweedie's approach for unsupervised point cloud denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 25993–26003, 2025.

Zeyong Wei, Honghua Chen, Liangliang Nan, Jun Wang, Jing Qin, and Mingqiang Wei. Pathnet: Path-selective point cloud denoising. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(6):4426–4442, 2024.

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

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

Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Ec-net: an edge-aware point set consolidation network. In *European Conference on Computer Vision*, 2018a.

Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2790–2799, 2018b.

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

Yaping Zhao, Haitian Zheng, Zhongrui Wang, Jiebo Luo, and Edmund Y Lam. Point cloud denoising via momentum ascent in gradient fields. In *2023 IEEE International Conference on Image Processing (ICIP)*, pp. 161–165. IEEE, 2023.

Junsheng Zhou, Xingyu Shi, Haichuan Song, Yi Fang, Yu-Shen Liu, and Zhizhong Han. U-can: Unsupervised point cloud denoising with consistency-aware noise2noise matching. *arXiv preprint arXiv:2510.25210*, 2025a.

Qingyuan Zhou, Weidong Yang, Ben Fei, Jingyi Xu, Rui Zhang, Keyi Liu, Yeqi Luo, and Ying He. 3dmambaipf: A state space model for iterative point cloud filtering via differentiable rendering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 10843–10851, 2025b.

Xusheng Zhu, Shuai Ma, Daixin Chen, Li Zhou, and Haibo Tang. Msad-net: a mix self-attention networks for 3d point cloud denoising. *IEEE Photonics Journal*, 15(3):1–7, 2023.

A Appendix

A.1 Details of the Coupled Paired-Subset Sampler

Algorithm 1 summarizes the procedure of the proposed coupled paired-subset sampler.

Algorithm 1 Coupled Paired-Subset Sampler $\mathcal{T}$

Require: $\mathbf{X} = \{x_i\}_{i=1}^{N}$; $\rho \le \frac{1}{2}$; r (max pairs/voxel); k_f (kNN); voxel size h
Ensure: $\mathbf{X}^{(1)}, \mathbf{X}^{(2)} \in \mathbb{R}^{M\times 3}$; $\mathbf{P} = \{(a_j, b_j)\}_{j=1}^{M}$

$M \leftarrow \min(\lfloor \rho N \rfloor,\ \lfloor N/2 \rfloor)$
if h is not given **then**
  $h \leftarrow q \cdot \mathrm{median}\{\|x_i - x_{\mathrm{NN}_k(i)}\|_2\}$
end if
$\mathbf{P} \leftarrow \textsc{VoxelMatch}(\mathbf{X}, h, r)$
$\mathbf{P} \leftarrow \mathbf{P} \cup \textsc{GreedyKNNMatch}(\mathbf{X}, k_f,\ M - |\mathbf{P}|)$
if $|\mathbf{P}| < M$ **then**
  $\mathbf{P} \leftarrow \mathbf{P} \cup \textsc{RandomMatch}(\mathbf{X},\ M - |\mathbf{P}|)$
end if
$(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) \leftarrow \textsc{SplitByCoinFlip}(\mathbf{X}, \mathbf{P})$
return $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{P}$

Implementation. The median in the adaptive voxel size is estimated on a random subset for efficiency. VoxelMatch: bucket indices by voxel grid of size h and sample up to r disjoint pairs per voxel. GreedyKNNMatch: build a k_f-NN graph on all points and greedily select the shortest edges whose endpoints are both unused as new disjoint pairs. RandomMatch: randomly pair remaining unused indices to reach M pairs (rare). SplitByCoinFlip: for each (a_j, b_j), sample $s_j \sim \mathrm{Bern}(1/2)$ and assign one endpoint to each subset.

A.2 Architecture Details

Before applying the invertible backbone, we augment each point with local geometric features following PD-LTS Mao et al. (2024). For each subset $m \in \{1, 2\}$, we use EdgeConv Wang et al. (2019) on the k-nearest-neighbor graph to extract local features, and concatenate them with the input coordinates to form the augmented representation $\mathbf{U}^{(m)}$. This augmentation provides local surface context beyond raw coordinates, which helps the following dual-stream factorization distinguish stable geometric structure from nuisance variation. We then split the augmented representation into a structure part and a nuisance part:

$$(\mathbf{U}_s^{(m)}, \mathbf{U}_n^{(m)}) = \mathrm{Split}\Big(\mathbf{U}^{(m)}; C_s, C_n\Big), \tag{11}$$

where $C_s + C_n = 3 + C$, $\mathbf{U}_s^{(m)} \in \mathbb{R}^{M\times C_s}$, and $\mathbf{U}_n^{(m)} \in \mathbb{R}^{M\times C_n}$. We keep C_n small to restrict nuisance capacity and use $C_n = 6$ in all experiments.

The dual-stream consists of two coupled invertible mappings: a nuisance stream F_n with $L_n = 4$ blocks and a structure stream F_s with $L_s = 8$ blocks conditioned on the nuisance code, applied in an interleaved manner. Before the ℓ-th block of F_s, the current nuisance state $\mathbf{H}_n^{(m,\ell)}$, i.e., the output of the first $\min(\ell, L_n)$ blocks of F_n, is summarized into a global condition:

$$\mathbf{Z}_n^{(m)} = F_n\Big(\mathbf{U}_n^{(m)}\Big), \tag{12}$$

$$\mathbf{c}^{(m,\ell)} = \psi\Big(\mathrm{pool}(\mathbf{H}_n^{(m,\ell)})\Big) \in \mathbb{R}^d, \tag{13}$$

where $\mathrm{pool}(\cdot)$ is permutation-invariant mean pooling over points and $\psi(\cdot)$ is a two-layer MLP shared across all blocks (we set $d = 16$). The condition is broadcast to all points inside the coupling networks of the ℓ-th block of F_s, yielding the conditioned structure encoding

$$\mathbf{Z}_s^{(m)} = F_s\Big(\mathbf{U}_s^{(m)}; \{\mathbf{c}^{(m,\ell)}\}_{\ell=1}^{L_s}\Big). \tag{14}$$

For denoising, the inverse transformation is applied with the canonical nuisance code $\bar{\mathbf{Z}}_n = \mathbf{0}$, from which the conditions are recomputed. Both streams are implemented with standard invertible coupling blocks following PD-Flow Mao et al. (2022), including actnorm Kingma & Dhariwal (2018), invertible channel mixing, and affine coupling transforms Dinh et al. (2016).

A.3 Implementation Details

We implement our method in PyTorch with PyTorch Lightning. Our model is trained with Adam using an initial learning rate of 2×10^{-3} and a ReduceLROnPlateau scheduler with patience 10, decay factor 0.5, and minimum learning rate 10^{-6}. We train for 100 epochs with batch size 10, 32-bit precision, gradient clipping of 10^{-3}, and random seed 2025. Each training patch contains 2048 points, from which our coupled sampler constructs two disjoint 1024-point subsets. Point clouds are normalized to the unit sphere; during training, the standard deviation of the Gaussian noise is sampled from $[0.005, 0.02]$, with random scaling in $[0.8, 1.2]$ and random rotations. For the coupled sampler, we set the target ratio to 0.5, producing two 1024-point subsets from each 2048-point patch. We first sample at most one pair from each voxel with voxel size 0.03 to encourage balanced spatial coverage. The remaining unmatched points are then paired by greedy kNN completion with $k = 32$ until the target number of pairs is reached. The training objective uses output-space EMD geometry consistency, Charbonnier displacement regularization ($\phi(t) = \sqrt{t^2 + \delta^2}$ with $\delta = 10^{-3}$), and latent-structure consistency, with loss weights $\lambda_{\text{geom}} = 4.0 \times 10^1$, $\lambda_{\text{disp}} = 3.0 \times 10^2$, and $\lambda_{\text{lat}} = 1.0 \times 10^3$. Validation and testing are performed patch-wise with patch size 1024, and the checkpoint with the lowest validation CD is used for final evaluation. All experiments for our method were conducted on a single NVIDIA RTX 3090 Ti GPU with 24 GB memory. Training our model for 100 epochs on the PU-Net training set took approximately 21 hours.

A.4 Evaluation Metrics

To quantitatively evaluate denoising quality, we employ two widely used geometric metrics, namely Chamfer Distance (CD) and Point-to-Mesh Distance (P2M). Among them, CD measures the average bidirectional discrepancy between the denoised point cloud and the ground-truth point set, and P2M evaluates the distance from the denoised points to the underlying reference surface.

1) Point-to-Point Metric.
Let $S_c = \{x_i\}_{i=1}^{|S_c|}$ denote the ground-truth point cloud and $S_p = \{y_i\}_{i=1}^{|S_p|}$ denote the denoised prediction. Since some methods may remove or retain different numbers of points during filtering, the cardinalities of S_c and S_p are not necessarily identical. Here, $\|x - y\|_2$ denotes the Euclidean distance between two points in $\mathbb{R}^3$, and $\min_{y \in S_p} \|x - y\|_2$ represents the nearest-neighbor distance from a point $x \in S_c$ to the set S_p.

$$\begin{aligned} e_{\text{CD}}(S_c, S_p) = &\frac{1}{|S_c|} \sum_{x \in S_c} \min_{y \in S_p} \|x - y\|_2^2 \\ &+ \frac{1}{|S_p|} \sum_{y \in S_p} \min_{x \in S_c} \|y - x\|_2^2 \end{aligned} \tag{15}$$

2) Point-to-Mesh Metric.
In addition to point-set based evaluation, we further adopt Point-to-Mesh Distance (P2M) to measure the geometric consistency between the denoised point cloud and the ground-truth mesh surface. This metric considers both the fidelity of predicted points to the reference mesh and the coverage of the mesh by the predicted point set. Specifically, for each denoised point $y \in S_p$, we compute its squared distance to the closest point on the mesh surface $\mathcal{M}$. Conversely, for each face $f \in \mathcal{M}$, we also measure its distance to the nearest point in S_p. The final P2M score is obtained by summing these two terms:

$$\begin{aligned} \text{P2M}(\mathcal{M}, S_p) = &\frac{1}{|S_p|} \sum_{y \in S_p} \min_{f \in \mathcal{M}} \left\| y - \text{proj}_f(y) \right\|_2^2 \\ &+ \frac{1}{|\mathcal{M}|} \sum_{f \in \mathcal{M}} \min_{y \in S_p} \left\| \text{proj}_f(y) - y \right\|_2^2 . \end{aligned} \tag{16}$$

Table 4: Quantitative comparison with supervised point cloud denoising methods on the PU-Net test set with 10K and 50K input points under Gaussian noise levels of 1%, 2%, and 3%. Chamfer Distance (CD) and Point-to-Mesh Distance (P2M) are multiplied by 10^4. The lowest and second-lowest values in each column are highlighted in **bold** and underlined, respectively.

Resolution	**10K (Sparse)**						**50K (Dense)**					
Noise level	**1%**		**2%**		**3%**		**1%**		**2%**		**3%**	
Method	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**
DeepPSR	3.425	1.200	3.761	**1.210**	5.291	2.234	0.768	0.390	0.970	**0.508**	2.610	1.675
ScoreDenoise	2.597	**0.827**	3.704	1.404	4.896	2.397	0.747	0.426	1.274	0.821	2.033	1.381
DMRDenoise	3.220	1.003	4.105	1.679	6.450	3.594	0.838	0.486	1.722	1.191	4.355	3.514
Pointfilter	2.602	0.843	3.655	1.242	5.099	2.187	0.896	0.537	1.036	0.595	**1.590**	**0.982**
PointCleanNet	3.687	1.600	7.931	4.764	13.370	9.586	1.071	0.626	2.036	1.426	5.710	4.769
Ours	**2.296**	0.920	**3.179**	1.402	**3.798**	**1.848**	**0.606**	**0.369**	**0.843**	0.562	2.019	1.492

Table 5: Sensitivity analysis on PU-Net (50K). Each row changes one setting of the default configuration (C_n=6, patch size 2048, h=0.03, k_f=32, r=1). CD and P2M are scaled by 10^4. Best and second-best results are in **bold** and underlined.

	1%		**2%**		**3%**	
Setting	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**
Default (Ours)	**0.606**	**0.369**	**0.843**	**0.562**	**2.019**	**1.492**
Nuisance-stream dimension						
$C_n = 3$	0.628	0.384	0.920	0.636	2.256	1.695
$C_n = 12$	0.620	0.379	1.252	0.863	3.171	2.485
Training patch size						
1024	1.381	0.890	2.763	2.096	5.915	4.947
4096	0.786	0.547	1.338	0.944	3.080	2.420

A.5 Comparison with Supervised Methods

Table 4 compares our method with supervised approaches. Despite being trained without clean ground truth, our method achieves the lowest CD across all three noise levels in the 10K setting, together with the lowest P2M under 3% noise. In the 50K setting, it achieves the lowest CD and P2M under 1% noise and the lowest CD under 2% noise, while obtaining the second-lowest CD under 3% noise.

A.6 More Ablation Studies and Analyses

Ablation on the Nuisance-Stream Dimension

The augmented representation contains $3 + C$ channels, where the additional three channels correspond to the 3D coordinates. The representation is divided into a structure stream and a nuisance stream according to

$$C_s + C_n = 3 + C,$$

where C_s and C_n denote the dimensions of the structure and nuisance streams, respectively. In our implementation, we set $C_n = 6$, while all remaining channels are assigned to the structure stream, i.e.,

$$C_s = (3 + C) - 6.$$

The nuisance stream is intentionally kept compact to reduce the possibility of encoding excessive structural information. To investigate the effect of its dimension, we evaluate different choices of C_n under multiple noise levels. The results are reported in Table 5.

As shown in Table 5, setting $C_n = 6$ consistently achieves the best performance across all evaluated noise levels. A smaller nuisance dimension ($C_n = 3$) provides insufficient capacity to represent

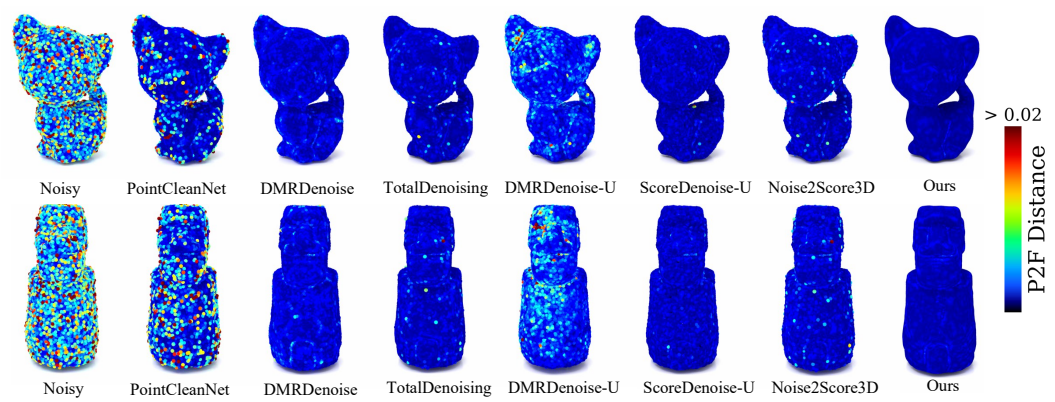

Figure 5: Visual comparison of denoising results on moai and kitten.

Table 6: Comparison with Noise2NoiseMapping on the PU-Net test set under 2% noise with 10K input points. CD and P2M are scaled by 10^4. Lower values indicate better performance.

Method	**CD↓**	**P2M↓**	**Avg. Time (s)↓**
Noise2NoiseMapping	3.923	1.737	506
Ours	**3.179**	**1.402**	**1.052**

noise-related variations, while increasing the dimension to $C_n = 12$ leads to a noticeable degradation, particularly under stronger noise. This suggests that an overly large nuisance stream may capture structural information that should instead be preserved in the structure stream. Therefore, we use $C_n = 6$ as a balanced configuration throughout our experiments.

Comparison with Noise2NoiseMapping

We further compare our method with Noise2NoiseMapping on the PU-Net test set. Although Noise2NoiseMapping can operate on a single noisy point cloud, it follows a per-shape optimization paradigm, where an implicit signed distance function (SDF) is optimized separately for each test shape. Following our setting, we run its official implementation with a single noisy observation per shape. In contrast, our method is trained at the dataset level and directly denoises unseen point clouds through a single forward inference, without requiring test-time per-shape optimization.

As shown in Table 6, our method achieves lower CD and P2M errors than Noise2NoiseMapping while requiring substantially less inference time. Specifically, Noise2NoiseMapping requires approximately 506 s to optimize the implicit representation for each 10K-point input, whereas our method requires only 1.052 s for direct inference, corresponding to an approximately $481\times$ reduction in inference time.

The large efficiency gap mainly stems from the different inference paradigms. Noise2NoiseMapping performs test-time optimization independently for each input shape, while our method learns a dataset-level denoising prior that can be directly applied to previously unseen point clouds. These results demonstrate that our method provides both improved denoising accuracy and substantially higher inference efficiency in this setting.

Validation of Latent Factorization

To further investigate whether the learned latent representation achieves the intended separation between geometric structure and nuisance information, we reconstruct a single test shape (Camel, PU-Net, 10K points) using different combinations of the latent variables. Specifically, given the structure representation Z_s and nuisance representation Z_n, we evaluate the reconstructions obtained with the inverse mapping $f^{-1} = F_\theta^{-1}$ from $f^{-1}(Z_s, Z_n)$, $f^{-1}(Z_s, 0)$, and $f^{-1}(0, Z_n)$. As shown in Table 7, reconstruction using the full latent representation $f^{-1}(Z_s, Z_n)$ closely reproduces the

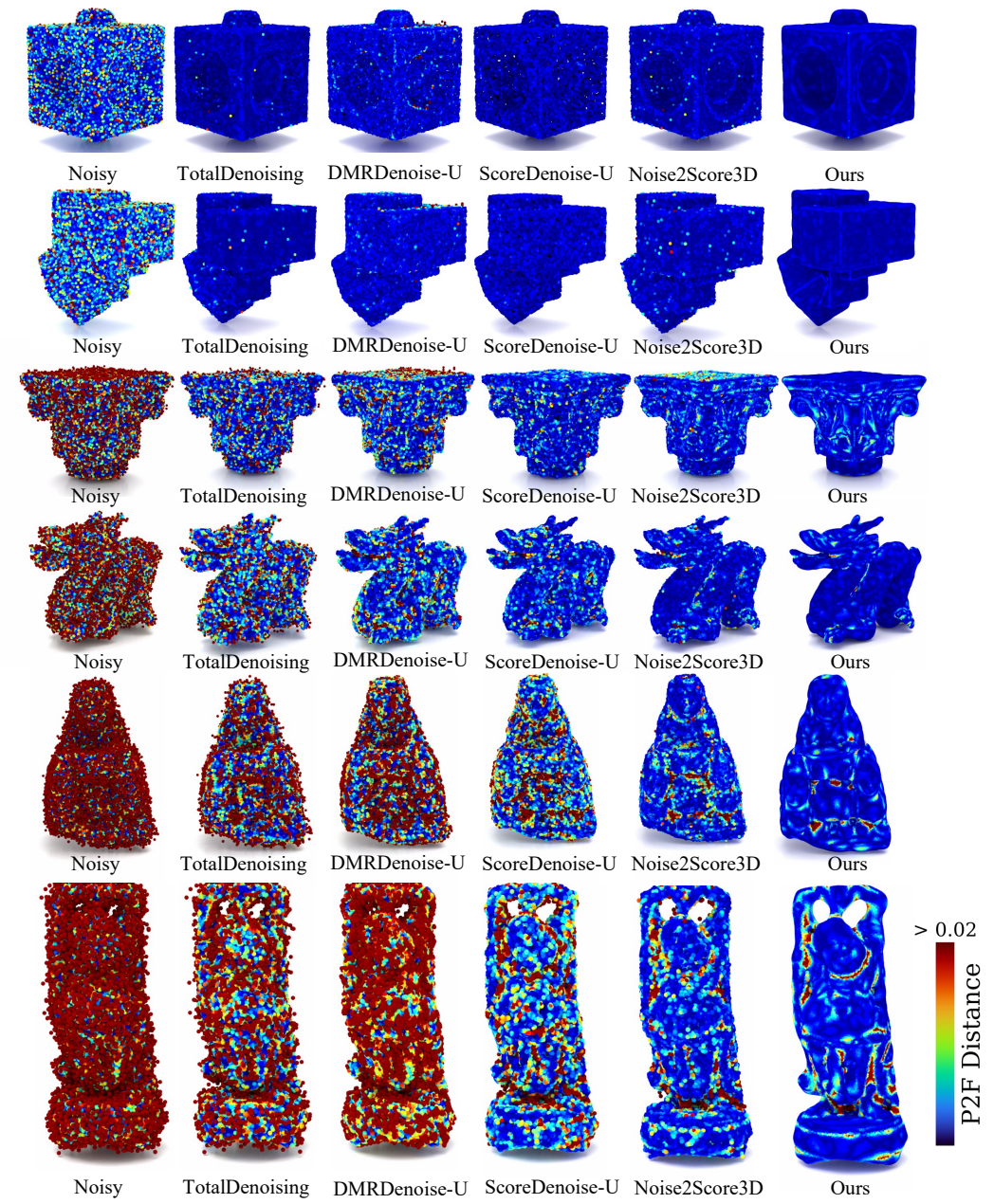

Figure 6: Visual comparison of denoising results on 1% (box_push and boxunion2), 2% (column_head and dragon), and 3% (galera and happy).

noisy input, indicating that the latent representation preserves sufficient information for faithful reconstruction.

More importantly, removing the nuisance component Z_n and reconstructing from Z_s alone consistently improves both CD and P2M across all evaluated noise levels. Specifically, the CD decreases from 2.732 to 1.468 at 1% noise, from 6.152 to 2.174 at 2% noise, and from 11.002 to 2.943 at 3% noise. The P2M metric exhibits the same trend, decreasing from 1.581 to 0.921, from 4.620 to 1.579, and from 9.268 to 2.242, respectively. These results suggest that Z_s retains the dominant geometric information while suppressing a substantial portion of the nuisance-related variations.

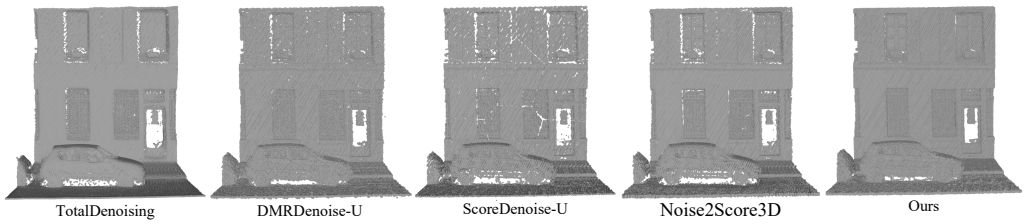

Figure 7: Visual comparison of denoising results on Rue_6.

Table 7: Decoding different parts of the latent code of a single test shape (Camel, PU-Net, 10K points): the full code $f^{-1}(Z_s, Z_n)$, the structure code alone $f^{-1}(Z_s, 0)$, and the nuisance code alone $f^{-1}(0, Z_n)$. CD and P2M are scaled by 10^4; the best result in each column is in **bold**. Values are for one shape and therefore differ from the dataset averages in Table 1.

	1%		2%		3%	
Decoding	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**	**CD↓**	**P2M↓**
Full code	2.732	1.581	6.152	4.620	11.002	9.268
$\mathbf{Z}_s$ only	**1.468**	**0.921**	**2.174**	**1.579**	**2.943**	**2.242**
$\mathbf{Z}_n$ only	172.4	145.3	200.3	176.0	280.5	238.7

In contrast, reconstruction using only the nuisance representation, $f^{-1}(0, Z_n)$, produces substantially larger CD and P2M errors and fails to recover the underlying geometric structure. This is expected, since the coordinates are carried by the structure stream and Z_n enters the reconstruction only through global conditions. Taken together, these results confirm that the latent representation is invertible and that Z_s primarily preserves geometric structure.

Hyperparameter Sensitivity Analysis

We evaluate the sensitivity of our method to several key hyperparameters, including the voxel size, the fallback neighbor number k_f, and the maximum number of pairs. All experiments are conducted on the PU-Net 50K setting under 2% noise. As shown in Table 8, the default configuration achieves the lowest CD and P2M among all evaluated settings. The performance remains relatively stable when the voxel size is slightly reduced to 0.015, while increasing the voxel size to 0.06 leads to a more noticeable degradation. This suggests that excessively coarse voxelization may discard useful local geometric details. The method also shows sensitivity to the fallback neighbor number k_f. Both $k_f = 16$ and $k_f = 64$ result in higher CD and P2M than the default configuration, indicating that an appropriate neighborhood size is important for balancing local geometric context and cross-structure interference. Similarly, increasing the maximum number of pairs per voxel from 1 to 2 degrades performance, suggesting that limiting the pairs per voxel keeps the spatial coverage of the two subsets balanced.

Effect of Patch Size

We further investigate the influence of patch size on denoising performance. Specifically, we evaluate three patch sizes, i.e., 1024, 2048 (Ours), and 4096 points, under different noise levels. As shown in Table 5, a patch size of 2048(Ours) consistently achieves the best CD and P2M results across all evaluated noise levels. Reducing the patch size to 1024 leads to a noticeable performance degradation, particularly under stronger noise. For example, at the 3% noise level, the CD increases from 2.019 with 2048 points to 5.915 with 1024 points. This suggests that an excessively small patch may provide insufficient local geometric context for reliably distinguishing underlying structures from noise perturbations. Increasing the patch size to 4096 also results in lower performance than the 2048-point configuration, although it remains substantially better than the 1024-point setting in most cases. A larger patch covers a broader spatial region and may introduce more heterogeneous geometric structures into the same local processing unit, making local structure modeling more challenging.

Table 8: Hyperparameter sensitivity on the PU-Net 50K setting under 2% noise. Lower CD and P2M values indicate better denoising performance.

Setting	**CD** $\downarrow$	**P2M** $\downarrow$
Default	**0.843**	**0.562**
Voxel size $= 0.015$	0.899	0.613
Voxel size $= 0.06$	1.112	0.783
$k_f = 16$	1.326	0.914
$k_f = 64$	1.280	0.884
Maximum pairs $= 2$	1.107	0.738

A.7 More Visual Comparisons

Figures 5, 6, and 7 provide additional visual comparisons of denoising results. Specifically, Figure 5 shows the denoising results on moai and kitten under 50000-point sampling with a noise level of 0.01. Figure 6 shows the results on 1%(box_push and boxunion2), 2%(column_head and dragon), and 3%(galera and happy) under 50000-point sampling. Figure 7 presents results on RueMadame_6.