U-CAN: Unsupervised Point Cloud Denoising with Consistency-Aware Noise2Noise Matching

Junsheng Zhou[1*] **Xingyu Shi**[1*] **Haichuan Song**[2†] **Yi Fang**[3]
Yu-Shen Liu[1†] **Zhizhong Han**[4]

School of Software, Tsinghua University, Beijing, China [1]
Computer Science and Technology, East China Normal University, Shanghai, China [2]
Center for AI and Robotics (CAIR), NYU Abu Dhabi, UAE [3]
Department of Computer Science, Wayne State University, Detroit, USA[4]

{zhou-js24,shixy22}@mails.tsinghua.edu.cn hcsong@cs.ecnu.edu.cn
yfang@nyu.edu liuyushen@tsinghua.edu.cn h312h@wayne.edu

Abstract

Point clouds captured by scanning sensors are often perturbed by noise, which have a highly negative impact on downstream tasks (e.g. surface reconstruction and shape understanding). Previous works mostly focus on training neural networks with noisy-clean point cloud pairs for learning denoising priors, which requires extensively manual efforts. In this work, we introduce **U-CAN**, an **U**nsupervised framework for point cloud denoising with **C**onsistency-**A**ware **N**oise2Noise matching. Specifically, we leverage a neural network to infer a multi-step denoising path for each point of a shape or scene with a noise to noise matching scheme. We achieve this by a novel loss which enables statistical reasoning on multiple noisy point cloud observations. We further introduce a novel constraint on the denoised geometry consistency for learning consistency-aware denoising patterns. We justify that the proposed constraint is a general term which is not limited to 3D domain and can also contribute to the area of 2D image denoising. Our evaluations under the widely used benchmarks in point cloud denoising, upsampling and image denoising show significant improvement over the state-of-the-art unsupervised methods, where U-CAN also produces comparable results with the supervised methods. Project page: https://gloriasze.github.io/U-CAN/.

1 Introduction

3D point clouds have been a fundamental representation in 3D computer vision and play a key role in autonomous driving [13], augmented/virtual reality [56] and robotics [10]. While in these real world applications, the point clouds captured with scanning sensors (e.g. LiDAR) contain unavoidable noise, which leads to large errors in 3D perception and understanding. Recent learning-based approaches [27, 26, 7] have shown convincing results in denoising point clouds with neural networks by learning denoising patterns with noisy-clean point cloud pairs. However, they are limited in the amount of clean 3D geometries which require manual efforts of human 3D CAD modelling.

A straightforward observation is that despite the limited clean models, the amount of real-captured noisy point clouds is growing rapidly everyday with the LiDARs in self-driving cars or consumer level digital devices in our daily life, such as iPhone. Consequently, it is desirable to learn denoising patterns by solely using the noisy data itself. The subsequent approaches, such as TotalDenoising [12], therefore turn to explore unsupervised point cloud denoising by leveraging a spatial prior term

[*] Equal contribution. [†] Corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

for total-level denoising. However, the current unsupervised approaches still struggle to predict precise clean point cloud while keeping high-fidelity local geometries due to the lack of sufficient constraints at local-level.

To solve these issues, we introduce U-CAN, an unsupervised learning framework for point cloud denoising with consistency-aware noise to noise matching. Instead of predicting a total-level denoising, we leverage a neural network to infer a multi-step denoising path for each point of a shape or scene with a point-wise noise to noise matching scheme. Specifically, we learn a mapping from one noisy point cloud to another with a novel loss function which enables a point-to-point matching for investigating denoising patterns from only noisy point clouds. The key idea of this noise to noise matching is to leverage the statistical reasoning to reveal the clean structures upon its several noisy observations.

Another challenge in predicting robust denoising arises from the unknown location of true surfaces when only noisy observations are available. This ambiguity can lead to unstable convergence due to inconsistencies in denoising results across different noisy observations. In response to this challenge, we introduce a novel consistency-aware constraint that specifically targets the denoising geometric consistency. We achieve this by minimizing the geometric differences from the denoising prediction of one noisy observation to the prediction of another. Furthermore, we demonstrate that the proposed consistency-aware denoising constraint is not limited to the 3D domain, which can also significantly contribute to the field of 2D image denoising. Extensive experiments demonstrate that the proposed U-CAN outperforms state-of-the-art methods in unsupervised point cloud denoising, upsampling and image denoising, where U-CAN even achieves comparable performances with the supervised methods. Our main contributions can be summarized as follows.

- We introduce U-CAN, a novel framework for unsupervised point cloud denoising by designing a neural network to infer a multi-step denoising path for each point of a noisy observation with a novel noise to noise matching loss.
- We propose a general constraint on the denoising geometric consistency across different denoising predictions, which significantly improves the denoising performance in both 3D and 2D domain.
- We achieve state-of-the-art results in unsupervised denoising for both point clouds and images under widely used benchmarks, while also delivering performance comparable to supervised methods. U-CAN is also capable of unsupervised point upsampling through denoising.

2 Related Work

2.1 Traditional Point Cloud Denoising

Point cloud denoising plays a key role in robust 3D understanding, as the point clouds captured by scanners often contain unavoidable noise. Traditional methods for optimizing noisy point clouds include local surface fitting [1, 14], sparse representation [3, 44], and graph filtering [41, 55], all of which use geometric priors for denoising. Local surface fitting methods, such as the widely used moving least squares (MLS) method [2] and its robust extensions [35, 9], approximate the point cloud with a smooth surface using simple function approximators and project the points onto this newly formed surface for denoising. Other techniques, like jet fitting [5] and the parameterization-free local projector operator (LOP) [25, 14], have also been developed for point cloud denoising. Sparsity-based methods [3, 44] address denoising by initially predicting surface normals through optimization problems with sparse constraints. Graph-based methods [41, 55] represent point clouds using graphs and use graph filters for denoising. The graph-based methods are sensitive to the noise distributions due to the potential instability in graph construction.

2.2 Learning-based Point Cloud Denoising

The deep learning based approaches for 3D point cloud [46, 67, 74, 54, 49, 18, 64, 16, 73, 75, 57, 62] have largely advanced point cloud processing tasks, such as upsampling [51, 24], surface reconstruction [71, 72, 4, 42, 29, 65, 11, 70, 15, 69, 34, 61], consolidation [33, 52], normal estimation [23, 63], generation [76, 68, 77, 48, 66] and denoising [40, 36, 27, 26, 19, 31, 58, 47, 60]. With the emergence

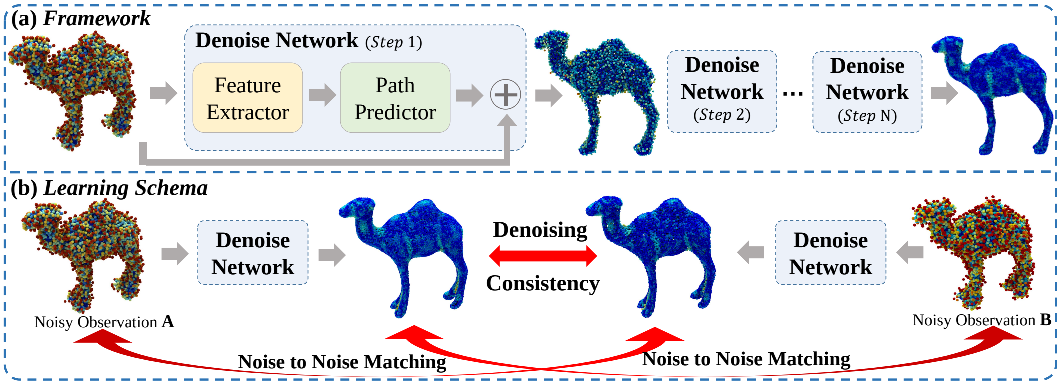

Figure 1: Overview of our method. (a) We design a multi-step denoising framework to gradually filter the noisy point cloud. (b) We introduce a novel learning schema for unsupervised learning of point cloud denoising by proposing two constraints, i.e., Noise to Noise Matching loss and Denoising Consistency loss.

of neural networks for point cloud processing, such as PointNet [37], PointNet++ [38] and DGCNN [46], learning representations on point sets for denoising has achieved convincing performances. PointCleanNet [40] is the pioneer of learning-based point denoising which introduces a neural network based on the PointNet model [37, 38]. PointFilter [59] advances PointCleanNet with a special designed filter modeling. The following work GPDNet [36] introduces graph convolutional networks for improving and stabilizing denoising process, introducing the strong capability of graph networks in handling complex geometric data to the point cloud denoising task. DMRDenoise [26] introduces a new perspective for denoising by adopting an innovative downsample-upsample framework. More recently, ScoreDenoise [27] attempts to estimate the gradient field around each point and iteratively update the position of each point. IterativeFPN [7] simulates a real iterative filtering process internally to reduce noise.

2.3 Unsupervised Point Cloud Denoising

Previous learning-based approaches merely focus on learning denoising patterns with noisy-clean point cloud pairs and are limited in the amount of clean 3D shapes which require manual efforts of human CAD modeling. TotalDenoising [12] is the most relevant work of ours which explores unsupervised point denoising by leveraging a spatial prior term for total-level denoising. However, it struggles to predict precise predictions with high-fidelity local geometries. The reason is that TotalDenoising only involves the global constraint and lacks the local-level constraint which plays the key role in producing detailed predictions. DMRDenoise [26] and ScoreDenoise [27] also provide an unsupervised version by introducing the total-denoising loss, but both of them face the same problem as TotalDenoising [12]. A recent work [28] introduces an unsupervised approach to over-fit each noisy point cloud for learning signed distance functions, where each point cloud takes about more than 10 minutes to converge. We focus on the learning-based point cloud denoising which enables a fast inference.

Different from these works, we learn a data-driven matching from one noisy point cloud to another with a novel loss function which enables point-to-point matching at local-level. This brings high-fidelity denoising results. We further introduce denoising consistency constraint for consistency-aware predictions with improved accuracy.

3 Architecture of U-CAN

Problem Statement. We design a neural network with a novel learning schema for unsupervised point cloud denoising. Current methods train neural networks to denoise a point cloud by matching it with its paired clean point cloud. Different from these supervised methods, we do not require any clean point clouds as supervision, and learn to filter a noisy observation $\mathcal{P}_a$ of a 3D shape or scene $\mathcal{S}$ with only other several noisy observation $\mathcal{P}_b$ of $\mathcal{S}$.

Overview. The overview of proposed U-CAN is shown in Fig. 1. We will start from our denoise network in Sec. 3.1 and introduce the noise to noise mapping schema with a novel point-wise matching loss in Sec. 3.2. We then present a novel constraint on denoising consistency in Sec. 3.3 and transfer it to enhance the unsupervised image denoising task in Sec. 3.4.

3.1 Denoise Network

Given a noisy point cloud $\mathcal{P}_a$ as input, we design a multi-step denoising framework to gradually filter $\mathcal{P}_a$ for achieving a cleaned point cloud $\mathbb{C}_a$. As illustrated in Fig. 1 (a), the *Denoise Network* at each step consists of a *Feature Extractor* and a *Path Predictor*. We implement the *Feature Extractor* as a series of dynamic EdgeConv from DGCNN [46] with residual connections for achieving robust representations, while the *Path Predictor* is composed with several linear layers to predict the moving path for each point from the extracted features.

During the i-th step of the denoising process, the Denoise Network f_i takes the filtered point clouds $\mathcal{C}_a^{i-1}$ from the previous step (the noisy point cloud $\mathcal{P}_a$ for the initial step) as input. It then predicts a distinct moving path $\triangle p_i$ for pulling each point to attain the filtered point cloud $\mathcal{C}_a^i$ at the current step as $\mathcal{C}_a^i = \mathcal{C}_a^{i-1} + \triangle p_i$. The final prediction $\mathbb{C}_a$ is obtained by moving $\mathcal{P}_a$ gradually, formulated as:

$$\mathbb{C}_a = \mathcal{P} + f_1(\mathcal{P}_a) + \sum_{i=2}^{N} f_i(\mathcal{C}_a^{i-1}), \tag{1}$$

where $N > 1$ is the number of steps.

3.2 Noise to Noise Matching

The common practice for predicting the clean point cloud from its noisy observations is to train a neural network with Noise to Clean Matching supervisions to minimize the distance between the predicted $\mathbb{C}_a$ with the ground truth clean point cloud G_a, formulated as:

$$\mathcal{L}_{N2C} = \mathcal{D}(\mathbb{C}_a, G_a), \tag{2}$$

where $\mathcal{D}(\cdot, \cdot)$ is a distance metric, typically the Chamfer Distance.

Preview Noise2Noise. Previously, Noise2Noise [22] has been proposed for unsupervised image denoising by encouraging a denoised image to resemble other noisy observations of the same image. Given the appealing results in 2D domain, it seems that we can denoise 3D point cloud unsupervisedly by simply transferring the success of Noise2Noise into 3D domain. However, the conclusion of Noise2Noise is built upon the one-to-one matching correspondences, as the pixels in images. The correspondences support the key assumption of Noise2Noise that the noisy values at the same pixel location of different observations are random realizations of a distribution around a clean pixel value. While the point clouds are irregular and unordered with no correspondences where a naive reproduction of Noise2Noise do not work for 3D point clouds.

A balancing approach for adapting unsupervised denoising in point clouds is to design a total-level loss for global denoising without specifying the correspondences like TotalDenoising [12], yet it struggles to predict precise clean point cloud while keeping local geometries with only the coarse constraint. NoiseMap [28] introduces an over-fitting approach to learn signed distance functions for each noisy observation with the Noise2Noise mechanism in local-level, but fails in generalizing to new observations.

One-to-One Point Correspondences. As discussed above, we justify that the key factor preventing the adaption of Noise2Noise schema in 3D domain is the lack of 3D correspondences. To solve this issue, we aim to build an one-to-one correspondence with a specific matching for each point. Instead of manually defining the point correspondences, we explore a suitable distance metric $\mathcal{D}$ that potentially contains the one-to-one point correspondences as the optimizing target, which satisfies the assumptions for Noise2Noise and can naturally enable the unsupervised denoising for 3D point clouds. In practice, we use Earth Moving Distance (EMD) as a suitable implementation of $\mathcal{D}$. The EMD between two point clouds X and Y is formulated as:

$$D_{\text{EMD}}(X, Y) = \min_{\phi: X \to Y} \sum_{x \in X} ||x - \phi(x)\|_2, \tag{3}$$

where ϕ is a one-to-one correspondence. With the Earth Moving Distance which potentially contains the point correspondences as the distance metric, we successfully adopt the Noise2Noise schema for unsupervised point cloud denoising. Specifically, as shown in Fig. 1 (b), given two noisy observations $\mathcal{P}_a$ and $\mathcal{P}_b$ randomly selected from a set of noisy observations at each epoch, we train the *Denoise Network* with the Noise to Noise Matching loss to push the denoised point cloud $\mathbb{C}_a$ to be similar to another noisy point cloud $\mathcal{P}_b$, and vice versa for $\mathbb{C}_b$. The Noise to Noise Matching loss is formulated as:

$$\mathcal{L}_{N2N} = \mathcal{D}_{\mathrm{EMD}}(\mathbb{C}_a, \mathcal{P}_b) + \mathcal{D}_{\mathrm{EMD}}(\mathbb{C}_b, \mathcal{P}_a). \tag{4}$$

With this loss, U-CAN leverages the statistical reasoning among multiple noisy observations and effectively infers clean structures.

3.3 Denoising Consistency Constraint

Another issue that none of the previous unsupervised denoising works on 2D or 3D domains noticed is that the unsupervised noise to noise matching schema struggles to produce a consistent denoising prediction with different noisy observations as input. This leads to ambiguous optimizations for the detailed geometries. For the supervised approaches, this is not a problem since each noisy input has a distinct clean point cloud as the target. While in the situation of unsupervised denoising, there is no true surface locations provided, and only multiple noisy observations are available as inputs and targets, which makes it hard for the neural networks to learn consistent predictions.

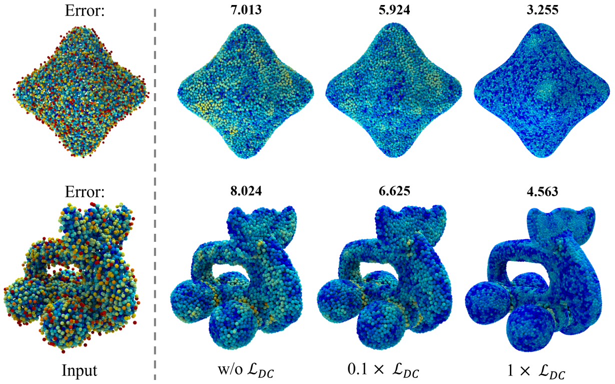

Figure 2: Illustrations on the effect of proposed constraint on denoising consistency. The noise errors indicate the Chamfer distance between the denoised and the clean point clouds.

Driven by this observation, we propose a novel constraint on denoising consistency for learning consistency-aware denoising patterns. Specifically, we push the denoised prediction of one noisy observation to be consistent with the denoised prediction of another noisy observation with a special designed loss, formulated as:

$$\mathcal{L}_{\mathrm{DC}} = \mathcal{D}_{\mathrm{EMD}}(\mathbb{C}_a, \mathbb{C}_b), \tag{5}$$

where $\mathbb{C}_a$ and $\mathbb{C}_b$ are the denoised predictions achieved by Eq. (1), respectively. With the simple but effective term, U-CAN is able to produce more consistent-aware predictions and further improve the denoising results at detailed local geometries.

We provide an illustration as shown in Fig. 2 to show the advantage of our proposed constraint $\mathcal{L}_{\mathrm{DC}}$. We train U-CAN for learning point cloud denoising without $\mathcal{L}_{\mathrm{DC}}$ and show the result as "w/o $\mathcal{L}_{\mathrm{DC}}$". We than visualize the denoising predictions of U-CAN trained with low coefficient (i.e. 0.1×) and high coefficient (i.e. 1×) of $\mathcal{L}_{\mathrm{DC}}$, shown as "0.1× $\mathcal{L}_{\mathrm{DC}}$" and "1× $\mathcal{L}_{\mathrm{DC}}$". The comparisons demonstrate the effectiveness of $\mathcal{L}_{\mathrm{DC}}$ where better performances are achieved with larger coefficient of $\mathcal{L}_{\mathrm{DC}}$.

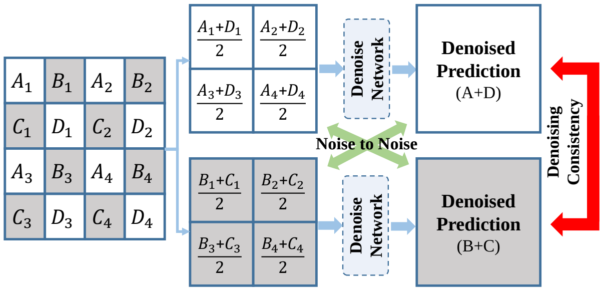
A_1 B_1 A_2 B_2
C_1 D_1 C_2 D_2
A_3 B_3 A_4 B_4
C_3 D_3 C_4 D_4
$\frac{A_1+D_1}{2}$ $\frac{A_2+D_2}{2}$
$\frac{A_3+D_3}{2}$ $\frac{A_4+D_4}{2}$
$\frac{B_1+C_1}{2}$ $\frac{B_2+C_2}{2}$
$\frac{B_3+C_3}{2}$ $\frac{B_4+C_4}{2}$
Denoise Network
Denoise Network
Noise to Noise
Denoised Prediction (A+D)
Denoised Prediction (B+C)
Denoising Consistency

Figure 3: Transferring the denoising consistency constraint of U-CAN to the unsupervised image denoising.

Table 1: Denoising comparisons under PUNet dataset. CD$\times 10^4$ and P2M $\times 10^4$.The best results under the unsupervised (Un-Sup) point cloud denoising setting are highlighted.

	Point Number	10K(Sparse)						50K(Dense)					
	Noise	1%		2%		3%		1%		2%		3%	
	Model	CD	P2M	CD	P2M	CD	P2M	CD	P2M	CD	P2M	CD	P2M
Classic	Bilateral [8]	3.646	1.342	5.007	2.018	6.998	3.557	0.877	0.234	2.376	1.389	6.304	4.730
	Jet [5]	2.712	0.613	4.155	1.347	6.262	2.921	0.851	0.207	2.432	1.403	5.788	4.267
	MRPCA [32]	2.972	0.922	3.728	1.117	5.009	1.963	0.669	0.099	2.008	1.003	5.775	4.081
	GLR [55]	2.959	1.052	3.773	1.306	4.909	2.114	0.696	0.161	1.587	0.830	3.839	2.707
Supervised	PCNet [40]	3.515	1.148	7.469	3.965	13.067	8.737	1.049	0.346	1.447	0.608	2.289	1.285
	GPDNet [36]	3.780	1.337	8.007	4.426	13.482	9.114	1.913	1.037	5.021	3.736	9.705	7.998
	DMR [26]	4.482	1.722	4.982	2.115	5.892	2.846	1.162	0.469	1.566	0.800	2.632	1.528
	ScoreDenoise [27]	2.521	0.463	3.686	1.074	4.708	1.942	0.716	0.150	1.288	0.566	1.928	1.041
Un-Sup	TTD [12]	3.390	**0.826**	7.251	3.485	13.385	8.740	1.024	**0.314**	2.722	1.567	7.474	5.729
	DMR-TTD	7.897	5.026	9.257	6.119	10.946	7.569	2.137	1.567	3.223	2.498	5.572	4.669
	ScoreDenoise-TTD	3.107	0.888	4.675	1.829	7.225	3.726	0.918	0.265	2.439	1.411	5.303	3.841
	Ours	**2.497**	1.105	**3.234**	**1.255**	**3.666**	**1.842**	**0.835**	0.609	**0.975**	**0.675**	**2.479**	**1.863**

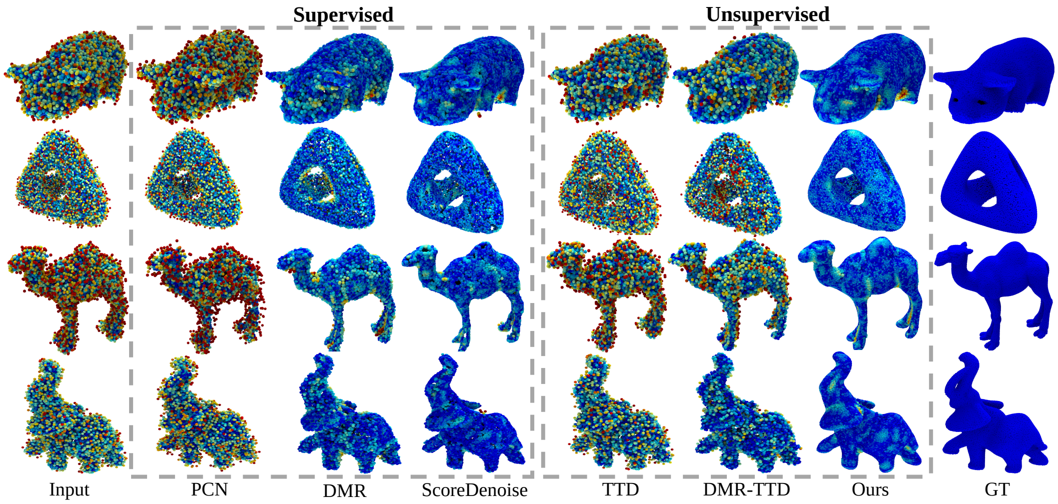

Figure 4: Visual comparisons under PUNet dataset. The noise errors at each point is shown in color, where the points closer to the ground truth surface are represented with bluer color, indicating lower error. And those with higher error are represented with redder color.

3.4 Transferring U-CAN to Image Denoising

We further justify that the observation in Sec. 3.3 is not limited in the unsupervised point cloud denoising, but is a common issue that also occurs in the unsupervised image denoising task. Therefore, we believe the proposed constraint for denoising consistency in Eq. (5) can also contribute to the area of 2D image denoising. We demonstrate the effectiveness of $\mathcal{L}_{\text{DC}}$ by adapting it to the state-of-the-art work ZS-N2N [30] on unsupervised image denoise.

The overview of modified ZS-N2N is shown in Fig. 3. ZS-N2N separates an image into two downsampled sub-images and treats them as two noisy observations to learn a residual-based noise to noise matching for image denoising. We further introduce our proposed constraint on denoising consistency to ZS-N2N by minimizing the differences between one denoised sub-image and the other denoised sub-image.

4 Experiments

4.1 Point Cloud Denoising on Synthetic Data

Dataset and Metrics. For the experiments on synthetic shapes, we follow ScoreDenoise [27] to train our network on the PUNet [53] dataset. We split the dataset into training and testing sets with the same setting as ScoreDenoise [27]. Poisson disk sampling algorithm is used to sample point clouds from

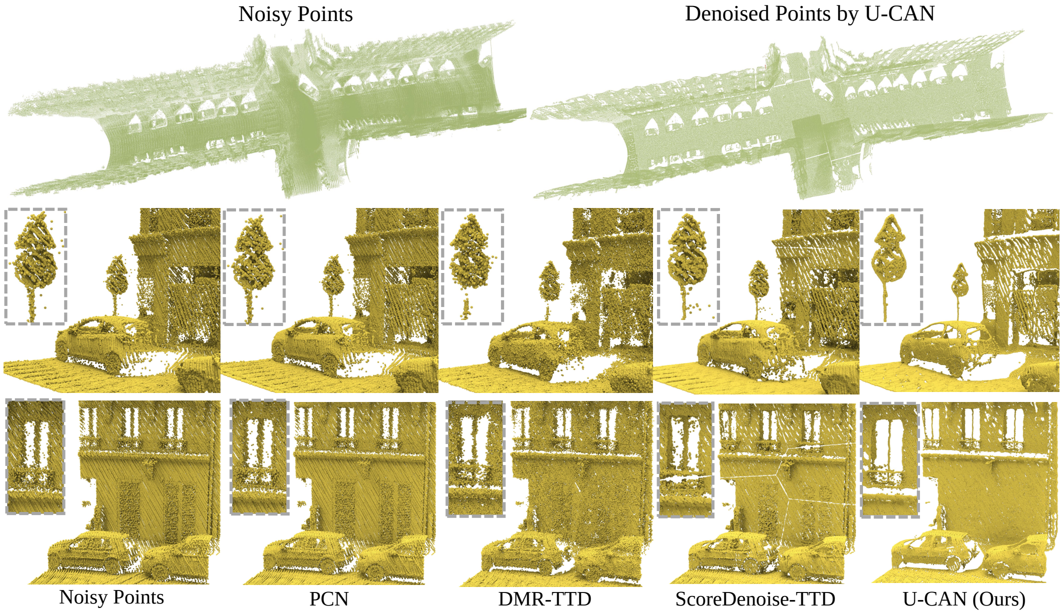

Figure 5: Denoising on the real scans under Paris-rue-Madame dataset. **Top:** The visualization of the noisy points and denoised points obtained by U-CAN under the whole scene. **Bottom**: The visual comparison with the supervised and unsupervised approaches on the local scene geometries.

the meshes at two resolutions: 10k and 50k points and the Gaussian noise is subsequently introduced at three different levels of standard deviations, i.e., 1%, 2% and 3% of the bounding sphere's radius. Following previous works PCNet [40] and DMR [26], we split point clouds into patches before being fed into the model, where the patch size is set to 1K. We evaluate the performance of U-CAN and other baselines under the commonly used metrics L2 Chamfer distance (CD) and the point-to-mesh distance (P2M), following previous methods [27, 26]

Comparisons. We quantitatively compare the proposed U-CAN with the state-of-the-art methods for both supervised and unsupervised point cloud denoising in Tab. 1. This includes classic optimization-based methods such as Bilateral [8], Jet [5], MRPCA [32], GLR [55]; supervised learning-based methods like PCNet [40], GPDNet [36], DMR [26], ScoreDenoise [27], and PointFilter [59]; and unsupervised methods including TTD [12], as well as unsupervised adaptations of DMR [26] and ScoreDenoise [27] with the TTD loss, shown as 'DMR-TTD' and 'ScoreDenoise-TTD'.

The comparative analysis of methods using synthetic data is presented in Tab. 1 and illustrated in Fig. 4. Traditional optimization-based point cloud denoising methods rely heavily on geometric priors to inform their smoothing algorithms and show increased sensitivity to noises with unseen variances, leading to degradation in denoising performance. For unsupervised denoising, the TTD [12] fails to produce high-fidelity local geometries with only the global constraints. The unsupervised versions of DMR [26] and ScoreDenoise [27] which leverage the same constraint as TTD, share same limitations of TTD and presents sub-optimal performance at both low and high resolutions due to the lack of local-level constraints.

As presented, our model significantly outperforms previous unsupervised denoising methods, especially for noises with large variances, and can even rival the results of supervised methods in the majority of cases. In particular, at the 10K resolution and under noise levels of 2% and 3%, our method outperforms all other supervised and unsupervised methods in the evaluation.

We provide the visual comparison among the state-of-the-art supervised and unsupervised point cloud denoising methods in Fig. 4. The error at each point in the point cloud is depicted in color. Points that are closer to the ground truth surface are shown in blue, indicating lower error, while those with higher error are shown in red. As shown in the figure, our results produces significantly more visual-appealing denoising results compared to other unsupervised approaches and even some supervised ones.

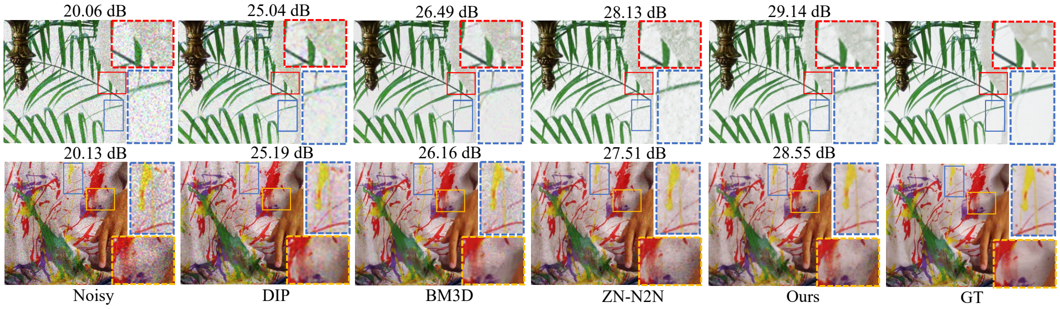

Figure 6: Visual comparison of unsupervised image denoising under McMaster18 dataset.

4.2 Point Cloud Denoising on Scanned Data

For demonstrating the capability of U-CAN to handle real-world point cloud noises, we conduct evaluations under the Paris-rue-Madame dataset [43] which is obtained from real world using laser scanners. We directly leverage the U-CAN model trained on PUNet dataset for evaluating, without requiring extra training. The visualization of denoised scene point cloud is shown in Fig. 5. Since the ground truth point cloud is not available, our evaluations are primarily qualitative, focusing on visual assessments rather than quantitative metrics.

As shown in Fig. 5, U-CAN preserves the intricate details better and yields a cleaner and smoother surface. On the scene in the top row, our method demonstrates a marked improvement over the other compared methods, particularly around complex structures like trees and cars. In the bottom row, we observe that windows are denoised with greater clarity and cleanliness with the proposed U-CAN. We also produce accurate denoising results of the surrounding structures such as the walls and vehicles.

4.3 Evaluatioins in Image Denoising

Dataset and Metrics. We further evaluate the proposed denoising consistency constraint for improving the image denoising qualities. For evaluating in the image denoising task, we follow ZS-N2N [30] to conduct experiments under the McMaster18 dataset [20]. Our evaluation setting keeps the same as ZS-N2N, and center-crop the images into patches of size 256 × 256. We examine under poisson noise with noise levels λ = 10, 25, 50. We leverage the commonly-used PSNR in dB as the evaluation metric.

Comparisons. We compare the proposed image denoising adaption of U-CAN with the state-of-the-art methods for unsupervised image denoising, including the dataset-based Noise2Clean (N2C), Neighbour2Neighbour (NB2NB) [17], Noise2Void (N2V) [21], and the dataset free methods BM3D [6], DIP [45], Self2Self (S2S) [39] and ZS-N2N [30]. We show the quantitative comparison in Tab. 2, where the denoising consistency constraint demonstrates superior performance compared to the previous methods. Specifically, by introducing the proposed denoising consistency constraint into ZS-N2N, we achieve significant improvements of nearly 1 dB over the baseline ZS-N2N. The visual comparison is shown in Fig. 6. The denoised images of U-CAN are more accurate and with more details than previous state-of-the-art methods. This is particularly clear in areas of high-frequency information, such as edges, textures, and intricate patterns, where our method maintains the integrity of these details while effectively reducing noise.

Table 2: Unsupervised image denoising under McMaster18 dataset. The PSNR scores in dB are reported. Best results are marked in bold and the second-best results are underlined.

Noise	Method			McMaster18		
			λ known?	$\lambda = 50$	$\lambda = 25$	$\lambda = 10$
Poisson	dataset-based	*N2C*	yes	29.89	28.20	26.42
			no	28.62	27.51	24.32
		NB2NB	yes	29.41	27.79	25.95
			no	28.03	27.66	24.58
		N2V	yes	27.86	25.65	23.47
			no	26.34	25.52	22.07
	dataset-free	BM3D	no	27.33	24.77	21.59
		DIP	-	28.73	27.37	24.67
		S2S	-	27.55	27.24	26.39
		ZS-N2N	-	30.36	28.41	25.75
		Ours	-	**31.03**	**29.14**	**26.52**

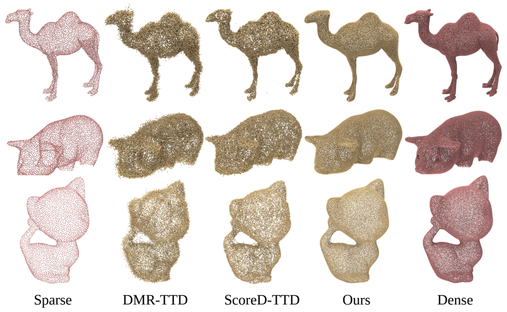

Figure 7: Visual Comparison under PU-Net.

#Points	5K		10K	
	CD	P2M	CD	P2M
PU-Net [53]	3.445	1.669	2.862	1.166
ScoreDenoise [27]	1.696	**0.295**	1.454	**0.181**
DMR-TTD [26, 12]	5.846	4.045	4.710	3.833
ScoreD-TTD [27, 12]	3.286	1.889	2.403	1.683
Ours	**1.532**	0.585	**1.212**	0.587

Table 3: Point cloud upsampling results under PU-Net dataset.

4.4 Point Upsampling via Denoising

Implementation. We further justify that the proposed U-CAN is applicable in point cloud upsampling task, without requiring sparse-dense point cloud pairs and even without requiring the clean point clouds. We follow ScoreDenoise [27] to conduct experiments in the point cloud upsampling task. Specifically, given a sparse point cloud with M points as the input, we add Gaussian noise to it for r times independently, resulting in a noisy dense point cloud containing rM points. We then feed the merged noisy point cloud to the trained U-CAN model to get the final upsampled point cloud by predicting the denoised points.

Dataset and Metrics. We follow ScoreDenoise [27] to conduct the point cloud upsampling experiments under the PU-Net dataset. We report the evaluation metrics of chamfer distance (CD) and point-to-mesh (P2M).

Comparisons. We compare our proposed U-CAN with the classical updsampling network PU-Net [53]. We further apply the adaption from denoising to upsampling to the state-of-the-art unsupervised point cloud denoising methods DMR-TTD and ScoreDenoise-TTD and report their upsampling performances. The quantitative results are shown in Tab. 3, where our method significantly outperforms DMR-TTD and ScoreDenoise-TTD, and also achieve better performance than the supervised method PU-Net designed for the upsampling task. Note that U-CAN does not require (1) sparse-to-dense point cloud pairs and (2) clean point clouds, where the only required data is the noise point clouds themselves. We further provide the visual comparison of point cloud upsampling with previous state-of-the-art unsupervised denoising methods in Fig. 7

4.5 Ablation Studies

Noise-to-Noise Matching Loss $\mathcal{L}_{N2N}$. We investigate the role of EMD-based one-to-one point correspondences in $\mathcal{L}_{N2N}$. As shown in Tab. 4, replacing EMD with CD leads to suboptimal patterns, while using Density-aware Chamfer Distance (DCD) [50] causes severe divergence. These results highlight the necessity of one-to-one matching for effective unsupervised denoising.

Denoising Consistency Loss $\mathcal{L}_{DC}$. To justify the effectiveness of constraint $\mathcal{L}_{DC}$, we remove it and vary the underlying distance metric. Without $\mathcal{L}DC$, performance significantly drops (e.g., CD

Table 4: Ablation studies on the framework and loss designs.

Dataset: PU		10K, 1%		10K, 2%		10K, 3%	
$\mathcal{L}_{N2N}$	$\mathcal{L}_{DC}$	CD	P2M	CD	P2M	CD	P2M
CD	EMD	15.48	11.36	17.42	13.22	21.39	17.01
DCD	EMD	broken	-	-	-	-	-
EMD	**EMD**	**2.497**	**1.105**	**3.234**	**1.255**	**3.666**	**1.842**
EMD	✗	2.208	0.725	3.731	1.631	7.218	4.468
EMD	CD	2.108	0.650	3.717	1.633	7.230	4.469
EMD	DCD	**2.036**	**0.608**	3.358	1.367	6.847	4.144
EMD	**EMD**	2.497	1.105	**3.234**	**1.255**	**3.666**	**1.842**

Table 5: Ablation studies on step numbers in the denoise network.

Dataset: PU	10K, 1%		10K, 2%		10K, 3%	
Ablation	CD	P2M	CD	P2M	CD	P2M
1 step	2.676	1.046	3.903	1.700	5.251	2.720
2 steps	2.606	1.159	3.507	1.670	4.096	2.069
3 steps	2.492	1.096	3.246	1.554	3.704	1.878
4 steps	2.497	1.105	3.234	1.255	**3.666**	**1.842**
5 steps	2.514	1.118	3.388	**1.151**	3.746	1.903
6 steps	2.509	1.107	**3.225**	1.235	3.753	1.857
7 steps	**2.470**	**1.080**	3.321	1.243	3.785	1.866

increases from 3.66 to 7.22 under '10K, 3%'), indicating its critical role in enforcing consistent predictions across noisy inputs. EMD again proves to be the most effective metric.

Number of Denoising Steps. We study the impact of varying the number of denoising steps N from 1 to 7. As shown in Tab. 5, performance improves up to $N{=}4$, beyond which gains saturate or slightly degrade. Thus, 4 steps offer a good trade-off between accuracy and efficiency.

5 Conclusion

In this work, we introduce **U-CAN**, an **U**nsupervised framework for point cloud denoising with **C**onsistency-**A**ware **N**oise2Noise matching. We train a neural network to infer a denoising path for each point of a shape with a noise to noise matching scheme. Our novel loss enables statistical reasoning on noisy point cloud observations. We also introduce a novel constraint on the denoising geometry consistency for learning consistency-aware denoising patterns. Our evaluation for point cloud denoising and image denoising demonstrates that even without clean supervision, U-CAN also produces comparable denoising results with the state-of-the-art supervised methods.

6 Acknowledgement

This work was supported by Deep Earth Probe and Mineral Resources Exploration – National Science and Technology Major Project (2024ZD1003405), and the National Natural Science Foundation of China (62272263), and in part by Kuaishou. Junsheng Zhou is also partially funded by Baidu Scholarship.

References

[1] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T Silva. Point set surfaces. In *Proceedings Visualization, 2001. VIS'01.*, pages 21–29. IEEE, 2001. 2

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

[3] Haim Avron, Andrei Sharf, Chen Greif, and Daniel Cohen-Or. L1-sparse reconstruction of sharp point set surfaces. *ACM Transactions on Graphics (TOG)*, 29(5):1–12, 2010. 2

[4] Matthew Berger, Joshua A Levine, Luis Gustavo Nonato, Gabriel Taubin, and Claudio T Silva. A benchmark for surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(2):1–17, 2013. 2

[5] Frédéric Cazals and Marc Pouget. Estimating differential quantities using polynomial fitting of osculating jets. *Computer Aided Geometric Design*, 22(2):121–146, 2005. 2, 6, 7

[6] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007. 8

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

[8] Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or. Bilateral mesh denoising. In *ACM SIGGRAPH 2003 Papers*, pages 950–953. 2003. 6, 7

[9] Gaël Guennebaud and Markus Gross. Algebraic point set surfaces. In *ACM siggraph 2007 papers*, pages 23–es. 2007. 2

[10] Shihui Guo, Lishuang Zhan, Yancheng Cao, Chen Zheng, Guyue Zhou, and Jiangtao Gong. Touch-and-heal: Data-driven affective computing in tactile interaction with robotic dog. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 7(2):1–33, 2023. 1

[11] Liang Han, Junsheng Zhou, Yu-Shen Liu, and Zhizhong Han. Binocular-guided 3d gaussian splatting with view consistency for sparse view synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2

[12] 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*, pages 52–60, 2019. 1, 3, 4, 6, 7, 9

[13] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17853–17862, 2023. 1

[14] Hui Huang, Shihao Wu, Minglun Gong, Daniel Cohen-Or, Uri Ascher, and Hao Zhang. Edge-aware point set resampling. *ACM transactions on graphics (TOG)*, 32(1):1–12, 2013. 2

[15] Han Huang, Yulun Wu, Junsheng Zhou, Ge Gao, Ming Gu, and Yu-Shen Liu. Neusurf: On-surface priors for neural surface reconstruction from sparse input views. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. 2

[16] Jiahui Huang, Zan Gojcic, Matan Atzmon, Or Litany, Sanja Fidler, and Francis Williams. Neural kernel surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4369–4379, 2023. 2

[17] 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*, pages 14781–14790, 2021. 8

[18] Chuan Jin, Tieru Wu, Yu-Shen Liu, and Junsheng Zhou. Music-udf: Learning multi-scale dynamic grid representation for high-fidelity surface reconstruction from point clouds. *Computers & Graphics*, 124:104081, 2024. 2

[19] Chuan Jin, Tieru Wu, and Junsheng Zhou. Multi-grid representation with field regularization for self-supervised surface reconstruction from point clouds. *Computers & Graphics*, 2023. 2

[20] Sandip M Kasar and Sachin D Ruikar. Image demosaicking by nonlocal adaptive thresholding. In *2013 International Conference on Signal Processing, Image Processing & Pattern Recognition*, pages 34–38. IEEE, 2013. 8

[21] 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*, pages 2129–2137, 2019. 8

[22] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189*, 2018. 4

[23] Shujuan Li, Junsheng Zhou, Baorui Ma, Yu-Shen Liu, and Zhizhong Han. NeAF: Learning neural angle fields for point normal estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. 2

[24] Shujuan Li, Junsheng Zhou, Baorui Ma, Yu-Shen Liu, and Zhizhong Han. Learning continuous implicit field with local distance indicator for arbitrary-scale point cloud upsampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. 2

[25] Yaron Lipman, Daniel Cohen-Or, David Levin, and Hillel Tal-Ezer. Parameterization-free projection for geometry reconstruction. *ACM Transactions on Graphics (ToG)*, 26(3):22–es, 2007. 2

[26] Shitong Luo and Wei Hu. Differentiable manifold reconstruction for point cloud denoising. In *Proceedings of the 28th ACM international conference on multimedia*, pages 1330–1338, 2020. 1, 2, 3, 6, 7, 9

[27] Shitong Luo and Wei Hu. Score-based point cloud denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4583–4592, 2021. 1, 2, 3, 6, 7, 9

[28] 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. 3, 4

[29] Baorui Ma, Junsheng Zhou, Yu-Shen Liu, and Zhizhong Han. Towards better gradient consistency for neural signed distance functions via level set alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17724–17734, 2023. 2

[30] Youssef Mansour and Reinhard Heckel. Zero-shot noise2noise: Efficient image denoising without any data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14018–14027, 2023. 6, 8

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

[32] Enrico Mattei and Alexey Castrodad. Point cloud denoising via moving rpca. In *Computer Graphics Forum*, volume 36, pages 123–137. Wiley Online Library, 2017. 6, 7

[33] Gal Metzer, Rana Hanocka, Raja Giryes, and Daniel Cohen-Or. Self-sampling for neural point cloud consolidation. *ACM Transactions on Graphics (TOG)*, 40(5):1–14, 2021. 2

[34] Takeshi Noda, Chao Chen, Junsheng Zhou, Weiqi Zhang, Yu-Shen Liu, and Zhizhong Han. Learning bijective surface parameterization for inferring signed distance functions from sparse point clouds with grid deformation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22139–22149, 2025. 2

[35] A Cengiz Öztireli, Gael Guennebaud, and Markus Gross. Feature preserving point set surfaces based on non-linear kernel regression. In *Computer graphics forum*, volume 28, pages 493–501. Wiley Online Library, 2009. 2

[36] Francesca Pistilli, Giulia Fracastoro, Diego Valsesia, and Enrico Magli. Learning graph-convolutional representations for point cloud denoising. In *European conference on computer vision*, pages 103–118. Springer, 2020. 2, 3, 6, 7

[37] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 3

[38] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 3

[39] Yuhui Quan, Mingqin Chen, Tongyao Pang, and Hui Ji. Self2self with dropout: Learning self-supervised denoising from single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1890–1898, 2020. 8

[40] 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, pages 185–203. Wiley Online Library, 2020. 2, 3, 6, 7

[41] Yann Schoenenberger, Johan Paratte, and Pierre Vandergheynst. Graph-based denoising for time-varying point clouds. In *2015 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4. IEEE, 2015. 2

[42] Silvia Sellán and Alec Jacobson. Neural stochastic poisson surface reconstruction. In *SIGGRAPH Asia 2023 Conference Papers*, SA '23, New York, NY, USA, 2023. Association for Computing Machinery. 2

[43] 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. 8

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

[45] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018. 8

[46] 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. 2, 3, 4

[47] 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. 2

[48] Xin Wen, Junsheng Zhou, Yu-Shen Liu, Hua Su, Zhen Dong, and Zhizhong Han. 3D shape reconstruction from 2D images with disentangled attribute flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3803–3813, 2022. 2

[49] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10130–10139, 2019. 2

[50] Tong Wu, Liang Pan, Junzhe Zhang, Tai Wang, Ziwei Liu, and Dahua Lin. Density-aware chamfer distance as a comprehensive metric for point cloud completion. *arXiv preprint arXiv:2111.12702*, 2021. 9

[51] Wang Yifan, Shihao Wu, Hui Huang, Daniel Cohen-Or, and Olga Sorkine-Hornung. Patch-based progressive 3d point set upsampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5958–5967, 2019. 2

[52] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Ec-net: an edge-aware point set consolidation network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 386–402, 2018. 2

[53] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2790–2799, 2018. 6, 9

[54] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19313–19322, 2022. 2

[55] Jin Zeng, Gene Cheung, Michael Ng, Jiahao Pang, and Cheng Yang. 3d point cloud denoising using graph laplacian regularization of a low dimensional manifold model. *IEEE Transactions on Image Processing*, 29:3474–3489, 2019. 2, 6, 7

[56] Lishuang Zhan, Tianyang Xiong, Hongwei Zhang, Shihui Guo, Xiaowei Chen, Jiangtao Gong, Juncong Lin, and Yipeng Qin. Toucheditor: interaction design and evaluation of a flexible touchpad for text editing of head-mounted displays in speech-unfriendly environments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 7(4):1–29, 2024. 1

[57] Lishuang Zhan, Enting Ying, Jiabao Gan, Shihui Guo, BoYu Gao, and Yipeng Qin. Satpose: Improving monocular 3d pose estimation with spatial-aware ground tactility. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6192–6201, 2024. 2

[58] Chengwei Zhang, Xueyi Zhang, Xianghu Yue, Mingrui Lao, Tao Jiang, Jiawei Wang, Fubo Zhang, and Longyong Chen. 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*, pages 3381–3390, 2024. 2

[59] 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. 3, 7

[60] Wenyuan Zhang, Emily Yue-ting Jia, Junsheng Zhou, Baorui Ma, Kanle Shi, Yu-Shen Liu, and Zhizhong Han. Nerfprior: Learning neural radiance field as a prior for indoor scene reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11317–11327, 2025. 2

[61] Wenyuan Zhang, Yu-Shen Liu, and Zhizhong Han. Neural signed distance function inference through splatting 3d gaussians pulled on zero-level set. *Advances in Neural Information Processing Systems*, 37:101856–101879, 2024. 2

[62] Wenyuan Zhang, Kanle Shi, Yu-Shen Liu, and Zhizhong Han. Learning unsigned distance functions from multi-view images with volume rendering priors. In *European Conference on Computer Vision*, pages 397–415. Springer, 2024. 2

[63] Wenyuan Zhang, Jimin Tang, Weiqi Zhang, Yi Fang, Yu-Shen Liu, and Zhizhong Han. MaterialRefGS: Reflective gaussian splatting with multi-view consistent material inference. In *Advances in Neural Information Processing Systems*, 2025. 2

[64] Wenyuan Zhang, Ruofan Xing, Yunfan Zeng, Yu-Shen Liu, Kanle Shi, and Zhizhong Han. Fast learning radiance fields by shooting much fewer rays. *IEEE Transactions on Image Processing*, 2023. 2

[65] Wenyuan Zhang, Yixiao Yang, Han Huang, Liang Han, Kanle Shi, Yu-Shen Liu, and Zhizhong Han. Monoinstance: Enhancing monocular priors via multi-view instance alignment for neural rendering and reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21642–21653, 2025. 2

[66] Weiqi Zhang, Junsheng Zhou, Haotian Geng, Wenyuan Zhang, and Yu-Shen Liu. Gap: Gaussianize any point clouds with text guidance. *arXiv preprint arXiv:2508.05631*, 2025. 2

[67] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 2

[68] Junsheng Zhou, Yu-Shen Liu, and Zhizhong Han. Zero-shot scene reconstruction from single images with deep prior assembly. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2

[69] Junsheng Zhou, Baorui Ma, Shujuan Li, Yu-Shen Liu, Yi Fang, and Zhizhong Han. Cap-udf: Learning unsigned distance functions progressively from raw point clouds with consistency-aware field optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2

[70] Junsheng Zhou, Baorui Ma, Shujuan Li, Yu-Shen Liu, and Zhizhong Han. Learning a more continuous zero level set in unsigned distance fields through level set projection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023. 2

[71] Junsheng Zhou, Baorui Ma, and Yu-Shen Liu. Fast learning of signed distance functions from noisy point clouds via noise to noise mapping. *IEEE transactions on pattern analysis and machine intelligence*, 2024. 2

[72] Junsheng Zhou, Baorui Ma, Yu-Shen Liu, Yi Fang, and Zhizhong Han. Learning consistency-aware unsigned distance functions progressively from raw point clouds. In *Advances in Neural*

Information Processing Systems (NeurIPS), 2022. 2

[73] Junsheng Zhou, Baorui Ma, Wenyuan Zhang, Yi Fang, Yu-Shen Liu, and Zhizhong Han. Differentiable registration of images and lidar point clouds with voxelpoint-to-pixel matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2

[74] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. *International Conference on Learning Representations*, 2024. 2

[75] Junsheng Zhou, Xin Wen, Baorui Ma, Yu-Shen Liu, Yue Gao, Yi Fang, and Zhizhong Han. 3d-oae: Occlusion auto-encoders for self-supervised learning on point clouds. *IEEE International Conference on Robotics and Automation (ICRA)*, 2024. 2

[76] Junsheng Zhou, Weiqi Zhang, and Yu-Shen Liu. Diffgs: Functional gaussian splatting diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2

[77] Junsheng Zhou, Weiqi Zhang, Baorui Ma, Kanle Shi, Yu-Shen Liu, and Zhizhong Han. Udiff: Generating conditional unsigned distance fields with optimal wavelet diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We analysis the limitations of our method in the appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the detailed information in reproducing our methods in Sec.3, Sec.4 of the main paper and the appendix. We also provide a demonstration code of our method in the supplementary materials.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide our demonstration code as a part of our supplementary materials. We will release the source code, data and instructions upon acceptance.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the training and testing details in the experiment section (Sec.4).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We report the average performance as the experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computer resources needed to reproduce the experiments are provided in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the applications and potential impacts of our method in the introduction.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We use the datasets under their licenses.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [No]

Justification: This paper does not use LLMs as important, original, or non-standard component of the core methods.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.