NoiseSDF2NoiseSDF: Learning Clean Neural Fields from Noisy Supervision

Anonymous Author(s)
Affiliation
Address
email

Abstract

Reconstructing accurate implicit surface representations from point clouds remains a challenging task, particularly when data is captured using low-quality scanning devices. These point clouds often contain substantial noise, leading to inaccurate surface reconstructions. Inspired by the Noise2Noise paradigm for 2D images, we introduce NoiseSDF2NoiseSDF, a novel method designed to extend this concept to 3D neural fields. Our approach enables learning clean neural SDFs directly from noisy point clouds through noisy supervision by minimizing the MSE loss between noisy SDF representations, allowing the network to implicitly denoise and refine surface estimations. We evaluate the effectiveness of NoiseSDF2NoiseSDF on benchmarks, including the ShapeNet, ABC, Famous, and Real datasets. Experimental results demonstrate that our framework significantly improves surface reconstruction quality from noisy inputs.

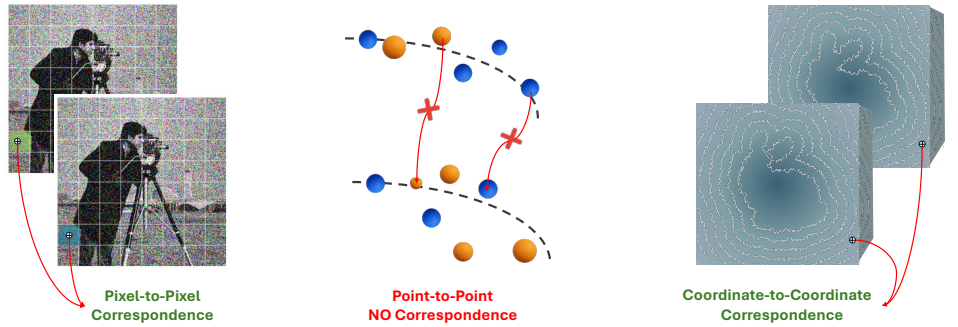
1 Introduction

Learning from imperfect targets [48, 18, 24, 5, 50] is a fundamental challenge in machine learning, particularly when obtaining clean training labels is impractical or unfeasible. In image processing, the pioneering work of Noise2Noise [24] demonstrated that image restoration could effectively be achieved by observing multiple corrupted instances of the same scene. Specifically, this method leverages the principle that pixel values at identical coordinates in different noisy images ideally represent the same underlying true signal. Consequently, the model learns to restore clean images by simply minimizing a straightforward MSE loss between noisy observations. See Figure 1 (a).

Extending Noise2Noise principles to 3D point clouds [19, 30], however, poses significant challenges due to their inherently unstructured nature. Unlike images organized on regular grids, point clouds exhibit deviations across all spatial coordinates without the benefit of a stable reference framework. This fundamental difference renders a direct extension of unsupervised image denoisers impractical. Standard loss functions such as Mean Squared Error (MSE) prove ineffective, necessitating specialized loss functions like Earth Mover's Distance (EMD) to capture geometric correspondences and spatial distributions inherent in point cloud data, see Figure 1 (b).

Recent advances in surface reconstruction have introduced neural fields, such as neural Signed Distance Function (neural SDF) [34, 31], which are capable of predicting continuous SDF values for any given 3D coordinate. Our key observation is that neural SDF, which encodes the SDF mapping 3D coordinates to scalar distance values for 3D shape, exhibits a conceptual parallel to the mapping between pixel coordinates and pixel intensities in 2D images, as shown in Figure 1 (c). Building on this analogy, we hypothesize that neural SDFs can be denoised by directly using noisy SDF observations with the same MSE loss strategy inspired by the Noise2Noise principle in image restoration.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.



(a) Noisy Images for Same Scene (b) Noisy Point Clouds for Same Shape (c) Noisy Neural SDFs for Same Shape

Figure 1: Comparison of coordinate correspondences: (a) Pixel coordinates represent correspondences between two noisy images of the same scene. In contrast, (b) point coordinates do not exhibit correspondences between two noisy point clouds of the same shape. (c) SDF coordinates establish correspondences between two noisy neural fields representing the same shape.

In this work, we explore the use of noisy-target supervision in neural SDFs for surface reconstruction from noisy 3D point clouds. We propose NoiseSDF2NoiseSDF, an adaptation of the Noise2Noise framework applied to neural fields. The workflow of our NoiseSDF2NoiseSDF is illustrated in Figure 2. The network first takes independently corrupted point clouds as input to predict the underlying clean SDF values. Instead of using clean SDFs as ground truth, we employ another noisy neural SDF, which is generated by off-the-shelf point-to-SDF methods, as the supervision target. We then minimize the discrepancy between the predicted SDF output and the noisy SDF target using MSE loss. The network learns to suppress noise and improve consistency across SDF values, leading to clean neural representations.

To evaluate the effectiveness of NoiseSDF2NoiseSDF, we conduct comprehensive experiments across benchmark datasets, including ShapeNet [8], ABC [22], Famous [14], and Real [14]. Our experimental results demonstrate that neural SDFs can indeed be denoised effectively by employing mean squared error loss directly between their noisy representations. This finding confirms our central hypothesis: neural SDFs can learn to produce cleaner outputs simply by observing and minimizing discrepancies among noisy neural fields, effectively extending the Noise2Noise paradigm into the domain of 3D neural surface reconstruction. Our approach eliminates the need for clean training data, making it practical and scalable for real-world scenarios where acquiring perfect data is difficult or infeasible.

2 Related Work

Noise2Noise. The Noise2Noise (N2N) framework [24] has significantly influenced recent image denoising. By leveraging pairs of noisy observations of the same scene, the N2N framework learns to predict one noisy realization from another via pixel-wise correspondence. Subsequent methods like Noise2Void [23], Noise2Self [3] employ blind-spot masking techniques, training models directly on individual noisy images without pairs. Noise2Same [43] derives self-supervised loss bounds to eliminate the blind-spot restriction altogether. Self2Self [38] and Neighbor2Neighbor [20] exploit internal image redundancy, employing dropout or pixel resampling to train directly on single noisy observations without explicit noise modeling. Noisier2Noise [33] extends N2N to explicitly introduce additional synthetic noise, learning to map noisier images back to their original noisy versions.

Extending the Noise2Noise framework to 3D [19, 42, 30, 40] is challenging due to the unordered nature of point clouds. Methods like TotalDenoising [19] and Noise2Noise Mapping [30] address this by leveraging local geometric correspondences instead of exact point matches, replacing MSE with Earth Mover's Distance (EMD) loss to better align noisy point clouds with the underlying surface. In our work, we exploit the structural similarities between neural fields and images, proposing a Noise2Noise denoising framework for 3D SDFs with MSE loss.

Implicit Surface Reconstruction. Learning implicit surfaces from point clouds has seen significant advances. Overfitting-based methods optimize a neural implicit function for a single point cloud,

often with ground-truth SDFs, normals or geometric constraints and physical priors. For example, SAL [1], SALD [2], and Sign-SAL [49] use point proximity and self-similarity cues. Gradient regularization techniques like IGR [17], DiGS [4], and Neural-Pull [29] improve stability and detail. Extensions such as SAP [36], LPI [9], and Implicit Filtering-Net [25] enhance reconstruction under sparse sampling and complex geometry. While accurate, these methods are typically sensitive to noise. Robust variants (e.g., SAP [36], PGR [26], Neural-IMLS [41], Noise2Noise Mapping [30] and LocalN2NM [10]) address this via smoothing, denoising priors, or self-supervision.

In contrast to overfitting approaches, data-driven methods learn shape priors from large datasets. Global-latent methods, such as OCCNet [31], IM-NET [11], and DeepSDF [34], encode entire shapes into fixed-length latent codes, capturing overall semantics but often over-smoothing details. Local prior methods improve expressiveness by operating at finer scales. Grid-based approaches divide space into cells and learn small implicit functions per cell (ConvOccNet [35], SSRNet [32], Local Implicit Grid [16], Deep Local Shapes [7]). Patch-based methods segment point clouds into local regions and learn shared atomic representations (PatchNets [39], POCO [6], neighborhood-based [21]). Hybrid methods combine global context with local detail. For instance, IF-Nets [12] and SG-NN [13] fuse PointNet features with voxel hierarchies or contrastive scene priors. Recent transformer-based models (ShapeFormer [44], 3DILG [46], 3DS2V [47], LaGeM [45]) leverage self-attention for long-range structure modeling. Data-driven models trained with ground-truth SDFs are generally more robust to noise. Hybrid approaches like P2S [37] and PPSurf [15] use dual-branch networks to predict SDFs with explicit noise-level supervision. However, their performance degrades under extreme sparsity and noise of input point clouds. In contrast, our NoiseSDF2NoiseSDF framework embraces noise as a training signal, enabling reliable surface recovery from severely degraded inputs.

3 Preliminaries

In *Noise2Noise* [24], the key idea is that given multiple noisy observations of the same underlying clean image, the pixel intensities at the same spatial coordinates are expected to share the same statistical properties. Formally, consider an image domain $\mathbf{X} \subset \mathbb{R}^2$, and let $y_1, y_2, \ldots, y_n$ be noisy observations of the same underlying clean image taken at different instances. For any pixel coordinate $x \in \mathbf{X}$, the pixel intensities $y_1(x), y_2(x), \ldots, y_n(x)$ are samples drawn from a distribution centered around the true pixel value at that location, perturbed by independent, zero-mean noise. The core insight of Noise2Noise is that even in the presence of such noise, the expectation of the noisy pixel values converges to the true signal:

$$\mathbb{E}[y_i(x)] = y(x), \quad \forall i \in \{1, 2, \ldots, n\}, \tag{1}$$

where $y_i(x)$ is the observed pixel value at coordinate x in the i-th noisy image, and $y(x)$ is the true underlying pixel value at that coordinate. This property enables training a neural network purely on noisy data, using other noisy images as supervision.

Let f_θ denote a neural network parameterized by θ, and let $x \in \mathbf{X}$ represent a spatial query coordinate. The network is designed to predict pixel intensities given a noisy image and the query coordinate. The prediction is written as:

$$\hat{y}(x \mid y_i) = f_\theta(y_i, x), \tag{2}$$

where y_i is the noisy input image, x is the queried pixel location, and $\hat{y}(x \mid y_i)$ is the predicted pixel intensity at x.

The model is trained to minimize the expected squared error between the predicted pixel value and the corresponding pixel value in another independent noisy observation. The loss function is:

$$\mathcal{L}(\theta) = \mathbb{E}_{y_1, y_2 \sim p(y|y), x \sim \mathcal{U}(\mathbb{R}^2)} \left[\|\hat{y}(x \mid y_1) - y_2(x)\|^2 \right], \tag{3}$$

where y_1, y_2 are independent noisy observations of the same clean image, and $x \in \mathbf{X}$ is sampled uniformly from the image domain.

4 Method

Our proposed method investigates whether clean neural fields can be effectively learned by observing their noisy counterparts. Drawing inspiration from Noise2Noise, where noisy images directly serve as inputs and targets, we adapt this principle to learning neural fields from noisy point cloud data. In contrast to the direct usage of noisy images as input in traditional Noise2Noise setups, we employ a neural network conditioned on a noisy point cloud to predict neural SDFs at specific query coordinates. Rather than utilizing clean SDFs as supervision, our approach leverages noisy neural fields at identical coordinates derived from another independently noisy version of the same underlying shape. This ensures one-to-one correspondence between the predicted and target neural fields, allowing effective noise suppression through direct MSE loss minimization.

4.1 NoiseSDF2NoiseSDF

Applying *Noise2Noise* [24] to Signed Distance Functions (SDFs) introduces new opportunities for denoising in 3D spaces. Unlike unstructured point clouds, SDFs represent 3D geometry in a structured and continuous manner, mapping each spatial coordinate $q \in \mathbb{R}^3$ to its signed distance from the surface of an underlying object. This continuity ensures that, for the same query coordinate across multiple noisy observations derived from the same shape, the SDF values should remain statistically consistent. Let $p_1, p_2, \ldots, p_n$ be noisy point cloud observations of the same underlying 3D shape, and let $s_1, s_2, \ldots, s_n$ be their corresponding noisy SDFs. Given a noisy point cloud p_i and a query coordinate q, a neural network f_θ, parameterized by θ, is trained to predict the SDF value $\hat{s}(q \mid p_i)$ at the queried location:

$$\hat{s}(q \mid p_i) = f_\theta(p_i, q). \tag{4}$$

The structured nature of SDFs enables the network to learn smooth and continuous surface representations, even from sparse or noisy inputs. This makes SDFs advantageous over unordered point clouds for tasks like 3D denoising and reconstruction.

Training Objective. The model is trained by minimizing the expected squared error between the predicted SDF value from one noisy observation and the SDF value at the same query location in another noisy observation of the same shape. The loss function is defined as:

$$\mathcal{L}(\theta) = \mathbb{E}_{p_1, p_2 \sim p(p|s),\, q \sim \mathcal{U}(\mathbb{R}^3)} \left[\left\| \hat{s}(q \mid p_1) - s_2(q \mid p_2) \right\|^2 \right], \tag{5}$$

where p_1, p_2 are independent noisy point cloud observations sampled from the same underlying shape s, and $s_2(q \mid p_2)$ is the noisy SDF value at coordinate q associated with noisy point cloud p_2.

This formulation takes advantage of the continuous nature of SDFs, which, unlike point clouds, allows for consistent supervision across noisy samples even if the raw point distributions are unstructured. By learning to map noisy coordinates to structured SDF representations, the neural network effectively filters noise, yielding a refined and more accurate 3D representation of the surface.

4.2 Implementation

Our framework is illustrated in Figure 2. The process begins with sampling sparse, noisy point clouds from a watertight surface. During training, a pair of noisy point clouds is randomly selected: one is processed through the neural SDF network to predict approximate clean SDF values for the underlying 3D shape. Simultaneously, a point-to-SDF method is applied to generate a noisy SDF target, which serves as noisy supervision during the denoising phase. For each query point, the corresponding SDF values from these two representations are extracted and compared using the Mean Squared Error (MSE) loss function. This loss is then utilized to update the weights of the neural SDF network during denoising.

Point Sampling. We first normalize the watertight meshes into a unit cube, then sample points from the surfaces to obtain the original point cloud p. Following the Noise2Noise Mapping protocol [24, 30], we apply zero-mean Gaussian noise to generate noisy point cloud pairs. The query point set consists of 50% near-surface points and 50% uniformly sampled points from the unit cube. To reduce

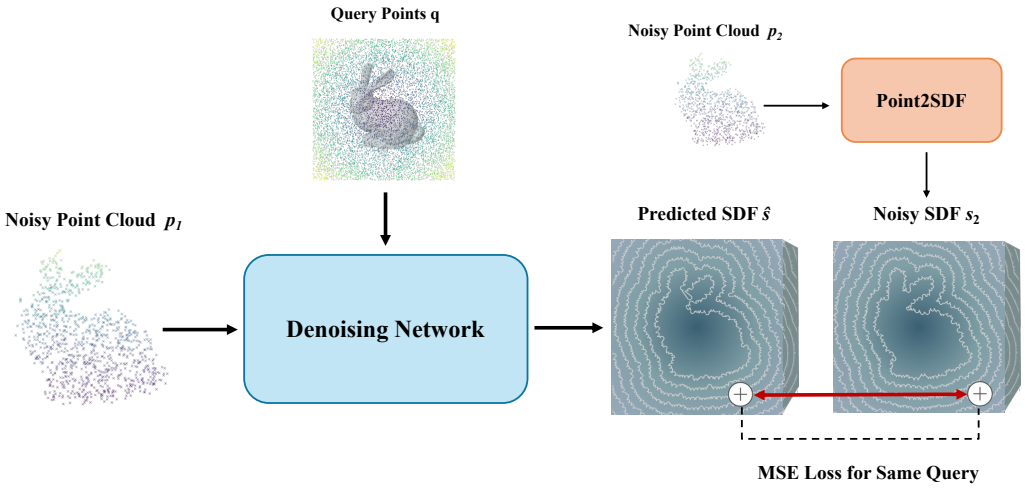


Figure 2: The training pipeline of the NoiseSDF2NoiseSDF framework. Given two independent noisy point clouds p_1 and p_2 of the same underlying shape, p_1 is fed into the denoising network to predict a smoothed SDF $\hat{s}$, while p_2 is passed through a Point2SDF network to generate a noisy SDF s_2. Both SDFs are evaluated at a shared set of query points q, and their mean squared error is used to update the denoising network weights.

dependency on the original clean surface, we directly use the two input noisy point clouds as the near-surface query points. Additionally, we uniformly sample N points within the cube as spatial query points.

Denoising Network. Our SDF prediction network is built on 3DS2V [47]. Initially, a noisy point cloud p_1 is sampled and transformed into positional embeddings, which are then encoded into a set of latent codes through a cross-attention module. Subsequently, self-attention is applied to aggregate and exchange information across the latent set, enhancing feature integration. A cross-attention module then computes interpolation weights for the query point q. These interpolated feature vectors are processed through a fully connected layer to predict SDF values. The network is initialized using pretrained weights from 3DS2V. Without the denoising learning, the predicted SDF values are typically noisy due to the inherent noise in the input point clouds. During our denoising phase, the encoder remains fixed while only the decoder is optimized, improving both efficiency and convergence speed. Importantly, our contribution is not limited to the specific architecture of the SDF prediction network; other point-to-SDF methods could also be utilized.

Noisy Target. Given another paired noisy point cloud p_2, a Point2SDF method is required to predict noisy SDF values s_2 from it. In our implementation, we also employ a pretrained 3DS2V [47] model. This network accepts as input a noisy point cloud p_2 and a query point q, producing the corresponding noisy SDF scalar value at q. To ensure that all SDF targets are consistently noisy, we freeze its parameters during this process. Notably, this Point2SDF method is also not restricted to a specific model; any data-driven or overfitting Point2SDF methods could be seamlessly integrated. The Mean Squared Error (MSE) loss function is employed to minimize discrepancies between corresponding SDF values. This loss guides the optimization of the neural SDF network's weights during the denoising training.

5 Experiment

We designed experiments to evaluate the performance of the NoiseSDF2NoiseSDF framework for surface reconstruction from raw noisy point clouds, assessing performance across different noise levels using a data-driven paradigm on large 3D shape datasets. We conducted ablation studies to validate key components and design choices.

5.1 Training Details

For optimization, we used the AdamW optimizer [28] with a fixed learning rate of 1×10^{-4}. For resource usage, we trained on three Nvidia A100 GPUs with a batch size of 32 per GPU, taking approximately 15 hours for the ShapeNet dataset and 2.5 hours for the ABC dataset.

We sampled 2048 points from watertight meshes as the initial point cloud. Following the Noise2Noise Mapping [30], we applied Gaussian noise with standard deviations of 1%, 2%, online to generate noisy and sparse point cloud pairs. Additionally, we sampled 8192 query points online. The noise magnitude is defined with respect to both the point-cloud bounding-box size and the point density. For a fixed numeric noise level, a smaller bounding box amplifies the relative impact of the perturbation. All point clouds are normalized to the cubes $[-0.5, 0.5]^3$ or $[-1, 1]^3$. Furthermore, sparser point sets are more susceptible to noise. With only 2048 points, noise levels of 0.01 and 0.02 constitute *severe* perturbations irrespective of the bounding-box scale. The clean underlying surface is recovered from the denoised SDF using the Marching Cubes [27].

5.2 Datasets and Metrics

We trained our NoiseSDF2NoiseSDF network on the ShapeNet dataset following [47], using the same data split and preprocessing procedures. To evaluate denoising effectiveness and surface reconstruction quality, we used several metrics, including Intersection-over-Union (IoU), Chamfer Distance, F1 Score, and Normal Consistency.

IoU was computed based on occupancy predictions over densely sampled volumetric points. Following methods [25, 29], we sampled 1×10^5 points from the reconstructed and ground-truth surfaces to compute the Chamfer Distance and F1 Score.

To further assess the generalization capability of NoiseSDF2NoiseSDF, we further trained our model on the ABC train set [22]—which was not used during the pretraining of 3DS2V—and then evaluated it on the ABC test set, as well as the Famous [14] and Real [14] datasets. We utilized the preprocessed datasets and data splits provided by [14, 15]. We reported evaluation metrics including Normal Consistency, Mesh Normal Consistency, Chamfer Distance, and F1 Score. All metrics reported above are evaluated on the reconstructed meshes. We excluded IoU from this benchmark because, under severe noise, many reconstructed meshes become non-watertight or heavily degenerated, making it infeasible to assign reliable inside/outside labels and rendering the IoU metric unreliable.

5.3 Results on ShapeNet

We compared our method against 3DS2V [47] and 3DILG [46] on the seven largest ShapeNet subsets, using the same training and testing data split. We used the official pretrained models released by the respective authors. Evaluation results are reported for each subset at noise levels of 0.01 (Table 1) and 0.02 (Table 2); see visualization results in Figure 3. The results demonstrate that our method maintains robustness under both mild $\sigma = 0.01$ and severe $\sigma = 0.02$ corruption levels. Under lower corruption ($\sigma = 0.01$), our method outperforms all competing methods across all evaluation metrics. For instance, IoU rises by 4% for chair and by about 9% for rifle. Chamfer Distance drops from 0.010 to 0.008 for airplane and from 0.011 to 0.009 for lamp. Normal Consistency and F-Score likewise see significant improvements. At the higher corruption level ($\sigma = 0.02$), while the performance of all methods degrades, our approach remains the most robust and stable with the best mean metrics surpassing all baseline models. In challenging categories such as table, rifle, and lamp, our model still leads: rifle achieves an IoU of 0.781 and a Chamfer Distance of 0.014.

5.4 Results on ABC, Famous, and Real

We compared results on the ABC, Famous, and Real test datasets offered by P2S [14]. We compared three data-driven methods P2S [14], PPSurf [15], and 3DS2V [47] and two overfitting methods SAP-O [36], PGR [26]. These methods are widely recognized for their strong resilience to noise in point cloud data. For the data-driven methods, we used the pretrained models provided by the original authors. For the overfitting baselines we adopted the training configurations recommended or set as default in their respective works. Quantitative results are reported in Table 3 and Table 4, and qualitative mesh reconstructions are visualized in Figure 4.

Table 1: Performance comparison of 3DS2V [47], 3DILG [46], and Ours on ShapeNet test datasets derived from the 3DS2V with an additional Gaussian noise $\sigma = 0.01$. Higher is better for IoU, NC, and F-Score; lower is better for Chamfer. Best results are highlighted in bold.

Category	IoU ↑			NC ↑			Chamfer ↓			F-Score ↑		
	3DS2V	3DILG	Ours	3DS2V	3DILG	Ours	3DS2V	3DILG	Ours	3DS2V	3DILG	Ours
table	0.879	0.856	**0.922**	0.930	0.932	**0.976**	0.013	0.014	**0.012**	0.991	0.982	**0.992**
car	0.946	0.931	**0.959**	0.890	0.861	**0.908**	0.022	0.025	**0.020**	0.925	0.899	**0.925**
chair	0.887	0.881	**0.921**	0.937	0.930	**0.966**	0.014	0.015	**0.013**	0.986	0.977	**0.986**
airplane	0.884	0.871	**0.931**	0.939	0.924	**0.972**	0.010	0.010	**0.008**	0.997	0.990	**0.997**
sofa	0.946	0.942	**0.964**	0.943	0.934	**0.974**	0.014	0.015	**0.012**	0.986	0.980	**0.987**
rifle	0.821	0.839	**0.910**	0.869	0.875	**0.960**	0.009	0.010	**0.007**	0.997	0.991	**0.998**
lamp	0.826	0.825	**0.894**	0.904	0.883	**0.952**	0.011	0.016	**0.009**	0.989	0.956	**0.989**
mean	0.884	0.878	**0.929**	0.916	0.905	**0.958**	0.0132	0.015	**0.0113**	0.981	0.968	**0.986**

Table 2: Performance comparison of 3DS2V [47], 3DILG [46], and Ours under an additional Gaussian noise $\sigma = 0.02$. Best results are highlighted in bold.

Category	IoU ↑			NC ↑			Chamfer ↓			F-Score ↑		
	3DS2V	3DILG	Ours	3DS2V	3DILG	Ours	3DS2V	3DILG	Ours	3DS2V	3DILG	Ours
table	0.528	0.430	**0.591**	0.765	0.756	**0.912**	0.029	0.036	**0.028**	0.792	0.723	**0.859**
car	0.434	**0.541**	0.491	0.715	0.699	**0.787**	**0.040**	0.052	0.044	0.669	0.600	**0.688**
chair	0.463	0.392	**0.530**	0.729	0.712	**0.868**	**0.034**	0.037	0.035	0.694	0.701	**0.721**
airplane	0.465	**0.564**	0.536	0.719	0.710	**0.856**	0.025	0.031	**0.022**	0.830	0.764	**0.899**
sofa	0.355	0.312	**0.425**	0.769	0.737	**0.866**	**0.036**	0.039	0.038	0.667	0.664	**0.677**
rifle	0.625	0.564	**0.781**	0.691	0.691	**0.891**	0.021	0.029	**0.014**	0.887	0.776	**0.968**
lamp	0.572	0.461	**0.649**	0.744	0.712	**0.896**	0.026	0.040	**0.025**	0.825	0.699	**0.880**
mean	0.492	0.466	**0.572**	0.733	0.717	**0.868**	0.030	0.038	**0.029**	0.766	0.704	**0.813**

For data-driven comparison, across all noise levels our method achieves the highest NC and Mesh NC, indicating the most coherent geometry and the smoothest surfaces; this is also evident in the visual reconstructions (Figure 4). At the 0.01 noise level, the ABC and Famous datasets reach NC scores of 0.865 and 0.831, respectively—an improvement of roughly 1.5–2.5 percentage points over the second-best approach. When the noise level increases to 0.02, all baselines degrade, yet our NC remains the best among them. *For methods that fit a surface to each test cloud individually*, Table 4 shows that SAP-O and PGR can sometimes obtain lower Chamfer distance and higher F1, but our approach still leads on NC and Mesh NC. Under 0.01 noise level, our average NC/ Mesh NC / F-Score reach 0.847/0.027/0.945, all of which are the top scores. However, when the noise level rises to 0.02, our Chamfer and F1 decrease substantially. This drop is partly due to the backbone 3DS2V model's sensitivity to heavy noise: its F1 plunges from 0.962 to 0.813 as the noise level increases from 0.01 to 0.02. As illustrated in Figure 4, the mesh geometry becomes severely corrupted and large holes appear. Although NoiseSDF2NoiseSDF averages the neural SDF and closes some gaps, it cannot perfectly recover the underlying surface. Moreover, the smoothing effect of our model at the 0.02 noise level removes certain fine details; while this yields visually smoother meshes, Chamfer distance and F1—metrics that emphasize global geometric fidelity rather than surface roughness—are consequently worse.

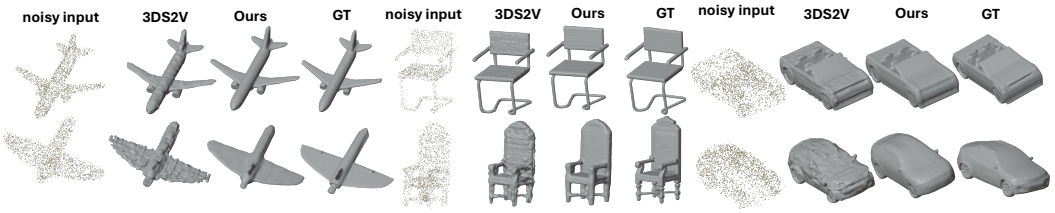


Figure 3: Comparison on the ShapeNet dataset. The first row corresponds to Gaussian noise with standard deviation $\sigma = 0.01$, and the second row to $\sigma = 0.02$. Compared to the baseline 3DS2V[47] method, our approach produces smoother reconstructions that more closely align with the underlying surfaces.

Table 3: Comparison of P2S [14], PPSurf [15], 3DS2V [47], and Ours on six noisy test datasets. Higher is better for NC, F-Score; lower is better for Mesh NC and Chamfer distance.

Dataset	NC ↑				Mesh NC ↓				Chamfer				F-Score			
	P2S	PPSurf	3DS2V	Ours	P2S	PPSurf	3DS2V	Ours	P2S	PPSurf	3DS2V	Ours	P2S	PPSurf	3DS2V	Ours
ABC ($\sigma = 0.01$)	0.790	0.770	0.859	**0.865**	0.330	0.059	0.036	**0.024**	0.017	0.017	**0.014**	0.015	0.919	0.935	**0.959**	0.938
ABC ($\sigma = 0.02$)	0.753	0.728	0.735	**0.812**	0.381	0.061	0.060	**0.018**	0.027	**0.022**	0.028	0.032	0.852	**0.870**	0.780	0.724
Famous ($\sigma = 0.01$)	0.771	0.761	0.819	**0.831**	0.268	0.053	0.040	**0.025**	0.017	**0.015**	0.014	0.016	0.928	**0.959**	0.962	0.941
Famous ($\sigma = 0.02$)	0.727	0.728	0.705	**0.767**	0.328	0.054	0.064	**0.024**	0.022	**0.020**	0.028	0.032	0.868	**0.899**	0.788	0.726
Real ($\sigma = 0.01$)	0.789	0.776	0.818	**0.845**	0.177	0.057	0.056	**0.031**	0.016	0.016	**0.014**	0.015	0.946	0.954	**0.964**	0.956
Real ($\sigma = 0.02$)	0.734	0.745	0.735	**0.793**	0.269	0.053	0.071	**0.020**	**0.021**	0.022	0.021	0.026	**0.877**	0.876	0.873	0.809
mean ($\sigma = 0.01$)	0.783	0.769	0.832	**0.847**	0.258	0.056	0.044	**0.027**	0.017	0.016	**0.014**	0.015	0.931	0.949	**0.962**	0.945
mean (all)	0.761	0.751	0.778	**0.819**	0.292	0.056	0.055	**0.024**	0.020	**0.019**	0.020	0.023	0.898	**0.916**	0.888	0.849

Table 4: Comparison of SAP-O [36], PGR [26], and Ours on six noisy test datasets. The released PGR implementation uses an adaptive Marching Cubes resolution that, for 2,048-point-point clouds, occasionally drops to 64 rather than 128, which can yield artificially smoother meshes.

Dataset	NC ↑			Mesh NC ↓			Chamfer			F-Score		
	SAP-O	PGR	Ours	SAP-O	PGR	Ours	SAP-O	PGR	Ours	SAP-O	PGR	Ours
ABC ($\sigma = 0.01$)	0.710	0.835	**0.865**	0.079	0.037	**0.024**	0.021	0.020	**0.014**	0.906	0.896	**0.938**
ABC ($\sigma = 0.02$)	0.622	0.778	**0.812**	0.095	0.065	**0.018**	0.026	**0.026**	0.032	**0.824**	0.815	0.724
Famous ($\sigma = 0.01$)	0.745	0.813	**0.831**	0.053	0.035	**0.025**	0.022	0.017	**0.016**	0.876	0.931	**0.941**
Famous ($\sigma = 0.02$)	0.614	0.755	**0.767**	0.104	0.064	**0.024**	**0.023**	0.024	0.032	0.849	0.834	**0.726**
Real ($\sigma = 0.01$)	0.683	0.827	**0.845**	0.097	0.032	**0.031**	0.023	0.015	**0.015**	0.902	0.956	**0.956**
Real ($\sigma = 0.02$)	0.595	0.756	**0.793**	0.122	0.062	**0.020**	**0.025**	0.026	0.026	**0.841**	0.824	0.809
mean ($\sigma = 0.01$)	0.713	0.825	**0.847**	0.076	0.035	**0.027**	0.022	0.017	**0.015**	0.895	0.928	**0.945**
mean (all)	0.661	0.794	**0.819**	0.092	0.049	**0.024**	0.023	**0.021**	0.023	0.866	**0.876**	0.849

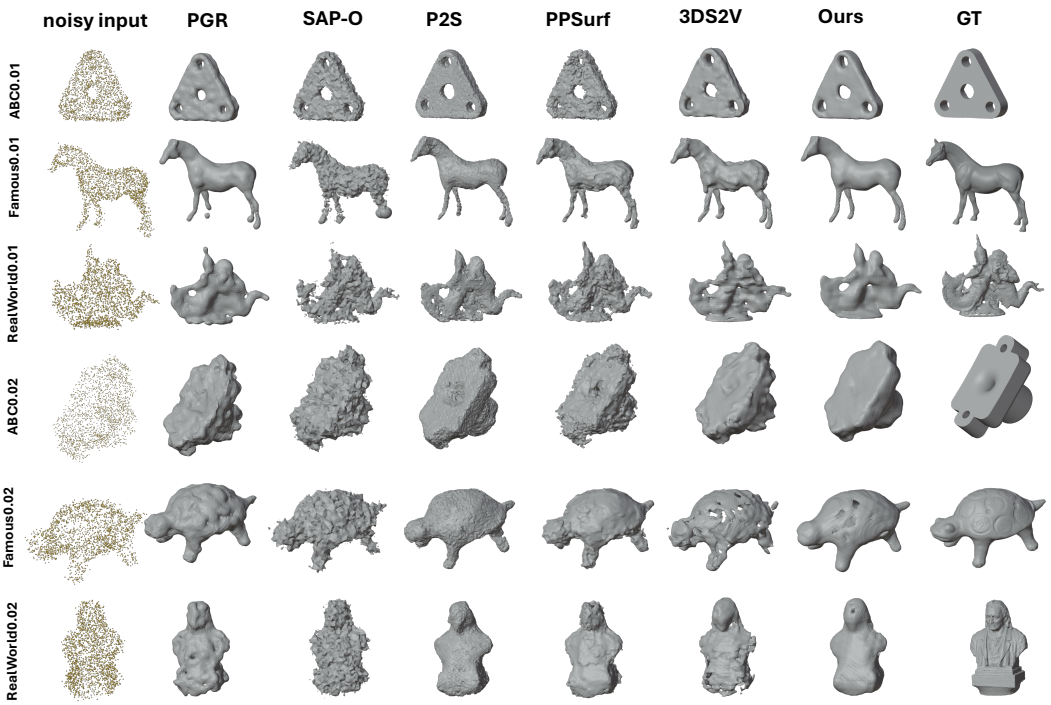


Figure 4: Visualization of surface reconstruction under Gaussian noise ($\sigma = 0.01$ and 0.02), comparing two overfitting-based methods (PGR [26] and SAP-O [36]), three data-driven approaches (P2S [14], PPSurf [15], 3DS2V [47]), and our proposed method. Results are shown for three benchmark datasets: ABC [22], Famous [14], and Real [14].

5.5 Ablation Study

Our ablation experiments were carried out on the "Chair" subset of ShapeNet, which contains 6271 models for training, 169 for validation, and 338 for testing. The configurations for all of our principal experiments originate with this subset and are progressively generalized to broader settings.

Denoising Network. The 3DS2V decoder contains 24 self-attention blocks, one cross-attention block, and a fully connected layer. Fine-tuning the entire decoder yields the best denoising performance, with metrics steadily improving over 15 epochs: IoU increases from 0.88 to 0.93, and Normal Consistency (NC) improves from 0.93 to 0.96. In comparison, fine-tuning only the fully connected layer offers virtually no benefit. Adding the cross-attention block introduces a clear gain—its final NC reaches 0.95, close to the best model—but at the cost of doubling the training time.

Table 5: Comparison of fine-tuning strategies on denoising performance.

Metric	FC Layer Only	FC + Cross-Attention Block	Entire Decoder
IoU	0.88 (no gain)	0.92	**0.93**
NC	0.93 (no gain)	0.95	**0.96**
Epochs	–	30	**15**

Noise Type. Beyond standard zero-mean Gaussian noise, we evaluated three additional noise types—Uniform, Discrete, and Laplace noise—each applied at a fixed magnitude of $\sigma = 0.01$. Furthermore, to assess the impact of non-zero bias in Gaussian perturbations, we conducted experiments over the domain $[-1, 1]^3$ using means of $\mu = 0.005$, 0.01, and 0.02. Comprehensive quantitative results are presented in Table 6.

At $\sigma = 0.01$ on $[-1, 1]^3$, our model consistently shows denoising performance under Uniform and Discrete noise, with notable gains in both IoU and NC. However, it shows limited benefit under Laplace noise. For Gaussian noise with $\sigma = 0.01$ increasing μ, our model remains effective at lower μ, but its advantage diminishes as the μ grows, eventually leading to degraded reconstruction quality.

Table 6: Performance comparison between the 3DS2V [47] and our method under various noise types and levels. Noise distributions tested are Uniform, Discrete and Laplace with $\sigma = 0.01$, and Gaussian with $\sigma = 0.01$ under different means $\mu = 0, 0.005, 0.01, 0.02$.

Dataset	IoU ↑		NC ↑		Chamfer		F-Score	
	3DS2V	Ours	3DS2V	Ours	3DS2V	Ours	3DS2V	Ours
Uniform ($\sigma = 0.01$)	0.873	**0.911**	0.920	**0.962**	0.015	**0.013**	0.985	**0.986**
Discrete ($\sigma = 0.01$)	0.867	**0.895**	0.915	**0.960**	0.015	**0.014**	0.984	**0.985**
Laplace ($\sigma = 0.01$)	0.909	0.908	0.956	**0.956**	0.014	**0.014**	0.985	**0.985**
Gaussian ($\sigma = 0.01$)	0.887	**0.927**	0.937	**0.966**	0.014	**0.013**	0.986	**0.986**
Gaussian ($\sigma = 0.01, \mu = 0.005$)	0.860	**0.881**	0.942	**0.964**	0.017	**0.016**	0.982	**0.984**
Gaussian ($\sigma = 0.01, \mu = 0.01$)	0.798	**0.800**	0.946	**0.949**	0.024	**0.023**	0.952	**0.960**
Gaussian ($\sigma = 0.01, \mu = 0.02$)	0.661	**0.666**	0.875	**0.898**	**0.038**	0.039	**0.549**	0.524

6 Conclusion

In this work, we introduced NoiseSDF2NoiseSDF, a framework capable of recovering clean surfaces from noisy and sparse point clouds by leveraging paired noisy SDFs in a Noise2Noise denoising approach. We demonstrated that, at noise intensities of 0.01 and 0.02, the reconstructed surfaces produced by our method are notably cleaner and smoother, both quantitatively and visually, compared to previous works. In future research, we aim to explore additional applications of the NoiseSDF2NoiseSDF framework, such as scaling up point cloud sizes to enhance geometric detail recovery, or replacing components within the framework with alternative architectures trained from scratch, thus improving the model's ability to represent point cloud noise and further enhancing denoising performance.

References

[1] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2565–2574, 2020.

[2] Matan Atzmon and Yaron Lipman. Sald: Sign agnostic learning with derivatives. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=7EDgLu9reQD.

[3] Joshua Batson and Loic Royer. Noise2Self: Blind denoising by self-supervision. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 524–533. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/batson19a.html.

[4] Yizhak Ben-Shabat, Chamin Hewa Koneputugodage, and Stephen Gould. Digs: Divergence guided shape implicit neural representation for unoriented point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19301–19310. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01872. URL https://doi.org/10.1109/CVPR52688.2022.01872.

[5] Ashish Bora, Eric Price, and Alexandros G. Dimakis. AmbientGAN: Generative models from lossy measurements. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hy7fDog0b.

[6] Alexandre Boulch and Renaud Marlet. Poco: Point convolution for surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6302–6314, 2022.

[7] Rohan Chabra, Jan E. Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX*, pages 608–625. Springer, 2020.

[8] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[9] Chao Chen, Yu-Shen Liu, and Zhizhong Han. Latent partition implicit with surface codes for 3d representation. In *European Conference on Computer Vision (ECCV) 2022*, pages 322–343. Springer, 2022.

[10] Chao Chen, Zhizhong Han, and Yu-Shen Liu. Inferring neural signed distance functions by overfitting on single noisy point clouds through finetuning data-driven based priors. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[11] Zhiqin Chen. Im-net: Learning implicit fields for generative shape modeling. 2019.

[12] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6970–6981, 2020.

[13] Angela Dai, Christian Diller, and Matthias Nießner. Sg-nn: Sparse generative neural networks for self-supervised scene completion of rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 849–858, 2020.

[14] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J. Mitra, and Michael Wimmer. Points2surf: Learning implicit surfaces from point clouds. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V*, volume 12350 of *Lecture Notes in Computer Science*, pages 108–124. Springer, 2020. doi: 10.1007/978-3-030-58558-7_7. URL https://doi.org/10.1007/978-3-030-58558-7_7.

[15] Philipp Erler, Lizeth Fuentes-Perez, Pedro Hermosilla, Paul Guerrero, Renato Pajarola, and Michael Wimmer. Ppsurf: Combining patches and point convolutions for detailed surface reconstruction. In *Computer Graphics Forum*, volume 43, page e15000. Wiley, 2024.

[16] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4857–4866, 2020.

[17] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13–18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3789–3799. PMLR, 2020. URL http://proceedings.mlr.press/v119/gropp20a.html.

[18] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018.

[19] 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.

[20] 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 (CVPR)*, pages 14781–14790, 2021. doi: 10.1109/CVPR46437.2021.01454. URL https://openaccess.thecvf.com/content/CVPR2021/html/Huang_Neighbor2Neighbor_Self-Supervised_Denoising_From_Single_Noisy_Images_CVPR_2021_paper.html.

[21] Haiyong Jiang, Jianfei Cai, Jianmin Zheng, and Jun Xiao. Neighborhood-based neural implicit reconstruction from point clouds. In *2021 International Conference on 3D Vision (3DV)*, pages 1259–1268. IEEE, 2021.

[22] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9601–9611, 2019.

[23] 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.

[24] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. In *Proceedings of the 35th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 2965–2974, 2018.

[25] Shengtao Li, Ge Gao, Yudong Liu, Ming Gu, and Yu-Shen Liu. Implicit filtering for learning neural signed distance functions from 3d point clouds. In *European Conference on Computer Vision*, pages 234–251. Springer, 2024.

[26] Siyou Lin, Dong Xiao, Zuoqiang Shi, and Bin Wang. Surface reconstruction from point clouds without normals by parametrizing the gauss formula. *ACM Transactions on Graphics*, 42(2):1–19, 2022.

[27] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169. ACM, 1987.

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

[29] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. *arXiv preprint arXiv:2011.13495*, 2020.

[30] Baorui Ma, Yu-Shen Liu, and Zhizhong Han. Learning signed distance functions from noisy 3d point clouds via noise to noise mapping. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 23338–23357. PMLR, 2023. URL https://proceedings.mlr.press/v202/ma23d.html.

[31] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019.

[32] Zhenxing Mi, Yiming Luo, and Wenbing Tao. Ssrnet: Scalable 3d surface reconstruction network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 970–979, 2020.

[33] Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. Noisier2noise: Learning to denoise from unpaired noisy data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12064–12072, 2020.

[34] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[35] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III*, pages 523–540. Springer, 2020.

[36] Songyou Peng, Chiyu Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as points: A differentiable poisson solver. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, pages 13032–13044, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/6cd9313ed34ef58bad3fdd504355e72c-Abstract.html.

[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.

[38] 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.

[39] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Carsten Stoll, and Christian Theobalt. Patchnets: Patch-based generalizable deep implicit 3d shape representations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 293–309. Springer, 2020.

[40] Weijia Wang, Xiao Liu, Hailing Zhou, Lei Wei, Zhigang Deng, M. Manzur Murshed, and Xuequan Lu. Noise4denoise: Leveraging noise for unsupervised point cloud denoising. *Computational Visual Media*, 10(4):659–669, 2024. doi: 10.1007/S41095-024-0423-3. URL https://doi.org/10.1007/s41095-024-0423-3.

[41] Zixiong Wang, Pengfei Wang, Pengshuai Wang, Qiujie Dong, Junjie Gao, Shuangmin Chen, Shiqing Xin, Changhe Tu, and Wenping Wang. Neural-imls: Self-supervised implicit moving least-squares network for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[42] Xiangbin Wei. Noise2score3d: Tweedie's approach for unsupervised point cloud denoising. *arXiv preprint arXiv:2503.09283*, 2025. URL https://arxiv.org/abs/2503.09283.

[43] Yaochen Xie, Zhengyang Wang, and Shuiwang Ji. Noise2same: Optimizing a self-supervised bound for image denoising. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/ea6b2efbdd4255a9f1b3bbc6399b58f4-Abstract.html.

[44] Xingguang Yan, Liqiang Lin, Niloy J Mitra, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Shapeformer: Transformer-based shape completion via sparse representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6239–6249, 2022.

[45] Biao Zhang and Peter Wonka. Lagem: A large geometry model for 3d representation learning and diffusion. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=72OSO38a2z.

[46] Biao Zhang, Matthias Nießner, and Peter Wonka. 3dilg: Irregular latent grids for 3d generative modeling. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/894ca1c4bc1c6abc4d4998ab94635fdf-Abstract-Conference.html.

[47] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)*, 42(4):1–16, 2023.

[48] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

[49] Wenbin Zhao, Jiabao Lei, Yuxin Wen, Jianguo Zhang, and Kui Jia. Sign-agnostic implicit learning of surface self-similarities for shape modeling and reconstruction from raw point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10256–10265, 2021.

[50] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately state the main contributions of the paper, including the proposal of NSDF2NSDF—a denoising method for signed distance Fields based on the Noise2Noise framework. The claims are supported by comprehensive experiments across multiple benchmark datasets.

   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 offer the limitation discussion in Appendix A.1.

   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: Our work is based on the theoretical analogy between neural fields and images, motivating the idea that the Noise2Noise framework can similarly be applied to neural fields. While we do not provide a formal mathematical proof of this assumption, our subsequent experiments empirically support its validity.

   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 offer detail in 4.2 and 5.1

   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: The code is included in our supplementary materials. We will release the full source code, dataset, and detailed instructions for reproduction.

   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 all the experiment setup in 5

   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 did not include confidence intervals or error bars due to the relatively long evaluation time required for each run. However, we observed that the results across repeated runs within the same experimental setup are highly consistent, with minimal variation. This empirical stability gives us confidence in the reliability of the reported outcomes, even without formal statistical testing.

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: We provide detailed information in 5.1.

   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: We have reviewed the NeurIPS Code of Ethics and confirm that our research fully conforms to its guidelines. Our work does not involve human subjects, personally identifiable data, or sensitive content. All datasets used are publicly available, and we have taken care to ensure that our method does not pose foreseeable risks related to safety, discrimination, or misuse. We also plan to release our code and models to support transparency and reproducibility.

   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 provide broader impacts in Appendix A.2.

    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: There is no such risk for our work.

    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 properly cite each dataset, pretrained model used in our paper and use them under their licence.

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: No assets released in our work.

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: Our work focuses solely on point cloud denoising and reconstruction and does not involve any crowdsourcing or human-subject experiments.

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: Our research does not involve any human subjects or crowdsourcing experiments. Therefore, no IRB or equivalent approvals were required.

    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: [NA]

    Justification: Our research does not involve large language models in any component of the core methodology.

    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.