Learning Signed Distance Functions from Noisy 3D Point Clouds via Noise to Noise Mapping

Baorui Ma [1]   **Yu-Shen Liu** [1]   **Zhizhong Han** [2]

(a) Noisy Points   (b) Denoised Points   (c) Surface Reconstruction
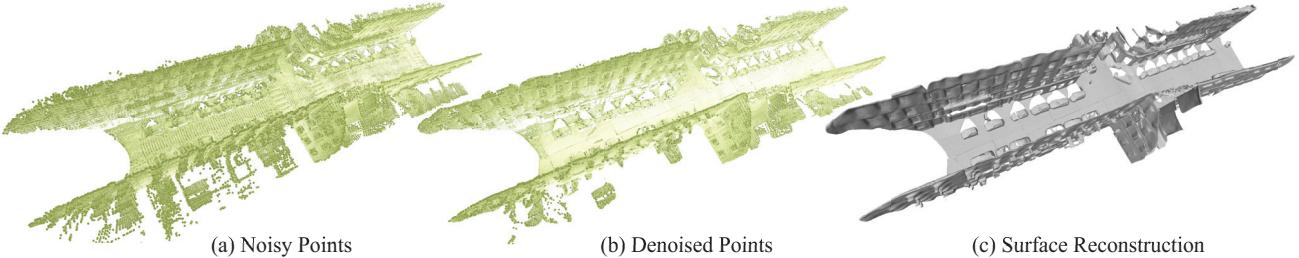
Figure 1. We introduce to learn signed distance functions (SDFs) for single noisy point clouds. Our method does not require ground truth signed distances, point normals or clean points as supervision for training. We achieve this via learning a mapping from one noisy observation to another or even on a single observation. Our novel learning manner is supported by modern Lidar systems which capture 10 to 30 noisy observations per second. We show the SDF learned from (a) a single real scan containing $10M$ points, (b) the denoised point cloud and (c) the reconstructed surface. Fig. 12 demonstrates our superiority over the latest surface reconstructions in this case.

Abstract

Learning signed distance functions (SDFs) from 3D point clouds is an important task in 3D computer vision. However, without ground truth signed distances, point normals or clean point clouds, current methods still struggle from learning SDFs from noisy point clouds. To overcome this challenge, we propose to learn SDFs via a noise to noise mapping, which does not require any clean point cloud or ground truth supervision for training. Our novelty lies in the noise to noise mapping which can infer a highly accurate SDF of a single object or scene from its multiple or even single noisy point cloud observations. Our novel learning manner is supported by modern Lidar systems which capture multiple noisy observations per second. We achieve this by a novel loss which enables statistical reasoning on point clouds and maintains geometric consistency although point clouds are irregular, unordered and have no point correspondence among noisy observations. Our evaluation under the widely used benchmarks demonstrates our superiority

over the state-of-the-art methods in surface reconstruction, point cloud denoising and upsampling. Our code, data, and pre-trained models are available at `https://github.com/mabaorui/Noise2NoiseMapping/`.

1. Introduction

3D point clouds have been a popular 3D representation. We can capture 3D point clouds not only on unmanned vehicles, such as self-driving cars, but also from consumer level digital devices in our daily life, such as the iPhone. However, the raw point clouds are discretized and noisy, which is not friendly to downstream applications like virtual reality and augmented reality requiring clean surfaces. This results in a large demand of learning signed distance functions (SDFs) from 3D point clouds, since SDFs are continuous and also capable of representing arbitrary 3D topology.

Deep learning based methods have shown various solutions of learning SDFs from point clouds (Gropp et al., 2020; Atzmon & Lipman, 2020; Ma et al., 2021; Jiang et al., 2020a; Peng et al., 2021). Different from classic methods (Kazhdan & Hoppe, 2013; Ohtake et al., 2003), they mainly leverage data-driven strategy to learn various priors from large scale dataset using deep neural networks. They usually require the signed distance ground truth (Liu et al., 2021), point normals (Jiang et al., 2020a; Chabra et al., 2020; Peng et al., 2021), additional constraints (Gropp et al., 2020; Atzmon & Lipman, 2020) or no noise assumption (Ma et al., 2021). These requirements significantly affect the accuracy of S-DFs learned for noisy point clouds, either caused by poor

---

[1]School of Software, Tsinghua University, Beijing, China [2]Department of Computer Science, Wayne State University, Detroit, USA. Correspondence to: Yu-Shen Liu <liuyushen@tsinghua.edu.cn>.

generalization or the incapability of denoising. Therefore, it is still challenging to learn SDFs from noisy point clouds without clean or ground truth supervision.

To overcome this challenge, we introduce to learn SDFs from noisy point clouds via noise to noise mapping. Our method does not require ground truth signed distances and point normals or clean point clouds to learn priors. As demonstrated in Fig. 1, our novelty lies in the way of learning a highly accurate SDF for a single object or scene from its several corrupted observations, i.e., noisy point clouds. Our learning manner is supported by modern Lidar systems which produce about 10 to 30 corrupted observations per second. By introducing a novel loss function containing a geometric consistency regularization, we are enabled to learn a SDF via a task of learning a mapping from one corrupted observation to another corrupted observation or even a mapping from one corrupted observation to the observation itself. The key idea of this noise to noise mapping is to leverage the statistical reasoning to reveal the uncorrupted structures upon its several corrupted observations. One of our contribution is the finding that we can still conduct statistical reasoning even there is no spatial correspondence among points on different corrupted observations. Our results achieve the state-of-the-art in different applications including surface reconstruction, point cloud denoising and upsampling under widely used benchmarks. Our contributions are listed below.

i) We introduce a method to learn SDFs from noisy point clouds without requiring ground truth signed distances, point normals or clean point clouds.

ii) We prove that we can leverage Earth Mover's Distance (EMD) to perform the statistical reasoning via noise to noise mapping and justify this idea using our novel loss function, even if 3D point clouds are irregular, unordered and have no point correspondence among different observations.

iii) We achieved the state-of-the-art results in surface reconstruction, point cloud denoising and upsampling for shapes or scenes under the widely used benchmarks.

2. Related Work

Learning implicit functions for 3D shapes and scenes has made great progress (Mildenhall et al., 2020; Oechsle et al., 2021; Han et al., 2020b; Chen et al., 2021; Xiang et al., 2021; Takikawa et al., 2021; Martel et al., 2021; Rematas et al., 2021; Feng et al., 2022; Han et al., 2020a; Wen et al., 2022; Li et al., 2023b; Han et al., 2020c; Wen et al., 2020; 2021; Zhang et al., 2023b; Li et al., 2023a; 2022a; LP-Sayed et al., 2022; Stier et al., 2023; Shue et al., 2023; Zhang et al., 2023a; Gupta et al., 2023; Rosu & Behnke, 2023; Zhou et al., 2022b). We briefly review methods with different supervision below.

Learning from 3D Supervision. It was explored on how to learn implicit functions, i.e., SDFs or occupancy fields, using 3D supervision including signed distances (Michalkiewicz et al., 2019; Park et al., 2019; Ouasfi & Boukhayma, 2022; Li et al., 2022c) and binary occupancy labels (Mescheder et al., 2019; Chen & Zhang, 2019). With a condition, such as a single image (Wang et al., 2019; Saito et al., 2019; Chibane et al., 2020a; Littwin & Wolf, 2019; Genova et al., 2019; Han et al., 2020d) or a learnable latent code (Park et al., 2019), neural networks can be trained as an implicit function to model various shapes. We can also leverage point clouds as conditions (Williams et al., 2019; Liu et al., 2020a; Mi et al., 2020; Genova et al., 2019) to learn implicit functions, and then leverage the marching cubes algorithm (Lorensen & Cline, 1987) to reconstruct surfaces (Jia & Kyan, 2020; Erler et al., 2020). To capture more detailed geometry, implicit functions are defined in local regions which are covered by voxel grids (Jiang et al., 2020a; Chabra et al., 2020; Songyou Peng, 2020; Martel et al., 2021; Takikawa et al., 2021; Liu et al., 2021; Tang et al., 2021), patches (Tretschk et al., 2020), 3D Gaussian functions (Genova et al., 2020), learnable codes (Li et al., 2022b; Boulch & Marlet, 2022).

Learning from 2D Supervision. We can also learn implicit functions from 2D supervision, such as multiple images. The basic idea is to leverage various differentiable renderers (Sitzmann et al., 2019; Liu et al., 2020b; Jiang et al., 2020b; Zakharov et al., 2020; Liu et al., 2019; Wu & Sun, 2020; Niemeyer et al., 2020; Lin et al., 2020) to render the learned implicit functions into images, so that we can obtain the error between rendered images and ground truth images. Neural volume rendering was introduced to capture the geometry and color simultaneously (Mildenhall et al., 2020; Yariv et al., 2020; 2021; Fu et al., 2022; Wang et al., 2021; Yu et al., 2022; Wang et al., 2022b; Vicini et al., 2022; Wang et al., 2022a; Guo et al., 2022).

Learning from 3D Point Clouds. Some methods were proposed to learn implicit functions from point clouds without 3D ground truth. These methods leverage additional constraints (Gropp et al., 2020; Atzmon & Lipman, 2020; Zhao et al., 2020; Atzmon & Lipman, 2021; Ben-Shabat et al., 2021; Yifan et al., 2020; Ben-Shabat et al., 2022), gradients (Ma et al., 2021; Chibane et al., 2020b), differentiable poisson solver (Peng et al., 2021) or specially designed priors (Ma et al., 2022a;b) to learn signed (Ma et al., 2021; Gropp et al., 2020; Atzmon & Lipman, 2020; Zhao et al., 2020; Atzmon & Lipman, 2021; Chen et al., 2022; Pumarola et al., 2022; Chen et al., 2023; Ma et al., 2023) or unsigned distance fields (Chibane et al., 2020b; Zhou et al., 2022a). One issue here is that they usually assume the point clouds are clean, which limits their performance in real applications due to the noise. Our method falls into this category, but we can resolve this problem using statistical reasoning via noise to noise mapping.
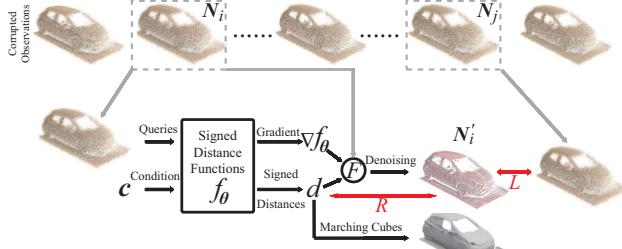
2

Figure 2. Given corrupted observations captured by a Lidar system per second, we learn a SDF without supervision or normals.

Deep Learning based Point Cloud Denoising. Point-CleanNet (Rakotosaona et al., 2020) was introduced to remove outliers and reduce noise from point clouds using a data-driven strategy. Graph convolution was also leveraged to reduce the noise based on dynamically constructed neighborhood graphs (Pistilli et al., 2020). Without supervision, TotalDenoising (Casajus et al., 2019) inherits the same idea as Noise2Noise (Lehtinen et al., 2018a). It leveraged a spatial prior term that can work for unordered point clouds. More recently, downsample-upsample architecture (Luo & Hu, 2020) and gradient fields (Luo & Hu, 2021; Cai et al., 2020) were leveraged to reduce noise. We were inspired by the idea of Noise2Noise (Lehtinen et al., 2018a), our contribution lies in our finding that we can still leverage statistical reasoning among multiple noisy point clouds with specially designed losses even there is no spatial correspondence among points on different observations like the one among pixels, which is totally different from TotalDenoising (Casajus et al., 2019).

3. Method

Overview. Given N corrupted observations $S = \{N_i | i \in [1, N], N \geq 1\}$ of an uncorrupted 3D shape or scene S, we aim to learn SDFs f of S from S without ground truth signed distances, point normals, or clean point clouds. Here, N_i is a noisy point cloud. SDFs f predicts a signed distance d for an arbitrary query location $q \in \mathbb{R}^{1 \times 3}$ around S, such that $d = f(q, c)$, where c is a condition denoting S. We train a neural network parameterized by θ to learn f, which we denote as f_θ. After training, we can further leverage the learned f_θ for surface reconstruction, point cloud denoising, and point cloud upsampling.

Our key idea of statistical reasoning is demonstrated in Fig. 2. Using a noisy point cloud N_i as input, our network aims to learn SDFs f_θ via learning a noise to noise mapping from N_i to another noisy point cloud N_j, where N_j is also randomly selected from the corrupted observation set S and $j \in [1, N]$. Our loss not only minimizes the distance between the denoised point cloud N_i' and N_j using a metric L but also constrains the learned SDFs f_θ to be correct using a geometric consistency regularization R. A denoising function F conducts point cloud denoising using signed

distances d and gradients ∇f_θ from f_θ.

Reducing Noise. A common strategy for estimating the uncorrupted data from its noise corrupted observations is to find a target that has the smallest average deviation from measurements according to some loss function L. The data could be a scalar, a 2D image or a 3D point cloud etc.. Here, to reduce noise on point clouds, we aim to find the uncorrupted point cloud N' from its corrupted observations $N \in S$ below,

$$\underset{N'}{\text{argmin}} \, \mathbb{E}_N \{L(N', N)\}. \tag{1}$$

As a conclusion of Noise2Noise (Lehtinen et al., 2018a) for 2D image denoising, we can learn a denoising function F by pushing a denoised image $F(x)$ to be similar to as many corrupted observations y as possible, where both x and y are corrupted observations. This is an appealing conclusion since we do not need the expensive pairs of the corrupted inputs and clean targets to learn the denoising function F.

We want to leverage this conclusion to learn to reduce noise without requiring clean point clouds. So we transform Eq. (1) into an equation with a denoising function F,

$$\underset{F}{\text{argmin}} \sum_{N_i \in S} \sum_{N_j \in S} L(F(N_i), N_j). \tag{2}$$

One issue we are facing is that the conclusion of Noise2Noise may not work for 3D point clouds, due to the irregular and unordered characteristics of point clouds. For 2D images, multiple corrupted observations have the pixel correspondence. This results in an assumption that all noisy observations at the same pixel location are random realizations of a distribution around a clean pixel value. However, this assumption is invalid for point clouds. This is also the reason why TotalDenoising (Casajus et al., 2019) does not think Eq. (1) can work for point cloud denoising, since the noise in 3D point clouds is total. Differently, our finding is in opposite direction. We think we can still leverage Eq. (1) to reduce noise in 3D point clouds, and the key is how to define the distance metric L, which is regarded as one of our contributions.

Another issue that we are facing is how we can learn SDFs f_θ via point cloud denoising in Eq. (2). Our solution is to leverage f_θ to define the denoising function F. This enables to conduct the learning of SDFs and point cloud denoising at the same time. Next, we will elaborate on our solutions to the aforementioned two issues.

Denoising Function F. The denoising function F aims to produce a denoised point cloud N' from a noisy point cloud N, so $N' = F(N)$.

To learn SDFs f_θ of N, we want the denoising procedure can also perceive the signed distance fields around N. The
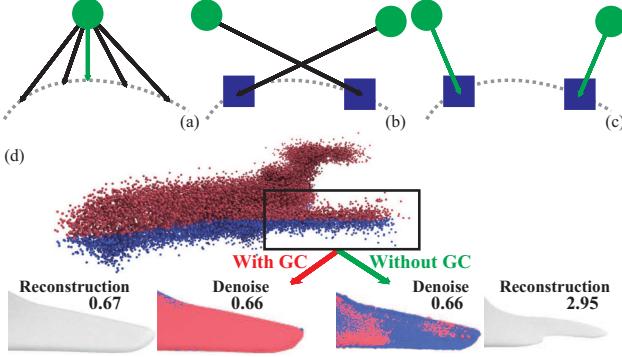
Figure 3. (a) Multiple paths (arrows) to pull a noise (green point) onto surface (dashed curve) but only one is the shortest (green arrows). (b) The incorrect paths (black arrows) to pull noises onto surface. (c) The expected paths (green arrows) to pull noises to points (blue square) on surface. (d) The effect of Geometric Consistency (GC).

essence of denoising is to move points floating off the surface of an object onto the surface. As shown in Fig. 3 (a), there are many potential paths to achieve this, but only one path is the shortest to the surface. If we leverage this shortest path to denoise point cloud N, we could involve the SDFs f_θ to define the denoising function F, since f_θ can determine the shortest path.

Here, inspired by the idea of NeuralPull (Ma et al., 2021), we also leverage the signed distance $d = f_\theta(n, c)$ and the gradient $\nabla f_\theta(n, c)$ to pull an arbitrary point n on the noisy point cloud N onto the surface. So we define the denoising function F below,

$$F(n, f_\theta) = n - d \times \nabla f_\theta(n, c)/||\nabla f_\theta(n, c)||_2. \quad (3)$$

With Eq. (3), we can pull all points on the noisy point cloud N onto the surface, which results in a point cloud $N' = F(N, f_\theta)$. But one issue remaining is how to constrain N' to converge to the uncorrupted surface.

Distance Metric L. We investigate the distance metric L so that we can constrain N' to reveal the uncorrupted surface by a statistical reasoning among the corrupted observations $S = \{N_i\}$ using Eq. (2). Our investigation conclusion is summarized in the following Theorem.

Theorem 1. *Assume there was a clean point cloud G which is corrupted into observations $S = \{N_i\}$ by sampling a noise around each point of G. If we leverage EMD as the distance metric L defined in Eq. (4), and learn a point cloud G' by minimizing the EMD between G' and each observation in S, i.e., $\min_{G'} \sum_{N_i \in S} L(G', N_i)$, then G' converges to the clean point cloud G, i.e., $L(G, G') = 0$.*

$$L(G, G') = \min_{\phi:G \to G'} \sum_{g \in G} ||g - \phi(g)||_2. \quad (4)$$

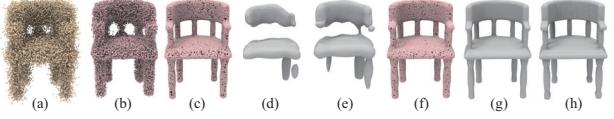We prove Theorem 1 in the following appendix. We believe

Figure 4. The comparison with CD and EMD as the distance metric L from in (b) to (e). The effect of geometric regularization in (f) and (g). (a) is noisy point cloud, (h) is the ground truth.

the one-to-one correspondence ϕ found in the calculation of EMD in Eq. (4) plays a big role in the statistical reasoning for denoising. This is very similar to the pixel correspondence among noisy images in Noise2Noise although point clouds are irregular, unordered and have no spatial correspondence among points on different observations. We highlight this by comparing the point cloud G' optimized with EMD and Chamfer Distance (CD) as L based on the same observation set S in Fig. 4. Given noisy point clouds N_i like in Fig. 4 (a), Fig. 4 (b) demonstrates that the point cloud G' optimized with CD is still noisy, while the one optimized with EMD in Fig. 4 (c) is very clean.

According to this theorem, we can learn the denoising function F using Eq. (2). F produces the denoised point cloud $N_i' = F(N_i, f_\theta)$ using EMD as the distance metric L. This also leads to one term in our loss function below,

$$\min_\theta \sum_{N_i \in S} \sum_{N_j \in S} L(F(N_i, f_\theta), N_j). \quad (5)$$

Geometric Consistency. Although the term in Eq. (5) can work for point cloud denoising well, as shown in Fig. 4 (c), we found that the SDFs f_θ may not describe a correct signed distance field. With f_θ either learned with CD or EMD, the surfaces reconstructed using marching cubes algorithms (Lorensen & Cline, 1987) in Fig. 4 (d) and (e) are poor. This is because Eq. (5) only constrains that points on the noisy point cloud should arrive onto the surface but there are no constraints on the paths to be the shortest. This is caused by the unawareness of the true surface which however is required as the ground truth by NeuralPull (Ma et al., 2021). The issue is further demonstrated in Fig. 3, one situation that may happen is shown in Fig. 3 (b). With the wrong signed distances f_θ and gradient ∇f_θ, noises can also get pulled onto the surface, which results in a denoised point cloud with zero EMD distance to the clean point clouds. This is much different from the correct signed distance field that we expected in Fig. 3 (c).

To resolve this issue, we introduce a geometric consistency to constrain f_θ to be correct. Our insight here is that, for an arbitrary query n around a noisy point cloud N_i, the shortest distance between n and the surface can be either predicted by the SDFs f_θ or calculated based on the denoised point cloud $N_i' = F(N_i, f_\theta)$, both of which should be consistent to each other. Therefore, the absolute value $|f_\theta(n, c)|$ of the signed distance predicted at n should equal to the minimum distance between n and the denoised point cloud $N_i' =$

$F(\boldsymbol{N}_i, f_\theta)$. Since the point density of $\boldsymbol{N}_i'$ may slightly affect the consistency, we leverage an inequality to describe the geometric consistency,

$$|f_\theta(\boldsymbol{n}, \boldsymbol{c})| \leq \min_{\boldsymbol{n}' \in F(\boldsymbol{N}_i, f_\theta)} ||\boldsymbol{n} - \boldsymbol{n}'||_2. \quad (6)$$

The geometric consistency is further illustrated in Fig. 3 (d). Noisy points above/below the wing can be correctly pulled onto the upper/lower surface without crossing the wing using the geometric consistency. It achieves the same denoising performance, and leads to a much more accurate SDF for surface reconstruction than the one without the geometric consistency.

Loss Function. With the geometric consistency, we can penalize the incorrect signed distance field shown in Fig. 3 (b) while encouraging the correct one in Fig. 3 (c). So, we leverage the geometric consistency as a regularization term R, which leads to our objective function below by combining Eq. (5) and Eq. (6),

$$\min_\theta \sum_{\boldsymbol{N}_i \in S} \left(\sum_{\boldsymbol{N}_j \in S} L(F(\boldsymbol{N}_i, f_\theta), \boldsymbol{N}_j) + \frac{\lambda}{|\boldsymbol{N}_i|} \sum_{\boldsymbol{n} \in \boldsymbol{N}_i} R(E), \right.$$
$$(7)$$

where $|\boldsymbol{N}_i|$ is the number of $\boldsymbol{n}$ on $\boldsymbol{N}_i$, E is the difference defined as $(|f_\theta(\boldsymbol{n}, \boldsymbol{c})| - \min_{\boldsymbol{n}' \in F(\boldsymbol{N}_i, f_\theta)} ||\boldsymbol{n} - \boldsymbol{n}'||_2)$, λ is a balance weight, and $R(E) = max(0, E)$. The effect of the geometric consistency is demonstrated in Fig. 4 (f) and (g). The denoised point cloud in Fig. 4 (f) shows points that are more uniformly distributed, compared with the one obtained without the geometric consistency in Fig. 4 (c). More importantly, we can learn correct SDFs f_θ to reconstruct plausible surface in Fig. 4 (g), compared to the one obtained without the geometric consistency in Fig. 4 (e) and the ground truth in Fig. 4 (h).

More Details. We sample more queries around the input noisy point cloud $\boldsymbol{N}_i$ using the method introduced in NeuralPull (Ma et al., 2021). We randomly sample a batch of B queries as input, and also randomly sample the same number of points from another noisy point cloud $\boldsymbol{N}_j$ as target. Using batches enables us to process large scale point clouds, makes it possible to leverage noisy point clouds with different point numbers even we use EMD as the distance metric L, and more importantly, does not affect the performance. We train f_θ to overfit to a single shape or scene or overfit to multiple shapes or scenes using conditions $\boldsymbol{c}$ to indicate different shapes or scenes.

We visualize the optimization process in 4 epochs in Fig. 5 (a). We show how the 3 queries (black cubes) get pulled progressively onto the surface (Cyan). For each query, we also show its corresponding target in each one of 100 batches in the same color (red, green, blue), and each target is established by the mapping ϕ in the metric L. The essence of statical reasoning in each epoch is that each query will

Points	5K			10K		
	PU-Net	SBP	**Ours**	PU-Net	SBP	**Ours**
CD	3.445	1.696	**0.592**	2.862	1.454	**0.418**
P2M	1.669	0.295	**0.156**	1.166	0.181	**0.155**

Table 2. Upsampling comparison. L2CD$\times 10^4$ and P2M $\times 10^4$.

be pulled to the average point of all targets from all batches since the distance between the query and each target should be minimized. Although the targets are found all over the shape in the first epoch, the targets surround the query more tightly as the query gets pulled to the surface in the following epochs. This makes queries get pulled onto the surface which results in an accurate SDF visualized in the surface reconstruction and level-sets in Fig. 5 (b).

One Noisy Point Cloud. Although we prove Theorem 1 based on multiple noisy point clouds ($N > 1$), we surprisingly found that our method can also work well when only one noisy point cloud ($N = 1$) is available. Specifically, we regard the queries sampled around the noisy point cloud $\boldsymbol{N}_i$ as input and regard $\boldsymbol{N}_i$ as target. We believe the reason why $N = 1$ works is that the knowledge learned via statistical reasoning in the batch based training can be well generalized to various regions. We will report our results learned from multiple or one noisy point clouds in experiments.

Noise Types. We work well with different types of noises in Fig. 6. We use zero-mean noises in our proof of Theorem 1, but we find we work well with unknown noises in real scans in experiments. In evaluations, we also use the same type of noises in benchmarks for fair comparisons.
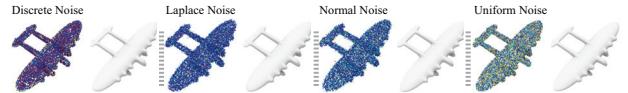

Figure 6. Reconstruction with different kinds of noises.

4. Experiments and Analysis

We evaluate our method in two steps. We first evaluate our method in applications that only care about points, such as point cloud denoising and upsampling. So, we only leverage Eq. (5) to produce the denoised or upsampled point clouds. Then, we evaluate our method trained with the loss in Eq. (7) in surface reconstruction, where $\lambda = 0.1$.

4.1. Point Cloud Denoising

Dataset and Metric. For the fair comparison with the state-of-the-art results, we follow SBP (Luo & Hu, 2021) to evaluate our method under two benchmarks named as PU and PC that were released by PUNet (Yu et al., 2018) and PointCleanNet (Rakotosaona et al., 2020). We report our results under 20 shapes in the test set of PU and 10 shapes in the test set of PC. We use Poisson disk to sample $10K$ and $50K$ points from each shape respectively as the ground truth clean point clouds in two different resolutions. The

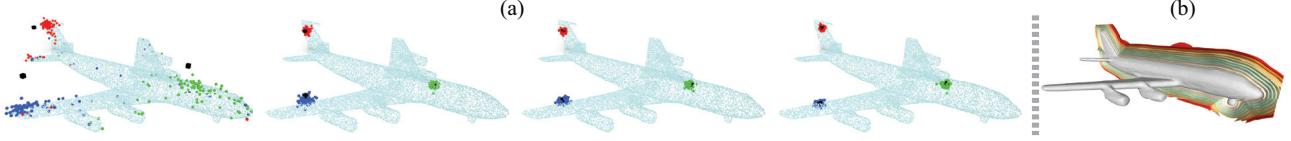| Point Number		10K(Sparse)						50K(Dense)					
Noise		1%		2%		3%		1%		2%		3%	
	Model	CD	P2M	CD	P2M	CD	P2M	CD	P2M	CD	P2M	CD	P2M
PU	Bilateral	3.646	1.342	5.007	2.018	6.998	3.557	0.877	0.234	2.376	1.389	6.304	4.730
	Jet	2.712	0.613	4.155	1.347	6.262	2.921	0.851	0.207	2.432	1.403	5.788	4.267
	MRPCA	2.972	0.922	3.728	1.117	5.009	1.963	0.669	**0.099**	2.008	1.003	5.775	4.081
	GLR	2.959	1.052	3.773	1.306	4.909	2.114	0.696	0.161	1.587	0.830	3.839	2.707
	PCNet	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	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	4.482	1.722	4.982	2.115	5.892	2.846	1.162	0.469	1.566	0.800	2.632	1.528
	SBP	2.521	0.463	3.686	1.074	4.708	1.942	0.716	0.150	1.288	0.566	1.928	1.041
	TTD-Un	3.390	0.826	7.251	3.485	13.385	8.740	1.024	0.314	2.722	1.567	7.474	5.729
	SBP-Un	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	**1.060**	**0.241**	**2.925**	**1.010**	**4.221**	**1.847**	**0.377**	0.155	**1.029**	**0.484**	**1.654**	**0.972**
PC	Bilaterall	4.320	1.351	6.171	1.646	8.295	2.392	1.172	0.198	2.478	0.634	6.077	2.189
	Jet	3.032	0.830	5.298	1.372	7.650	2.227	1.091	0.180	2.582	0.700	5.787	2.144
	MRPCA	3.323	0.931	4.874	1.178	6.502	1.676	0.966	0.140	2.153	0.478	5.570	1.976
	GLR	3.399	0.956	5.274	1.146	7.249	1.674	0.964	0.134	2.015	0.417	4.488	1.306
	PCNet	3.849	1.221	8.752	3.043	14.525	5.873	1.293	0.289	1.913	0.505	3.249	1.076
	GPDNet	5.470	1.973	10.006	3.650	15.521	6.353	5.310	1.716	7.709	2.859	11.941	5.130
	DMR	6.602	2.152	7.145	2.237	8.087	2.487	1.566	0.350	2.009	0.485	2.993	0.859
	SBP	3.369	0.830	5.132	1.195	6.776	1.941	1.066	0.177	1.659	0.354	2.494	**0.657**
	Ours	**2.047**	**0.518**	**2.056**	**0.519**	**5.331**	**1.935**	**0.426**	**0.129**	**1.043**	**0.316**	**2.22**	1.096

Table 1. Denoising comparison. L2CD$\times 10^4$ and P2M $\times 10^4$.

Figure 5. (a) Visualization of optimization in 4 epochs via noise to noise mapping. 3 queries (black cubes) sampled from one noisy point cloud get pulled onto the surface. For each query, we minimize its distance to all targets (in the same color) matched from another noisy point cloud by the mapping ϕ in metric L. More details can be found in our video. (b) Surface reconstruction and multiple level-sets.

clean point cloud is normalized into the unit sphere. In each resolution, we add Gaussian noise with three standard deviations including 1%, 2%, 3% to the clean point clouds. We leverage L2 Chamfer Distance (L2CD) and point to mesh distance (P2M) to evaluate the denoising performance. For each test shape, we generate $N = 200$ noisy point clouds to train our method. We sample $B = 250$ points in each batch. We report our results and numerical comparison in Tab. 1. The compared methods include Bilateral (Fleishman et al., 2003), Jet (Cazals & Pouget, 2005), MRPCA (Mattei & Castrodad, 2017), GLR (Zeng et al., 2020), PCNet (Rakotosaona et al., 2020), GPDNet (Pistilli et al., 2020), DMR (Luo & Hu, 2020), TTD (Casajus et al., 2019), and SBP (Luo & Hu, 2021). These methods require learned priors and can not directly use multiple observations. The comparison with different conditions indicates that our method significantly outperforms traditional point cloud denoising methods and deep learning based point cloud denoising methods in both supervised and unsupervised ("-Un") settings. Error map comparison with TTD (Casajus et al., 2019) and SBP (Luo & Hu, 2021) in Fig. 7 further demonstrates our state-of-the-art denoising performance.

4.2. Point Cloud Upsampling

Dataset and Metric. We use the PU dataset mentioned before to evaluate the f_{θ} learned in our denoising experiments in point cloud upsampling. Following SBP (Luo & Hu, 2021), we produce an upsampled point cloud with an upsampling rate of 4 from a sparse point cloud by denoising the sparse point cloud with noise. We compare the denoised point cloud and the ground truth, and report L2CD and P2M comparison in Tab. 2. We compared with PU-Net (Yu et al., 2018) and SBP (Luo & Hu, 2021). The comparison demonstrates that our method can perform the statistical reasoning to reveal points on the surface more accurately.

4.3. Surface Reconstruction for Shapes

ShapeNet. We first report our surface reconstruction performance under the test set of 13 classes in ShapeNet (Chang et al., 2015). The train and test splits follow COcc (Peng et al., 2020). Following IMLS (Liu et al., 2021), we lever-
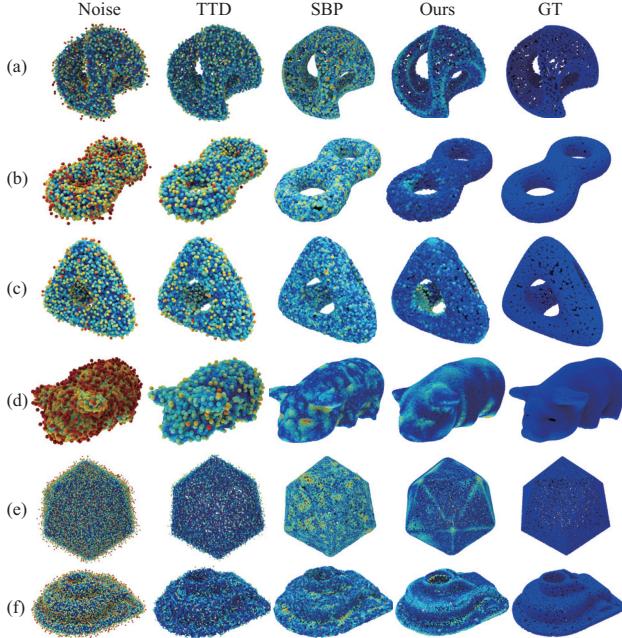
6

Figure 7. Visual comparison in point cloud denoising. Error at each point is shown in color. (a) and (b) $10K$ points with 3% noise. (c) $10K$ points with 2% noise. (d) and (e) $50K$ points with 3% noise. (f) $50K$ points with with 2% noise.

	PSR	PSG	R2N2	Atlas	COcc	SAP	OCNN	IMLS	POCO	**Ours**
L1CD×10	0.299	0.147	0.173	0.093	0.044	0.034	0.067	0.031	0.030	**0.026**
NC	0.772	-	0.715	0.855	0.938	0.944	0.932	0.944	0.950	**0.962**
F-Score	0.612	0.259	0.400	0.708	0.942	0.975	0.800	0.983	0.984	**0.991**

Table 3. L1CD, NC and F-Score comparison under ShapeNet.

age point clouds with 3000 points as clean truth, and add Gaussian noise with a standard deviation of 0.005. For each clean point cloud, we generate $N = 200$ noisy point clouds with a batch size of $B = 3000$. We leverage L1 Chamfer Distance (L1CD), Normal Consistency (NC) (Mescheder et al., 2019), and F-score (Tatarchenko et al., 2019) with a threshold of 1% as metrics.

We compare our methods with methods including P-SR (Kazhdan & Hoppe, 2013), PSG (Fan et al., 2017), R2N2 (Choy et al., 2016), Atlas (Groueix et al., 2018), COcc (Peng et al., 2020), SAP (Peng et al., 2021), OCNN (Wang et al., 2020), IMLS (Liu et al., 2021) and POCO (Boulch & Marlet, 2022). The numerical comparison in Tab. 3 demonstrates our state-of-the-art surface reconstruction accuracy over 13 classes. Although we do not require the ground truth supervision, our method outperforms the supervised methods such as SAP (Peng et al., 2021), COcc (Peng et al., 2020) and IMLS (Liu et al., 2021). We further demonstrate our superiority in the reconstruction of complex geometry in the visual comparison in Fig. 8. More numerical and visual comparisons can be found in the following appendix.

FAMOUS and ABC. We further evaluate our method using the test set in FAMOUS and ABC dataset provided by P2S (Erler et al., 2020). The clean point cloud is corrupted with noise at different levels. We follow NeuralPull (Ma
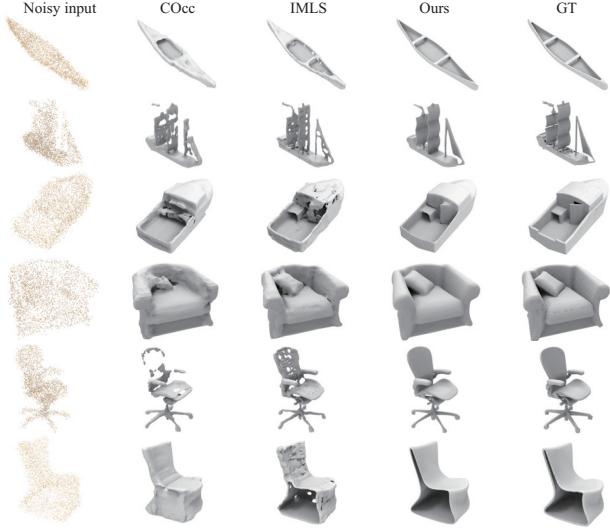

Figure 8. Comparison in surface reconstruction under ShapeNet.

Dataset	DSDF	Atlas	PSR	P2S	NP	IMLS	PCP	POCO	OnSF	Ours
ABC var	12.51	4.04	3.29	2.14	0.72	0.57	0.49	2.01	3.52	**0.113**
ABC max	11.34	4.47	3.89	2.76	1.24	0.68	0.57	2.50	4.30	**0.139**
F-med	9.89	4.54	1.80	1.51	0.28	0.80	0.07	1.50	0.59	**0.033**
F-max	13.17	4.14	3.41	2.52	0.31	0.39	0.30	2.75	3.64	**0.117**

Table 4. L2CD×100 comparison under ABC and Famous.

et al., 2021) to report L2 Chamfer Distance (L2CD). Different from previous experiments, we only leverage single $N = 1$ noisy point clouds to train our method with a batch size of $B = 1000$.

We compare our methods with methods including DS-DF (Park et al., 2019), Atlas (Groueix et al., 2018), P-SR (Kazhdan & Hoppe, 2013), P2S (Erler et al., 2020), NP (Ma et al., 2021), IMLS (Liu et al., 2021), PCP (Ma et al., 2022b), POCO (Boulch & Marlet, 2022), and OnS-F (Ma et al., 2022a). The comparison in Tab. 4 demonstrates that our method can reveal more accurate surfaces from noisy point clouds even we do not have training set, ground truth supervision or even multiple noisy point clouds. The statistical reasoning on point clouds and geometric regularization produce more accurate surfaces as demonstrated by the error map comparison under FAMOUS in Fig. 9.

D-FAUST and SRB. Finally, we evaluate our method under the real scanning dataset D-FAUST (Bogo et al., 2017) and
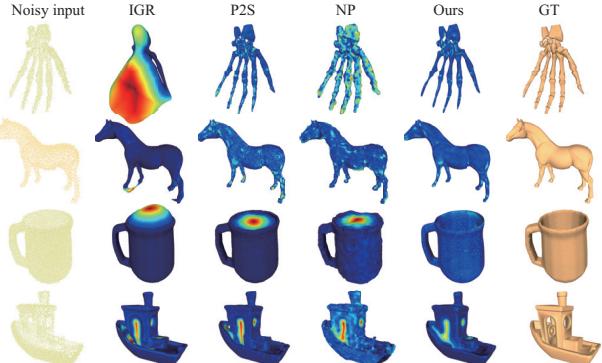

Figure 9. Visual comparison in surface reconstruction under FA-MOUS. Point to surface error at each vertex is shown in color.

7

Metrics	IGR	Point2Mesh	PSR	SAP	**Ours**
L1CD×10	0.235	0.071	0.044	0.043	**0.037**
F-Score	0.805	0.855	0.966	0.966	**0.996**
NC	0.911	0.905	0.965	0.959	**0.970**

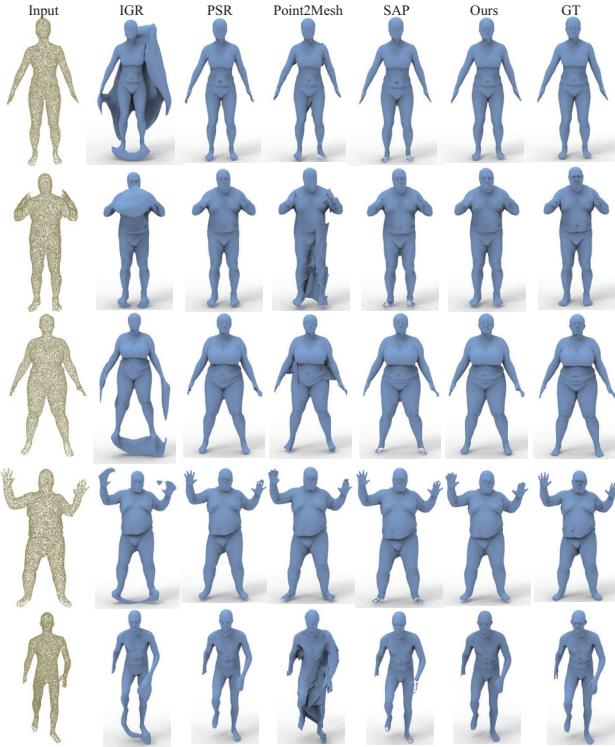*Table 5.* Comparison in surface reconstruction under D-FAUST.

Figure 10. Comparison in surface reconstruction under DFAUST.

SRB (Williams et al., 2019). We follow SAP (Peng et al., 2021) to evaluate our result using L1CD, NC (Mescheder et al., 2019), and F-score (Tatarchenko et al., 2019) with a threshold of 1% using the same set of shapes. We use single $N = 1$ noisy point clouds to train our method with a batch size of $B = 5000$.

We compare our methods with the methods including I-GR (Gropp et al., 2020), Point2Mesh (Hanocka et al., 2020), PSR (Kazhdan & Hoppe, 2013), SAP (Peng et al., 2021). We report numerical comparison in Tab. 5 and Tab. 6. Although we only do statistical reasoning on a single noisy point cloud and do not require point normals as SAP (Peng et al., 2021), our method still handles the noise in real s-canning well, which achieves much smoother and more accurate structure. The comparison in Fig. 10 and Fig. 14 shows that our method can produce more accurate surfaces without missing parts on both rigid and non-rigid shapes.

4.4. Surface Reconstruction for Scenes

3D Scene. We evaluate our method under real scene scan dataset (Zhou & Koltun, 2013). We sample 1000 points per m^2 from Lounge and Copyroom, and only leverage $N = 1$ noisy point cloud to train our method with a batch size of $B = 5000$. We leverage the pretrained models of COcc and

Metrics	IGR	Point2Mesh	PSR	SAP	**Ours**
L1CD×10	0.178	0.116	0.232	0.076	**0.067**
F-Score	0.755	0.648	0.735	0.830	**0.835**

Table 6. Comparison in surface reconstruction under SRB.

	Lounge			Copyroom		
	L2CD	L1CD	NC	L2CD	L1CD	NC
COcc (Peng et al., 2020)	9.540	0.046	0.894	10.97	0.045	0.892
LIG (Jiang et al., 2020a)	9.672	0.056	0.833	3.61	0.036	0.810
DeepLS (Chabra et al., 2020)	6.103	0.053	0.848	0.609	0.021	0.901
NP (Ma et al., 2021)	1.079	0.019	0.910	5.795	0.036	0.862
Ours	**0.602**	**0.016**	**0.923**	**0.442**	**0.016**	**0.903**

Table 7. Surface reconstruction under 3D Scene dataset. L2-CD×10^3. The unit of error is mm.

LIG and retrain NP and DeepLS to produce their results with the same input. We also provide LIG and DeepLS with the ground truth point normals. Numerical comparison in Tab. 7 demonstrates that our method significantly out-performs the state-of-the-art. Fig. 11 further demonstrates that we can produce much smoother surfaces with more geometry details.

Paris-rue-Madame. We further evaluate our method under another real scene scan dataset (Serna et al., 2014). We only use $N = 1$ noisy point cloud with a batch size of $B = 5000$. We split the $10M$ points into 50 chunks each of which is used to learn a SDF. Similarly, we use each chunk to evaluate IMLS (Liu et al., 2021) and LIG (Jiang et al., 2020a) with their pretrained models. Our superior performance over the latest methods in large scale surface reconstruction is demonstrated in Fig. 12. Our denoised point clouds in a smaller scene are detailed in Fig. 13.

B	100	250	1000	2000	5000	10000
L2CD×10^4	12.398	**4.221**	4.578	5.628	5.998	6.217
P2M×10^4	5.482	**1.847**	1.901	2.112	2.221	2.342

Table 8. Effect of batch size B under PU.

4.5. Ablation Studies

We conduct ablation studies under the test set of PU. We first explore the effect of batch size B, training iterations, and the number N of noisy point clouds in point cloud denoising. Tab. 16 indicates that more points in each batch will slow down the convergence. Tab. 9 demonstrates that more training iterations help perform statistical reasoning better to remove noise. Tab. 10 indicates that more corrupted observations are the key to increase the performance of statistical reasoning although one corrupted observation is also fine to perform statistical reasoning well.

We further highlight the effect of EMD as the distance met-ric L and geometric consistency regularization R in denois-ing and surface reconstruction in Tab. 17. The comparison shows that we can not perform statistical reasoning on point clouds using CD, and EMD can only reveal the surface in statistical reasoning for denoising but not learn meaning-ful signed distance fields without R. Moreover, we found the λ weighting R slightly affects our performance. More additional studies are in the following appendix.
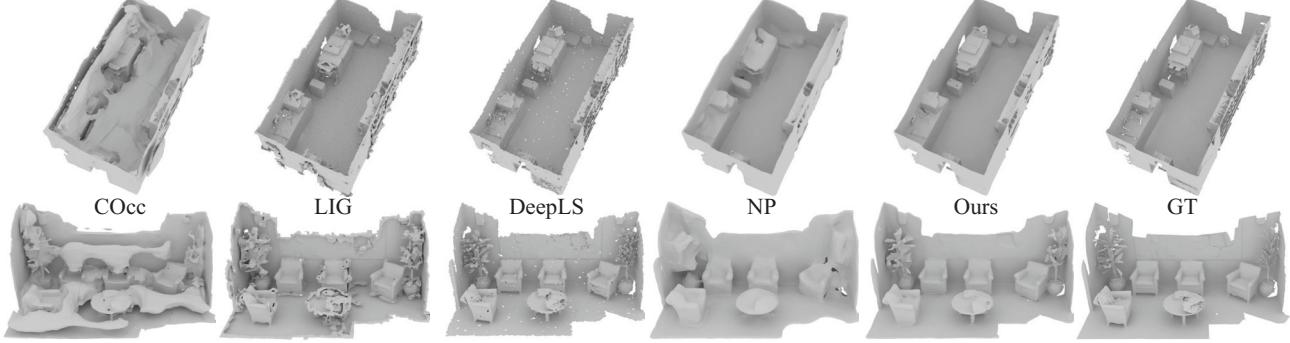
Figure 11. Visual comparison in surface reconstruction under 3D Scene dataset.

Figure 14. Comparison in surface reconstruction under SRB.

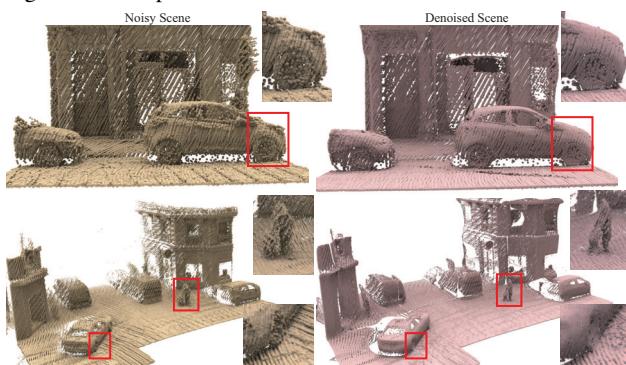*Figure 12.* Comparison in surface reconstruction from real scans.

	CD	EMD,$\lambda=0$	EMD,$\lambda=0.05$	EMD,$\lambda=0.1$	EMD,$\lambda=0.2$
Denoise	73.786	**4.221**	4.245	4.252	4.832
Reconstruction	81.573	80.917	5.721	**4.277**	4.993

Table 11. Effect of CD and EMD as the distance metric L and geometry consistency regularization R under PU. L2CD$\times 10^4$.

5. Conclusion

We introduce to learn SDFs from noisy point clouds via noise to noise mapping. We explore the feasibility of learning SDFs from multiple noisy point clouds or even one noisy point cloud without the ground truth signed distances, point normals or clean point clouds. Our noise to noise mapping enables the statistical reasoning on point clouds although there is no spatial correspondence among points on different noisy point clouds. Our key insight in noise to noise mapping is to use EMD as the metric in the statistical reasoning. With the capability of the statistical reasoning, we successfully reveal surfaces from noisy point clouds by learning highly accurate SDFs. We evaluate our method under synthetic dataset or real scanning dataset for both shapes or scenes. The effectiveness of our method is justified by our state-of-the-art performance in different applications.

Acknowledgement

Figure 13. Demonstration of denoising on real scans.

Iterations $\times 10^4$	40	60	80	100
L2CD$\times 10^4$	4.887	4.364	**4.221**	4.224
P2M$\times 10^4$	2.032	1.885	**1.847**	1.849

Table 9. Number of training iterations under PU.

N	1	2	10	20	50	100	200
L2CD$\times 10^4$	4.976	4.898	4.665	4.558	4.432	4.224	**4.221**
P2M$\times 10^4$	2.132	2.079	1.997	1.996	1.899	**1.847**	1.847

Table 10. Effect of N under PU.

References

Lp-dif: Learning local pattern-specific deep implicit function for 3d objects and scenes, author=Meng Wang and Yu-Shen Liu and Yue Gao and Kanle Shi and Yi Fang and Zhizhong Han, journal=IEEE International Conference on Computer Vision, year=2023.

Atzmon, M. and Lipman, Y. SAL: Sign agnostic learning of shapes from raw data. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Atzmon, M. and Lipman, y. SALD: sign agnostic learning with derivatives. In *International Conference on Learning Representations*, 2021.

Ben-Shabat, Y., Koneputugodage, C. H., and Gould, S. DiGS : Divergence guided shape implicit neural representation for unoriented point clouds. *CoRR*, abs/2106.10811, 2021.

Ben-Shabat, Y., Hewa Koneputugodage, C., and Gould, S. Digs: Divergence guided shape implicit neural representation for unoriented point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

Bogo, F., Romero, J., Pons-Moll, G., and Black, M. J. Dynamic FAUST: Registering human bodies in motion. In *IEEE Computer Vision and Pattern Recognition*, 2017.

Boulch, A. and Marlet, R. POCO: Point convolution for surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6302–6314, 2022.

Cai, R., Yang, G., Averbuch-Elor, H., Hao, Z., Belongie, S., Snavely, N., and Hariharan, B. Learning gradient fields for shape generation. In *European Conference on Computer Vision*, 2020.

Casajus, P. H., Ritschel, T., and Ropinski, T. Total denoising: Unsupervised learning of 3d point cloud cleaning. In *IEEE International Conference on Computer Vision*, pp. 52–60, 2019.

Cazals, F. and Pouget, M. Estimating differential quantities using polynomial fitting of osculating jets. *Computer Aided Geometry Design*, 22:121–146, 2005.

Chabra, R., Lenssen, J. E., Ilg, E., Schmidt, T., Straub, J., Lovegrove, S., and Newcombe, R. A. Deep local shapes: Learning local SDF priors for detailed 3D reconstruction. In *European Conference on Computer Vision*, volume 12374, pp. 608–625, 2020.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

Chen, C., Han, Z., Liu, Y.-S., and Zwicker, M. Unsupervised learning of fine structure generation for 3D point clouds by 2D projections matching. In *IEEE International Conference on Computer Vision*, 2021.

Chen, C., Liu, Y.-S., and Han, Z. Latent partition implicit with surface codes for 3d representation. In *European Conference on Computer Vision*, 2022.

Chen, C., Liu, Y.-S., and Han, Z. Unsupervised inference of signed distance functions from single sparse point clouds without learning priors. In *IEEE Conference on Computer Vsion and Pattern Recognition*, 2023.

Chen, Z. and Zhang, H. Learning implicit fields for generative shape modeling. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Chibane, J., Alldieck, T., and Pons-Moll, G. Implicit functions in feature space for 3d shape reconstruction and completion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6968–6979, 2020a.

Chibane, J., Mir, A., and Pons-Moll, G. Neural unsigned distance fields for implicit function learning. *arXiv*, 2010.13938, 2020b.

Choy, C. B., Xu, D., Gwak, J., Chen, K., and Savarese, S. 3D-r2n2: A unified approach for single and multiview 3d object reconstruction. In Leibe, B., Matas, J., Sebe, N., and Welling, M. (eds.), *European Conference on Computer Vision*, volume 9912, pp. 628–644, 2016.

Erler, P., Guerrero, P., Ohrhallinger, S., Mitra, N. J., and Wimmer, M. Points2Surf: Learning implicit surfaces from point clouds. In *European Conference on Computer Vision*, 2020.

Fan, H., Su, H., and Guibas, L. J. A point set generation network for 3D object reconstruction from a single image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2463–2471, 2017.

Feng, W., Li, J., Cai, H., Luo, X., and Zhang, J. Neural points: Point cloud representation with neural fields for arbitrary upsampling. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

Fleishman, S., Drori, I., and Cohen-Or, D. Bilateral mesh denoising. *ACM Transactions on Graphics*, 22(3):950–953, 2003.

Fu, Q., Xu, Q., Ong, Y., and Tao, W. Geo-Neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. 2022.

Genova, K., Cole, F., Vlasic, D., Sarna, A., Freeman, W. T., and Funkhouser, T. Learning shape templates with structured implicit functions. In *International Conference on Computer Vision*, 2019.

Genova, K., Cole, F., Sud, A., Sarna, A., and Funkhouser, T. Local deep implicit functions for 3d shape. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2020.

Gropp, A., Yariv, L., Haim, N., Atzmon, M., and Lipman, Y. Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3789–3799, 2020.

Groueix, T., Fisher, M., Kim, V. G., Russell, B. C., and Aubry, M. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 216–224, 2018.

Guo, H., Peng, S., Lin, H., Wang, Q., Zhang, G., Bao, H., and Zhou, X. Neural 3d scene reconstruction with the manhattan-world assumption. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

Gupta, A., Xiong, W., Nie, Y., Jones, I., and Ouz, B. 3dgen: Triplane latent diffusion for textured mesh generation. 2023.

Han, Z., Chen, C., Liu, Y.-S., and Zwicker, M. ShapeCaptioner: Generative caption network for 3D shapes by learning a mapping from parts detected in multiple views to sentences. In *ACM International Conference on Multimedia*, 2020a.

Han, Z., Chen, C., Liu, Y.-S., and Zwicker, M. DRWR: A differentiable renderer without rendering for unsupervised 3D structure learning from silhouette images. In *International Conference on Machine Learning*, 2020b.

Han, Z., Ma, B., Liu, Y.-S., and Zwicker, M. Reconstructing 3d shapes from multiple sketches using direct shape optimization. *IEEE Transactions on Image Processing*, 29:8721–8734, 2020c.

Han, Z., Qiao, G., Liu, Y.-S., and Zwicker, M. SeqXY2SeqZ: Structure learning for 3D shapes by sequentially predicting 1D occupancy segments from 2D coordinates. In *European Conference on Computer Vision*, 2020d.

Hanocka, R., Metzer, G., Giryes, R., and Cohen-Or, D. Point2mesh: a self-prior for deformable meshes. *ACM Transactions on Graphics*, 39(4):126, 2020.

Jia, M. and Kyan, M. Learning occupancy function from point clouds for surface reconstruction. *arXiv*, 2010.11378, 2020.

Jiang, C., Sud, A., Makadia, A., Huang, J., Nießner, M., and Funkhouser, T. Local implicit grid representations for 3D scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020a.

Jiang, Y., Ji, D., Han, Z., and Zwicker, M. SDFDiff: Differentiable rendering of signed distance fields for 3D shape optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020b.

Kazhdan, M. M. and Hoppe, H. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3): 29:1–29:13, 2013.

Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., and Aila, T. Noise2noise: Learning image restoration without clean data. In Dy, J. G. and Krause, A. (eds.), *International Conference on Machine Learning*, volume 80, pp. 2971–2980, 2018a.

Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., and Aila, T. Noise2noise: Learning image restoration without clean data. In *International Conference on Machine Learning*, volume 80, pp. 2971–2980, 2018b.

Li, Q., Liu, Y.-S., Cheng, J.-S., Wang, C., Fang, Y., and Han, Z. HSurf-Net: Normal estimation for 3D point clouds by learning hyper surfaces. 2022a.

Li, Q., Feng, H., Shi, K., Gao, Y., Fang, Y., Liu, Y.-S., and Han, Z. Shs-net: Learning signed hyper surfaces for oriented normal estimation of point clouds. In *IEEE International Conference on Computer Vision*, 2023a.

Li, S., Zhou, J., Ma, B., Liu, Y.-S., and Han, Z. Neaf: Learning neural angle fields for point normal estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023b.

Li, T., Wen, X., Liu, Y., Su, H., and Han, Z. Learning deep implicit functions for 3D shapes with dynamic code clouds. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12830–12840, 2022b.

Li, T., Wen, X., Liu, Y.-S., Su, H., and Han, Z. Learning deep implicit functions for 3D shapes with dynamic code clouds. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022c.

Lin, C.-H., Wang, C., and Lucey, S. SDF-SRN: Learning signed distance 3D object reconstruction from static images. In *Advances in Neural Information Processing Systems*, 2020.

Littwin, G. and Wolf, L. Deep meta functionals for shape representation. In *IEEE International Conference on Computer Vision*, 2019.

Liu, M., Zhang, X., and Su, H. Meshing point clouds with predicted intrinsic-extrinsic ratio guidance. In *European Conference on Computer vision*, 2020a.

Liu, S., Saito, S., Chen, W., and Li, H. Learning to infer implicit surfaces without 3D supervision. In *Advances in Neural Information Processing Systems*, 2019.

Liu, S., Zhang, Y., Peng, S., Shi, B., Pollefeys, M., and Cui, Z. DIST: Rendering deep implicit signed distance function with differentiable sphere tracing. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020b.

Liu, S.-L., Guo, H.-X., Pan, H., Wang, P., Tong, X., and Liu, Y. Deep implicit moving least-squares functions for 3D reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Lorensen, W. E. and Cline, H. E. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics*, 21(4):163–169, 1987.

Luo, S. and Hu, W. Differentiable manifold reconstruction for point cloud denoising. In *ACM International Conference on Multimedia*, pp. 1330–1338. ACM, 2020.

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

Ma, B., Han, Z., Liu, Y.-S., and Zwicker, M. Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. In *International Conference on Machine Learning*, 2021.

Ma, B., Liu, Y., and Han, Z. Reconstructing surfaces for sparse point clouds with on-surface priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6305–6315, 2022a.

Ma, B., Liu, Y., Zwicker, M., and Han, Z. Surface reconstruction from point clouds by learning predictive context priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6316–6327, 2022b.

Ma, B., Zhou, J., Liu, Y.-S., and Han, Z. Towards better gradient consistency for neural signed distance functions via level set alignment. In *IEEE Conference on Computer Vsion and Pattern Recognition*, 2023.

Martel, J. N. P., Lindell, D. B., Lin, C. Z., Chan, E. R., Monteiro, M., and Wetzstein, G. Acorn: adaptive coordinate networks for neural scene representation. *ACM Trans. Graph.*, 40(4):58:1–58:13, 2021.

Mattei, E. and Castrodad, A. Point cloud denoising via moving RPCA. *Computer Graphics Forum*, 36(8):123–137, 2017.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. Occupancy networks: Learning 3D reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Mi, Z., Luo, Y., and Tao, W. SSRNet: Scalable 3D surface reconstruction network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Michalkiewicz, M., Pontes, J. K., Jack, D., Baktashmotlagh, M., and Eriksson, A. P. Deep level sets: Implicit surface representations for 3D shape inference. *CoRR*, abs/1901.06802, 2019.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020.

Niemeyer, M., Mescheder, L., Oechsle, M., and Geiger, A. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Oechsle, M., Peng, S., and Geiger, A. UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *International Conference on Computer Vision*, 2021.

Ohtake, Y., Belyaev, A. G., Alexa, M., Turk, G., and Seidel, H. Multi-level partition of unity implicits. *ACM Transactions on Graphics*, 22(3):463–470, 2003.

Ouasfi, A. and Boukhayma, A. Few 'zero level set'-shot learning of shape signed distance functions in feature space. In *European Conference on Computer Vision*, 2022.

Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. DeepSDF: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Peng, S., Niemeyer, M., Mescheder, L. M., Pollefeys, M., and Geiger, A. Convolutional occupancy networks. In *European Conference on Computer Vision*, volume 12348, pp. 523–540, 2020.

Peng, S., Jiang, C. M., Liao, Y., Niemeyer, M., Pollefeys, M., and Geiger, A. Shape as points: A differentiable poisson solver. In *Advances in Neural Information Processing Systems*, 2021.

Pistilli, F., Fracastoro, G., Valsesia, D., and Magli, E. Learning graph-convolutional representations for point cloud denoising. In *European Conference on Computer Vision*, volume 12365, pp. 103–118, 2020.

Pumarola, A., Sanakoyeu, A., Yariv, L., Thabet, A., and Lipman, Y. Visco grids: Surface reconstruction with viscosity and coarea grids. In *Advances in Neural Information Processing Systems*, 2022.

Rakotosaona, M., Barbera, V. L., Guerrero, P., Mitra, N. J., and Ovsjanikov, M. Pointcleannet: Learning to denoise and remove outliers from dense point clouds. *Computer Graphics Forum*, 39(1):185–203, 2020.

Rematas, K., Martin-Brualla, R., and Ferrari, V. Sharf: Shape-conditioned radiance fields from a single view. In *International Conference on Machine Learning*, 2021.

Rosu, R. A. and Behnke, S. Permutosdf: Fast multi-view reconstruction with implicit surfaces using permutohedral lattices. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Saito, S., , Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., and Li, H. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. 2019.

Sayed, M., Gibson, J., Watson, J., Prisacariu, V., Firman, M., and Godard, C. Simplerecon: 3d reconstruction without 3d convolutions. In *European Conference on Computer Vision*, 2022.

Serna, A., Marcotegui, B., Goulette, F., and Deschaud, J. Paris-rue-madame database - A 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods. In *International Conference on Pattern Recognition Applications and Methods*, pp. 819–824, 2014.

Shue, J. R., Chan, E. R., Po, R., Ankner, Z., Wu, J., and Wetzstein, G. 3d neural field generation using triplane diffusion. In *IEEE International Conference on Computer Vision*, 2023.

Sitzmann, V., Zollhöfer, M., and Wetzstein, G. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, 2019.

Songyou Peng, Michael Niemeyer, L. M. M. P. A. G. Convolutional occupancy networks. In *European Conference on Computer Vision*, 2020.

Stier, N., Ranjan, A., Colburn, A., Yan, Y., Yang, L., Ma, F., and Angles, B. Finerecon: Depth-aware feed-forward network for detailed 3d reconstruction. 2023.

Takikawa, T., Litalien, J., Yin, K., Kreis, K., Loop, C., Nowrouzezahrai, D., Jacobson, A., McGuire, M., and Fidler, S. Neural geometric level of detail: Real-time rendering with implicit 3D shapes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Tang, J., Lei, J., Xu, D., Ma, F., Jia, K., and Zhang, L. SA-ConvONet: Sign-agnostic optimization of convolutional occupancy networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Tatarchenko, M., Richter, S. R., Ranftl, R., Li, Z., Koltun, V., and Brox, T. What do single-view 3D reconstruction networks learn? In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Stoll, C., and Theobalt, C. PatchNets: Patch-Based Generalizable Deep Implicit 3D Shape Representations. *European Conference on Computer Vision*, 2020.

Vicini, D., Speierer, S., and Jakob, W. Differentiable signed distance function rendering. *ACM Transactions on Graphics*, 41(4):125:1–125:18, 2022.

Wang, J., Wang, P., Long, X., Theobalt, C., Komura, T., Liu, L., and Wang, W. NeuRIS: Neural reconstruction of indoor scenes using normal priors. In *European Conference on Computer Vision*, 2022a.

Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., and Wang, W. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems*, pp. 27171–27183, 2021.

Wang, P.-S., Liu, Y., and Tong, X. Deep octree-based cnns with output-guided skip connections for 3d shape and scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 266–267, 2020.

Wang, W., Xu, Q., Ceylan, D., Mech, R., and Neumann, U. DISN: Deep implicit surface network for high-quality single-view 3D reconstruction. In *Advances In Neural Information Processing Systems*, 2019.

Wang, Y., Skorokhodov, I., and Wonka, P. HF-NeuS: Improved surface reconstruction using high-frequency details. 2022b.

Wen, X., Li, T., Han, Z., and Liu, Y.-S. Point cloud completion by skip-attention network with hierarchical folding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Wen, X., Xiang, P., Han, Z., Cao, Y.-P., Wan, P., Zheng, W., and Liu, Y.-S. Pmp-net: Point cloud completion by learning multi-step point moving paths. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Wen, X., Zhou, J., Liu, Y.-S., Su, H., Dong, Z., and Han, Z. 3D shape reconstruction from 2D images with disentangled attribute flow. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

Williams, F., Schneider, T., Silva, C., Zorin, D., Bruna, J., and Panozzo, D. Deep geometric prior for surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Wu, Y. and Sun, Z. DFR: differentiable function rendering for learning 3D generation from images. *Computer Graphics Forum*, 39(5):241–252, 2020.

Xiang, P., Wen, X., Liu, Y.-S., Cao, Y.-P., Wan, P., Zheng, W., and Han, Z. SnowflakeNet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *IEEE International Conference on Computer Vision*, 2021.

Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., and Lipman, Y. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33, 2020.

Yariv, L., Gu, J., Kasten, Y., and Lipman, Y. Volume rendering of neural implicit surfaces. In *Advances in Neural Information Processing Systems*, 2021.

Yifan, W., Wu, S., Oztireli, C., and Sorkine-Hornung, O. Iso-Points: Optimizing neural implicit surfaces with hybrid representations. *CoRR*, abs/2012.06434, 2020.

Yu, L., Li, X., Fu, C., Cohen-Or, D., and Heng, P. Pu-net: Point cloud upsampling network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2790–2799, 2018.

Yu, Z., Peng, S., Niemeyer, M., Sattler, T., and Geiger, A. MonoSDF: Exploring monocular geometric cues for neural implicit surface reconstruction. 2022.

Zakharov, S., Kehl, W., Bhargava, A., and Gaidon, A. Auto-labeling 3D objects with differentiable rendering of sdf shape priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Zeng, J., Cheung, G., Ng, M., Pang, J., and Yang, C. 3D point cloud denoising using graph laplacian regularization of a low dimensional manifold model. *IEEE Transactions on Image Processing*, 29:3474–3489, 2020.

Zhang, B., Tang, J., Nießner, M., and Wonka, P. 3d-shape2vecset: A 3d shape representation for neural fields and generative diffusion models. *CoRR*, abs/2301.11445, 2023a.

Zhang, W., Xing, R., Zeng, Y., Liu, Y.-S., Shi, K., and Han, Z. Fast learning radiance fields by shooting much fewer rays. *IEEE Transactions on Image Processing*, 32: 2703–2718, 2023b.

Zhao, W., Lei, J., Wen, Y., Zhang, J., and Jia, K. Sign-agnostic implicit learning of surface self-similarities for shape modeling and reconstruction from raw point clouds. *CoRR*, abs/2012.07498, 2020.

Zhou, J., Ma, B., Liu, Y.-S., Fang, Y., and Han, Z. Learning consistency-aware unsigned distance functions progressively from raw point clouds. In *Advances in Neural Information Processing Systems*, 2022a.

Zhou, J., Wen, X., Ma, B., Liu, Y.-S., Fang, Y., and Han, Z. Self-supervised point cloud representation learning with occlusion auto-encoder. *arXiv preprint arXiv:2203.14084*, 2022b.

Zhou, Q. and Koltun, V. Dense scene reconstruction with points of interest. *ACM Transactions on Graphics*, 32(4): 112:1–112:8, 2013.

14

A. Network Architectures

We employ a network that is modified based on Occ-Net (Mescheder et al., 2019). Since the output of OccNet is a value with a range of [0,1], we replace the sigmoid function that produces this output with the tanh function, which can output a signed distance value with a range of [-1,1], where the sign indicates the inside or outside of the 3D shape. In addition, we also replace the Resblock used in OccNet by simple fully connected layers to simplify the OccNet, which highlights the advantage of our method.

B. Query Sampling

We sample more queries around a noisy point cloud if there is only one noisy point cloud available. We leverage a method introduced by NeuralPull (Ma et al., 2021) to sample queries around each point on the noisy point cloud.

C. Surface Reconstruction

Numerical Comparison. We report more detailed comparison under ShapeNet (Chang et al., 2015). Due to the text limit in the main body, we only report the mean metric over all 13 classes under ShapeNet. We compare our methods with methods including PSR (Kazhdan & Hoppe, 2013), PSG (Fan et al., 2017), R2N2 (Choy et al., 2016), Atlas (Groueix et al., 2018), COcc (Peng et al., 2020), S-AP (Peng et al., 2021), OCNN (Wang et al., 2020), and IMLS (Liu et al., 2021). We report the numerical comparison in terms of L1CD, NC, and F-score in Tab. 12, Tab. 13, and Tab. 14, respectively.

Visual Comparison. We report more surface reconstruction results under ShapeNet (Chang et al., 2015) in Fig. 15, Fig. 16 and Fig. 17. This comparison demonstrates that our method can reconstruct more geometry details than the state-of-the-art methods.

We also highlight our performance on point denoising and surface reconstructions on a large scale real scan in our video.

D. Point Cloud Denoising

Additionally, we visualize our results with larger noises which we use to learn an SDF in point cloud denoising in Fig. 19. We tried noises with different variances including $\{2\%, 4\%, 6\%, 8\%, 10\%\}$. We can see that our method can reveal accurate geometry with large noises. While our method may fail if the noises are too large to observe the structures, such as the variance of 10 percent. Note that variances larger than 3 percent are not widely used in evaluations in previous studies.

Figure 15. Visual comparison with COcc (Peng et al., 2020) and IMLS (Liu et al., 2021) in surface reconstruction under ShapeNet.

E. Results on KITTI

Additionally, we report our reconstruction on a road from KITTI in Fig. 18. Our method can also reconstruct plausible and smooth surfaces from a single real scan containing sparse and noisy points, please see our reconstruction

F. Computational Complexity

We report our computational complexity in the following table. We report numerical comparisons with the latest overfitting based methods including NeuralPull (N-P) and PCP using different point numbers including $\{20K, 40K, 80K, 160K\}$ in Tab. 15, where all methods search the nearest neighbors for queries online. NerualPull does not use learned priors while PCP uses learned priors parameterized by a neural network, both of which require

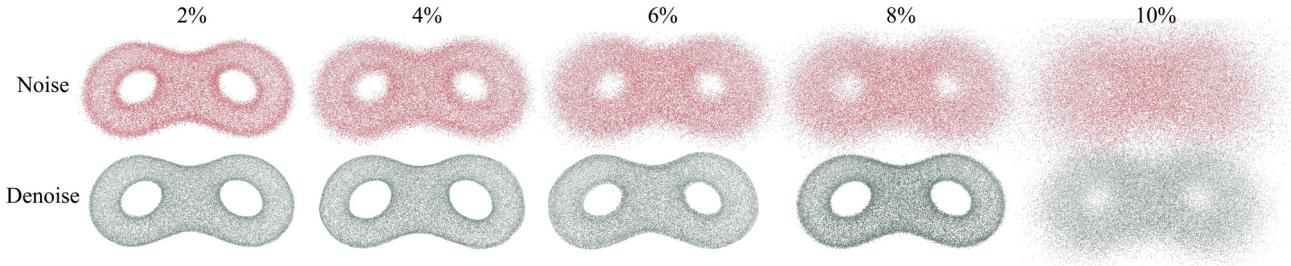|	PSR	PSG	R2N2	Atlas	COcc	SAP	OCNN	IMLS	POCO	**Ours**
airplane	0.437	0.102	0.151	0.064	0.034	0.027	0.063	0.025	0.023	**0.022**
bench	0.544	0.128	0.153	0.073	0.035	0.032	0.065	0.030	0.028	**0.025**
cabinet	0.154	0.164	0.167	0.112	0.047	0.037	0.071	0.035	0.037	**0.034**
car	0.180	0.132	0.197	0.099	0.075	0.045	0.077	0.040	0.041	**0.037**
chair	0.369	0.168	0.181	0.114	0.046	0.036	0.066	0.035	0.033	**0.026**
display	0.280	0.160	0.170	0.089	0.036	0.030	0.066	0.029	0.028	**0.022**
lamp	0.278	0.207	0.243	0.137	0.059	0.047	0.067	0.031	0.033	**0.027**
speaker	0.148	0.205	0.199	0.142	0.063	0.041	0.073	0.040	0.041	**0.033**
rifle	0.409	0.091	0.167	0.051	0.028	0.023	0.062	0.021	0.019	**0.019**
sofa	0.227	0.144	0.160	0.091	0.041	0.032	0.066	0.031	0.030	**0.027**
table	0.393	0.166	0.177	0.102	0.038	0.033	0.066	0.032	0.031	**0.028**
telephone	0.281	0.110	0.130	0.054	0.027	0.023	0.061	0.023	0.022	**0.017**
vessele	0.181	0.130	0.169	0.078	0.043	0.030	0.064	0.027	0.025	**0.024**
mean	0.299	0.147	0.173	0.093	0.044	0.034	0.067	0.031	0.030	**0.026**

Table 12. L1CD×10 comparison under ShapeNet.

Figure 19. Point clouds denoising with large noises.

the nearest neighbor search as ours. We report the time used to train these methods in 50K iterations. The comparisons indicate that our method uses less storage and less time than its counterparts.

Table 15. Comparison of Computational Complexity.

Time/GPU Memory	20K	40K	80K	160K
NP	12min/**1.5G**	15min/2.3G	19min/4.1G	33min/**8.0G**
PCP	14min/1.9G	18min/2.7G	22min/4.6G	35min/8.4G
Ours	**10**min/**1.5G**	**12**min/**2.2G**	**15**min/**4.0G**	**21**min/**8.0G**

Since NP and PCP can not handle noises well, their reconstructions contain severe artifacts on the surface. While our method can handle that well. Please see more numerical comparisons with these methods in our paper. In addition, our results may get more improvements if we train our method more iterations.

G. Ablation Studies

Number of Noisy Point Clouds. We report additional ablation studies to explore the effect of the number of noisy point clouds in all the three tasks including point cloud denoising, point cloud upsampling, and surface reconstruction under the PU test set below. We can see we achieve the best performance with 200 noisy point clouds in all tasks, and the improvement over 100 point clouds is small. So we used 200 to report our results with multiple noisy point clouds in our paper.

Point Density. We report the effect of point density in all the three tasks including point cloud denoising, point cloud upsampling, and surface reconstruction under the PU test set below. We learn an SDF from a single noisy point cloud. With more noises, our method can achieve better performance in all the three tasks.

One Observation vs. Multiple Observations. Since our

	PSR	PSG	R2N2	Atlas	COcc	SAP	OCNN	IMLS	POCO	**Ours**
airplane	0.747	-	0.669	0.854	0.931	0.931	0.918	0.937	0.944	**0.960**
bench	0.649	-	0.691	0.820	0.921	0.920	0.914	0.922	0.928	**0.935**
cabinet	0.835	-	0.786	0.875	0.956	0.957	0.941	0.955	0.961	**0.975**
car	0.783	-	0.719	0.827	0.893	0.897	0.867	0.882	0.894	**0.937**
chair	0.715	-	0.673	0.829	0.943	0.952	0.941	0.950	0.956	**0.965**
display	0.749	-	0.747	0.905	0.968	0.972	0.960	0.973	0.975	**0.981**
lamp	0.765	-	0.598	0.759	0.900	0.921	0.911	0.922	0.929	**0.957**
speaker	0.843	-	0.735	0.867	0.938	0.950	0.936	0.947	0.952	**0.977**
rifle	0.788	-	0.700	0.837	0.929	0.937	0.932	0.943	**0.949**	0.938
sofa	0.826	-	0.754	0.888	0.958	0.963	0.949	0.963	0.967	**0.978**
table	0.706	-	0.734	0.867	0.959	0.962	0.946	0.962	0.966	**0.970**
telephone	0.805	-	0.847	0.957	0.983	0.984	0.974	0.984	0.985	**0.987**
vessele	0.820	-	0.641	0.837	0.918	0.930	0.922	0.932	0.940	**0.951**
mean	0.772	-	0.715	0.855	0.938	0.944	0.932	0.944	0.950	**0.962**

Table 13. NC comparison under ShapeNet.

Table 18. Effect of mixing multiple noise point clouds under PU.

Strategy	Metric	Mixing	W/O Mixing
Denoise	L2CD$\times 10^4$	4.244	**4.221**
	P2M$\times 10^4$	1.851	**1.847**
Reconstruction	L2CD$\times 10^4$	**4.315**	4.355
	P2M$\times 10^4$	**1.831**	1.877
UpSampling	L2CD$\times 10^4$	4.299	**4.272**
	P2M$\times 10^4$	**1.897**	1.897

method can learn from multiple observations and single observation, we investigate the effect of learning from these two training settings. Here, we combine multiple noisy observations into one noisy observation by concatenation, where we keep the total number of points the same. Table. 18 indicates that there is almost no performance difference with these two training settings. The reason i

H. Optimization Visualization

We visualize the optimization process in our video. We visualize the noisy points matched by EMD for each query in each epoch. In addition, we also visualize the denoised points using the gradient in the learned SDF in different epochs.

I. Proof

We proof Theorem 1 in our submission in the following.

Theorem 1. *Assume there was a clean point cloud* $\boldsymbol{G}$ *which*

is corrupted into observations $S = \{\boldsymbol{N}_i\}$ *by sampling a noise around each point of* $\boldsymbol{G}$. *If we leverage EMD as the distance metric L defined in Eq. (8), and learn a point cloud* $\boldsymbol{G}'$ *by minimizing the EMD between* $\boldsymbol{G}'$ *and each observation in S, i.e.,* $\min_{\boldsymbol{G}'} \sum_{\boldsymbol{N}_i \in S} L(\boldsymbol{G}', \boldsymbol{N}_i)$, *then* $\boldsymbol{G}'$ *converges to the clean point cloud* $\boldsymbol{G}$, *i.e.,* $L(\boldsymbol{G}, \boldsymbol{G}') = 0$.

$$L(\boldsymbol{G}, \boldsymbol{G}') = \min_{\phi:\boldsymbol{G}\rightarrow\boldsymbol{G}'} \sum_{\boldsymbol{g}\in\boldsymbol{G}} ||\boldsymbol{g} - \phi(\boldsymbol{g})||_2, \quad (8)$$

where ϕ is a one-to-one mapping.

Proof: Suppose each corrupted observation $\boldsymbol{N}_i$ in the set $S = \{\boldsymbol{N}_i | i \in [1, N]\}$ is formed by m points, and $\boldsymbol{N}_i = \{n_i^k | k \in [1, m], m \geq 1\}$. With the same assumption, either $\boldsymbol{G}$ or $\boldsymbol{G}'$ is also formed by m points, $\boldsymbol{G} = \{g^k | k \in [1, m], m \geq 1\}$, $\boldsymbol{G}' = \{g'^k | k \in [1, m], m \geq 1\}$. Assuming each noise n_i^k is corrupted from the clean g^k, we leverage this assumption to justify the correctness of our proof. $L(\boldsymbol{G}', S) = \sum_{\boldsymbol{N}_i \in S} L(\boldsymbol{G}', \boldsymbol{N}_i)$.

(a) When $m = 1$, this is similar to Noise2Noise (Lehtinen et al., 2018b),

$$L(\boldsymbol{G}', S) = \sum_{i=1}^{N} (g'^1 - n_i^1)^2.$$

$$\frac{\partial L(\boldsymbol{G}', S)}{\partial \boldsymbol{G}'} = 2\sum_{i=1}^{N} (g'^1 - n_i^1). \quad (9)$$

$$\frac{\partial L(\boldsymbol{G}', S)}{\partial \boldsymbol{G}'} = 0 \rightarrow g'^1 = 1/N \sum_{i=1}^{N} n_i^1.$$

	PSR	PSG	R2N2	Atlas	COcc	SAP	OCNN	IMLS	POCO	**Ours**
airplane	0.551	0.476	0.382	0.827	0.965	0.981	0.810	0.992	0.994	**0.995**
bench	0.430	0.266	0.431	0.786	0.965	0.979	0.800	0.986	0.988	**0.993**
cabinet	0.728	0.137	0.412	0.603	0.955	0.975	0.789	0.981	0.979	**0.996**
car	0.729	0.211	0.348	0.642	0.849	0.928	0.747	0.952	0.946	**0.964**
chair	0.473	0.152	0.393	0.629	0.939	0.979	0.799	0.982	0.985	**0.993**
display	0.544	0.175	0.401	0.727	0.971	0.990	0.811	0.994	0.994	**0.998**
lamp	0.586	0.204	0.333	0.562	0.892	0.959	0.800	0.979	0.975	**0.990**
speaker	0.731	0.107	0.405	0.516	0.892	0.957	0.779	0.963	0.964	**0.977**
rifle	0.590	0.615	0.381	0.877	0.980	0.990	0.826	0.996	0.998	**0.998**
sofa	0.712	0.184	0.427	0.717	0.953	0.982	0.801	0.987	0.989	**0.992**
table	0.442	0.158	0.404	0.692	0.967	0.986	0.801	0.987	0.991	**0.992**
telephone	0.674	0.317	0.484	0.867	0.989	0.997	0.825	0.998	0.998	**0.999**
vessele	0.771	0.363	0.394	0.7757	0.931	0.974	0.809	0.987	0.989	**0.997**
mean	0.612	0.259	0.400	0.708	0.942	0.975	0.800	0.983	0.984	**0.991**

Table 14. F-Score comparison under ShapeNet.

Table 16. Effect of batch size B under PU.

B	Metric	1	2	10	20	50	100	200
Denoise	L2CD$\times10^4$	4.976	4.898	4.665	4.558	4.432	4.224	**4.221**
	P2M$\times10^4$	2.132	2.079	1.997	1.996	1.899	**1.847**	**1.847**
Reconstruction	L2CD$\times10^4$	5.102	4.995	4.795	4.599	4.456	4.369	**4.355**
	P2M$\times10^4$	2.423	2.217	2.007	2.001	1.978	1.886	**1.877**
UpSampling	L2CD$\times10^4$	4.988	4.886	4.687	4.574	4.461	4.328	**4.272**
	P2M$\times10^4$	2.152	2.082	2.001	1.997	1.977	1.919	**1.897**

Since $S = \{N_i\}$ is a set corrupted from the clean point cloud G, $g^1 = 1/N \sum_{i=1}^{N} n_i^1$. Furthermore, we also get $g'^1 = g^1$.

From Eq. (9), we can also get the following conclusion,

$$\min_{G'} L(G', S) \leftrightarrow G' = \mathbb{E}(\phi(G')), \qquad (10)$$

where $\phi = \{\phi_i | i \in [1, N]\}$ is a set of one-to-one mapping ϕ_i which maps G' to each corrupted observation N_i in S.

(b) When $m \geq 2$, assuming that we know which noisy point n_i^k on each point cloud N_i is corrupted from the clean point g^k. We regard the correspondence c_i between $\{n_i^k | i \in [1, N]\}$ and g^k as the ground truth, so that we can verify the correctness of our following proof. Note that we did not use this assumption in the proof process. So, we can represent the correspondence using the following equation,

$$\mathbb{E}(n(k)) = 1/N \sum_{i=1}^{N} n_i^k = g^k, \qquad (11)$$

where $n(k) = \{n_i^k | i \in [1, N]\}$.

As defined before, ϕ_i is the one-to-one mapping established in the calculation of EMD between G' and N_i. Therefore, the distance between G' and noisy point cloud set S is, $L(G', S) = \sum_{k=1}^{m} (\sum_{i=1}^{N} ((g'^k - \phi_i(g'^k))^2))$,

There are two cases. One is that the one-to-one mapping ϕ_i is exactly the correspondence ground truth c_i. The other is that ϕ_i is not the correspondence ground truth.

Case (1): When $\phi_i(g'^k) = n_i^k$, $i \in [1, N]$, this is consistent with (a), so the Theorem 1 gets proved.

Case (2): When $\phi_i(g'^k) \neq n_i^k$, assuming $\phi_i(g'^k) = n_i^{a_{k,i}}$, $A_k = \{n_i^{a_{k,i}} | i \in [1, N]\}$, A_k is a set corresponding to g'^k. When minimizing $L(G', S) = \sum_{k=1}^{m} \sum_{i=1}^{N} (g'^k - \phi_i(g'^k))^2$, according to Eq. (10), $g'^k = \mathbb{E}(\phi_i(g'^k))$, so $Var(A_k) = 1/N \sum_{i=1}^{N} ((g'^k - \mathbb{E}(\phi_i(g'^k)))^2)$. When $m = 2$, $\min_{G'} L(G', S) = \min(Var(A_1) + Var(A_2))$. We assume $A_1 = n_s^1 + n_{cs}^2$ to simply the following proof, where s is a subset of set $[1, N]$, cs is the complement of set s, so $A_2 = n_s^2 + n_{cs}^1$. Assuming $\mathbb{E}(A_1) = g^1 + \Delta$, Δ is the point offset of g^1, because of $\mathbb{E}(A_1) + E(A_2) = g^1 + g^2$, so $\mathbb{E}(A_2) = g^2 - \Delta$,

18

Table 17. Effect of Density D of point Cloud under PU.

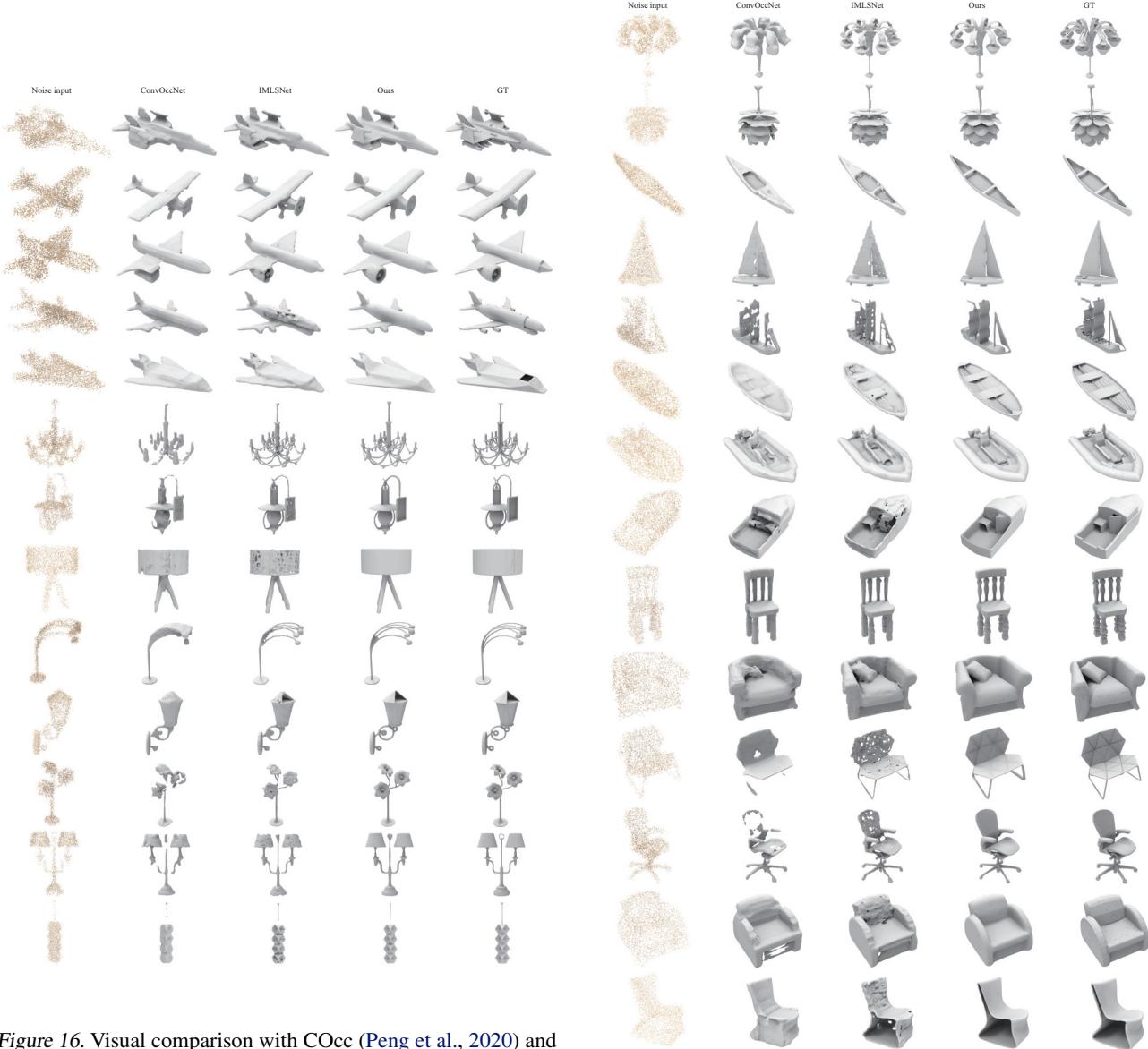| D	Metric	1K	2K	5K	10K	20K	50K	100K
Denoise	L2CD$\times10^4$	5.168	5.098	4.850	4.221	2.312	1.654	**1.543**
	P2M$\times10^4$	2.223	2.179	2.097	1.847	1.229	0.972	**0.959**
Reconstruction	L2CD$\times10^4$	5.445	5.283	4.981	4.355	2.388	1.691	**1.579**
	P2M$\times10^4$	2.330	2.212	2.159	1.877	1.292	0.998	**0.982**
UpSampling	L2CD$\times10^4$	5.281	5.187	4.984	4.272	2.392	1.682	**1.561**
	P2M$\times10^4$	2.398	2.212	2.167	1.897	1.289	0.997	**0.973**

Figure 16. Visual comparison with COcc (Peng et al., 2020) and IMLS (Liu et al., 2021) in surface reconstruction under ShapeNet.

Figure 17. Visual comparison with COcc (Peng et al., 2020) and IMLS (Liu et al., 2021) in surface reconstruction under ShapeNet.
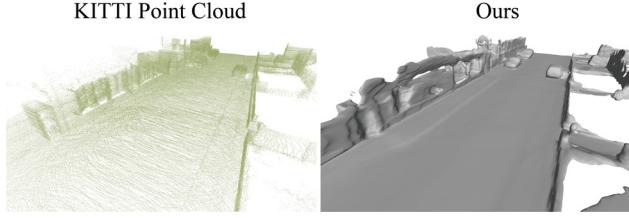
KITTI Point Cloud   Ours

Figure 18. Reconstruction on a real scan from KITTI.

Because the first two terms of the formula are constants, the entire formula becomes a quadratic formula, so when $\Delta = 0$ or $\Delta = g^2 - g^1$, the value of $L(\boldsymbol{G'}, S)$ is minimized. $\Delta = 0$ is consistent with Case (1). $\Delta = g^2 - g^1$, $\phi_i(g^1) = n_i^2$, $\phi_i(g^2) = n_i^1$, this is also the same correspondence as the ground truth, so Theorem 1 gets proved. When $m > 2$. We can extend the proof from the two sets A_1 and A_2 to multiple sets $A_1, A_2, \cdots, A_m$, and the proof process is similar to the above.

$$
\begin{aligned}
&L(\boldsymbol{G'}, S)\\
&= (Var(A_1) + Var(A_2))\\
&= \mathbb{E}(A_1 - (g^1 + \Delta))^2 + \mathbb{E}(A_2 - (g^2 - \Delta))^2\\
&= 1/N (\sum_{i=1}^{N}(n_i^{a_{1,i}})^2 + \sum_{i=1}^{N}(n_i^{a_{2,i}})^2 + N(g^1 + \Delta)^2\\
&\quad + N(g^2 - \Delta)^2 - 2\sum_{i=1}^{N} n_i^{a_{1,i}}(g^1 + \Delta)\\
&\quad - 2\sum_{i=1}^{N} n_i^{a_{2,i}}(g^2 - \Delta))\\
&= \mathbb{E}((n(1))^2) + \mathbb{E}((n(2))^2) + \mathbb{E}^2(n(1)) +\\
&\quad \mathbb{E}^2(n(2)) + 2\Delta^2 + 2g^1\Delta - 2g^2\Delta -\\
&\quad 2/N(g^1\sum_{i=1}^{N}n_i^{a_{1,i}} + g^2\sum_{i=1}^{N}n_i^{a_{2,i}} +\\
&\quad \Delta\sum_{i=1}^{N}n_i^{a_{1,i}} - \Delta\sum_{i=1}^{N}n_i^{a_{2,i}})\\
&= \mathbb{E}((n(1))^2) + \mathbb{E}((n(2))^2) + \mathbb{E}^2(n(1)) +\\
&\quad \mathbb{E}^2(n(2)) + 2/N(\Delta(n_s^1 + n_{cs}^1) - \Delta(n_s^2 + n_{cs}^2)\\
&\quad - \Delta(n_s 1 + n_{cs}^2) + \Delta(n_{cs}^1 + n_s^2) -\\
&\quad g^1\sum_{i=1}^{N}n_i^{a_{1,i}} - g^2\sum_{i=1}^{N}n_i^{a_{2,i}})\\
&= \mathbb{E}((n(1))^2) + \mathbb{E}((n(2))^2) + \mathbb{E}^2(n(1)) +\\
&\quad \mathbb{E}^2(n(2)) + 2\Delta^2 + 2\Delta/N(2n_{cs}^1 -\\
&\quad 2n_{cs}^2) - 2/N(g^1 N(g^1 + \Delta) + g^2 N(g^2 - \Delta))\\
&= \mathbb{E}((n(1))^2) + \mathbb{E}((n(2))^2) - \mathbb{E}^2(n(1)) -\\
&\quad \mathbb{E}^2(n(2)) + 2\Delta^2 +\\
&\quad 2\Delta/N(2n_{cs}^1 - 2n_{cs}^2 - n_{cs}^1 - n_s^1 + n_{cs}^2 + n_s^2)\\
&= \mathbb{E}((n(1))^2) + \mathbb{E}((n(2))^2) - \mathbb{E}^2(n(1)) -\\
&\quad E^2(n(2)) + 2\Delta(g^2 - g^1) - 2\Delta^2\\
&= Var(n(1)) + Var(n(2)) + 2\Delta(g^2 - g^1) - 2\Delta^2
\end{aligned}
$$

$$(12)$$