Fast Learning of Signed Distance Functions From Noisy Point Clouds via Noise to Noise Mapping

Junsheng Zhou, Baorui Ma, Yu-Shen Liu, *Member, IEEE*, and Zhizhong Han

***Abstract*—Learning signed distance functions (SDFs) from 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. 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 observations. 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. To accelerate training, we use multi-resolution hash encodings implemented in CUDA in our framework, which reduces our training time by a factor of ten, achieving convergence within one minute. We further introduce a novel schema to improve multi-view reconstruction by estimating SDFs as a prior. Our evaluations under widely-used benchmarks demonstrate our superiority over the state-of-the-art methods in surface reconstruction from point clouds or multi-view images, point cloud denoising and upsampling.**

***Index Terms*—Surface reconstruction, noise to noise mapping, signed distance functions, point cloud denoising, fast learning.**

I. 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 [1], [2], [3], [4], [5]. Different from classic methods [6], [7], 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 [8], point normals [4], [5], [9], additional constraints [1], [2] or no noise assumption [3]. These requirements significantly affect the accuracy of SDFs learned for noisy point clouds, either caused by poor 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, 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 when there is no spatial correspondence among points on different corrupted observations.

We originally presented our method at ICML2023 [10], and then extend our method in more applications with novel frameworks. Specifically, we get inspiration from Instant-NGP [11] and propose a fast learning framework to leverage multi-resolution hash tables of learnable features. This design improves the efficiency of SDF inference in our noise to noise mapping process, and reduces our training time from 15 minutes to one minute.

We further extend our method in the multi-view setting. Recent methods [12], [13] learn SDFs in multi-view reconstruction by minimizing the volume rendering error introduced in NeRF [14] which learns radiance fields for scene representation. Using the SDF estimated by our method as a prior, we improve

Manuscript received 17 August 2023; revised 29 April 2024; accepted 5 June 2024. Date of publication 18 June 2024; date of current version 5 November 2024. This work was supported in part by the National Key R&D Program of China under Grant 2022YFC3800600, and in part by the National Natural Science Foundation of China under Grant 62272263 and under Grant 62072268. Project page: https://mabaorui.github.io/Noise2NoiseMapping/. Recommended for acceptance by P. Mordohai. *(Junsheng Zhou and Baorui Ma are contributed equally to this work.) (Corresponding author: Yu-Shen Liu.)*

Junsheng Zhou is with the School of Software, Tsinghua University, Beijing 100190, China (e-mail: zhoujs21@mails.tsinghua.edu.cn).

Baorui Ma is with the School of Software, Tsinghua University, Beijing 100190, China, and also with the BAAI, Beijing 100084, China (e-mail: brma@baai.ac.cn).

Yu-Shen Liu is with the School of Software, Tsinghua University, Beijing 100190, China (e-mail: liuyushen@tsinghua.edu.cn).

Zhizhong Han is with the Department of Computer Science, Wayne State University, Detroit, MI 48202 USA (e-mail: h312h@wayne.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2024.3416068, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2024.3416068

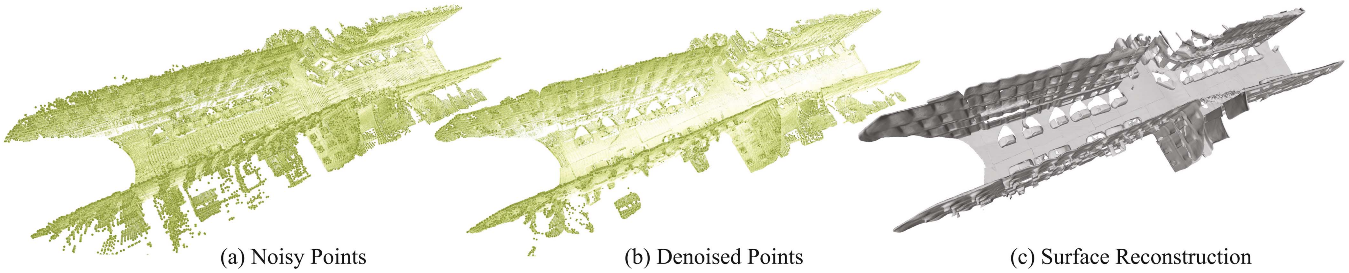

Fig. 1. We introduce to learn signed distance functions (SDFs) for single noisy point clouds. Our method does not require ground truth signed distances, point normals or clean points as supervision for training. We achieve this via learning a mapping from one noisy observation to another or even on a 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 10 M points, (b) the denoised point cloud and (c) the reconstructed surface. Fig. 17 demonstrates our superiority over the latest surface reconstructions in this case.

the multi-view reconstruction accuracy by removing artifacts in empty space. Specifically, we first leverage Structure from Motion (SfM) methods (e.g., COLMAP [15]) to estimate a noisy point cloud from multi-view images and learn a signed distance field from it using our fast learning framework. The learned field is then adopted as a prior to guide multi-view reconstruction methods (e.g., NeuS [12]) for reconstructing surfaces from multi-view images. The key reason that our method can serve as the guidance for multi-view reconstruction is that our method learns clean and accurate SDFs from noisy point clouds obtained by SfM.

Our results achieve the state-of-the-art in different applications including surface reconstruction from point clouds and multi-view images, point cloud denoising and upsampling under widely used benchmarks. Our contributions are listed below.

1) We introduce a method to learn SDFs from noisy point clouds without requiring ground truth signed distances, point normals or clean point clouds.
2) 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.
3) We significantly accelerate the training process to enable the convergence of SDFs within one minute by integrating multi-resolution hash encoding into our framework.
4) We propose a novel schema to use the SDF learned from noisy SfM points as a prior for improving multi-view reconstruction.
5) We achieved the state-of-the-art results in surface reconstruction from point clouds or multi-view images, point cloud denoising and upsampling for shapes or scenes under widely used benchmarks.

II. Related Work

Learning implicit functions for 3D shapes and scenes has made great progress [14], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29]. 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 [30], [31], [32], [33], [34], [35], [36] and binary occupancy labels [37], [38]. With a condition, such as a single image [39], [40], [41], [42] or a learnable latent code [31], neural networks can be trained as an implicit function to model various shapes. We can also leverage point clouds as conditions [41], [43], [44], [45] to learn implicit functions, and then leverage the marching cubes algorithm [46] to reconstruct surfaces [47], [48]. To capture more detailed geometry, implicit functions are defined in local regions which are covered by voxel grids [4], [8], [9], [21], [22], [49], [50], patches [51], 3D Gaussian functions [52], learnable codes [53], [54].

Learning from Multi-view Images: Some recent works learn implicit functions from 2D supervision, such as multiple images. The basic idea is to leverage various differentiable renderers [55], [56], [57], [58], [59], [60] 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 [12], [13], [14], [61], [62], [63], [64], [65]. By sampling rays emitted from pixels into the field, UNISURF [16] and NeuS [12] employ a modified rendering procedure to transform occupancy and signed distance fields, along with radiance, into pixel colors. Subsequent methods enhance the accuracy of implicit functions by leveraging multi-view consistency [62], [66] and incorporating additional priors such as depth [63] and normals [67], [68].

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 [1], [2], [69], [70], [71], [72], [73], [74], [75], [76], [77], [78], gradients [3], [79], differentiable poisson solver [5] or specially designed priors [80], [81] to learn signed [1], [2], [3], [19], [70], [71], [82], [83] or unsigned distance fields [79], [84], [85], [86]. 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.

Deep Learning Based Point Cloud Denoising: PointCleanNet [87] was introduced to remove outliers and reduce noise from point clouds using a data-driven strategy. Graph

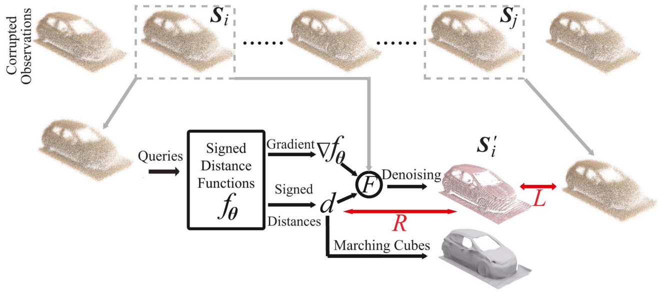

Fig. 2. Given corrupted observations captured by a Lidar system per second, we learn a SDF without supervision or normals.

convolution was also leveraged to reduce the noise based on dynamically constructed neighborhood graphs [88]. Without supervision, TotalDenoising [89] inherits the same idea as Noise2Noise [90]. It leveraged a spatial prior term that can work for unordered point clouds. More recently, downsample-upsample architecture [91] and gradient fields [92], [93] were leveraged to reduce noise. We were inspired by the idea of Noise2Noise [90], our contribution lies in our finding that we can still leverage statistical reasoning among multiple noisy point clouds with specially designed losses even when there is no spatial correspondence among points on different observations like the one among pixels, which is totally different from TotalDenoising [89].

III. Method

Overview: Given N corrupted observations $S = \{\boldsymbol{S}_i | i \in [1, N], N \geq 1\}$ of an uncorrupted 3D shape or scene $\boldsymbol{Z}$, we aim to learn SDFs f of $\boldsymbol{Z}$ from S without ground truth signed distances, point normals, or clean point clouds. Here, $\boldsymbol{S}_i$ is a noisy point cloud. The signed distance function f predicts a signed distance d for an arbitrary query location $\boldsymbol{q} \in \mathbb{R}^{1\times 3}$ around $\boldsymbol{Z}$, such that $d = f(\boldsymbol{q})$. We train a neural network with parameters $\boldsymbol{\theta}$ to learn f, which we denote as $f_{\boldsymbol{\theta}}$. After training, we can leverage the learned $f_{\boldsymbol{\theta}}$ 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 $\boldsymbol{S}_i$ as input, our network aims to learn SDFs $f_{\boldsymbol{\theta}}$ via learning a noise to noise mapping from $\boldsymbol{S}_i$ to another noisy point cloud $\boldsymbol{S}_j$, where $\boldsymbol{S}_j$ 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 $\boldsymbol{S}_i'$ and $\boldsymbol{S}_j$ using a metric L but also constrains the learned SDFs $f_{\boldsymbol{\theta}}$ to be correct using a geometric consistency regularization R. A denoising function F conducts point cloud denoising using signed distances d and gradients $\nabla f_{\boldsymbol{\theta}}$ from $f_{\boldsymbol{\theta}}$.

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 $\boldsymbol{S}'$ from its corrupted observations $\boldsymbol{S} \in S$ below,

$$\underset{\boldsymbol{S}'}{\text{argmin}}\, \mathbb{E}_{\boldsymbol{S}}\{L(\boldsymbol{S}', \boldsymbol{S})\}. \tag{1}$$

As a conclusion of Noise2Noise [90] for 2D image denoising, we can learn a denoising function F by pushing a denoised image $F(\boldsymbol{x})$ to be similar to as many corrupted observations $\boldsymbol{y}$ as possible, where both $\boldsymbol{x}$ and $\boldsymbol{y}$ 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 (1) into an equation with a denoising function F,

$$\underset{F}{\text{argmin}} \sum_{\boldsymbol{S}_i \in S} \sum_{\boldsymbol{S}_j \in S} L(F(\boldsymbol{S}_i), \boldsymbol{S}_j). \tag{2}$$

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 [89] does not think (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 (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 $f_{\boldsymbol{\theta}}$ via point cloud denoising in (2). Our solution is to leverage $f_{\boldsymbol{\theta}}$ 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 $\boldsymbol{S}'$ from a noisy point cloud $\boldsymbol{S}$, so $\boldsymbol{S}' = F(\boldsymbol{S})$.

To learn SDFs $f_{\boldsymbol{\theta}}$ of $\boldsymbol{S}$, we want the denoising procedure can also perceive the signed distance fields around $\boldsymbol{S}$. The 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 $\boldsymbol{S}$, we could involve the SDFs $f_{\boldsymbol{\theta}}$ to define the denoising function F, since $f_{\boldsymbol{\theta}}$ can determine the shortest path.

Here, inspired by the idea of NeuralPull [3], we also leverage the signed distance $d = f_{\boldsymbol{\theta}}(\boldsymbol{q})$ and the gradient $\nabla f_{\boldsymbol{\theta}}(\boldsymbol{q})$ to pull an arbitrary point $\boldsymbol{q}$ on the noisy point cloud $\boldsymbol{S}$ onto the surface. So we define the denoising function F below,

$$F(\boldsymbol{q}, f_{\boldsymbol{\theta}}) = \boldsymbol{q} - d \times \nabla f_{\boldsymbol{\theta}}(\boldsymbol{q}) / ||\nabla f_{\boldsymbol{\theta}}(\boldsymbol{q})||_2. \tag{3}$$

With (3), we can pull all points on the noisy point cloud $\boldsymbol{S}$ onto the surface, which results in a point cloud $\boldsymbol{S}' = F(\boldsymbol{S}, f_{\boldsymbol{\theta}})$.

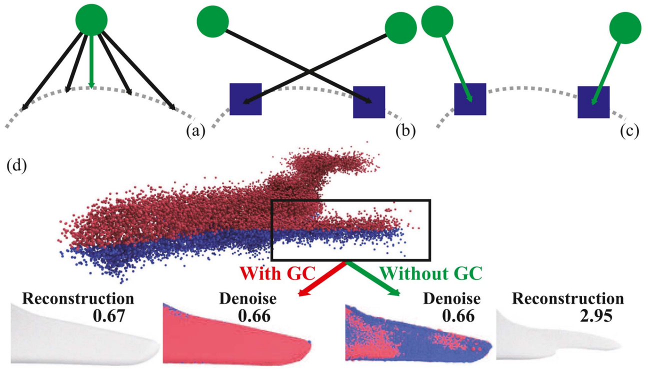

Fig. 3. (a) Multiple paths (arrows) to pull a noise (green point) onto surface (dashed curve) but only one is the shortest (green arrows). (b) The incorrect paths (black arrows) to pull noises onto surface. (c) The expected paths (green arrows) to pull noises to points (blue square) on surface. (d) The effect of Geometric Consistency (GC).

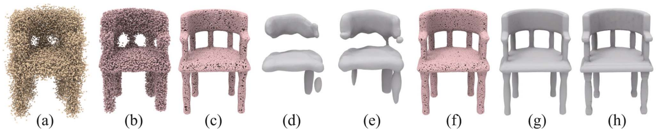

Fig. 4. The comparison with CD and EMD as the distance metric L from in (b) to (e). The effect of geometric regularization in (f) and (g). (a) is noisy point cloud, (h) is the ground truth.

But one issue remaining is how to constrain $\boldsymbol{S}'$ to converge to the uncorrupted surface.

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

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

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

We prove Theorem 1 in Sec.A of the supplementary materials. We believe the one-to-one correspondence ϕ found in the calculation of EMD in (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 $\boldsymbol{G}'$ optimized with EMD and Chamfer Distance (CD) as L based on the same observation set S in Fig. 4. Given noisy point clouds $\boldsymbol{S}_i$ like in Fig. 4(a) and (b) demonstrates that the point cloud $\boldsymbol{G}'$ 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 (2). F produces the denoised point cloud $\boldsymbol{S}_i' = F(\boldsymbol{S}_i, f_{\boldsymbol{\theta}})$ using EMD as the distance metric L. This also leads to one term in our loss function below,

$$\min_{\theta} \sum_{\boldsymbol{S}_i\in S} \sum_{\boldsymbol{S}_j\in S} L_{\text{EMD}}(F(\boldsymbol{S}_i, f_{\boldsymbol{\theta}}), \boldsymbol{S}_j). \qquad (5)$$

Geometric Consistency: Although the term in (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 [46] in Fig. 4(d) and (e) are poor. This is because (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 [3]. The issue is further demonstrated in Fig. 3, one situation that may happen is shown in Fig. 3(b). With the wrong signed distances $f_{\boldsymbol{\theta}}$ and gradient $\nabla f_{\boldsymbol{\theta}}$, 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 $\boldsymbol{q}$ around a noisy point cloud $\boldsymbol{S}_i$, the shortest distance between $\boldsymbol{q}$ and the surface can be either predicted by the SDFs f_θ or calculated based on the denoised point cloud $\boldsymbol{S}_i' = F(\boldsymbol{S}_i, f_\theta)$, both of which should be consistent to each other. Therefore, the absolute value $|f_\theta(\boldsymbol{q})|$ of the signed distance predicted at $\boldsymbol{q}$ should equal to the minimum distance between $\boldsymbol{q}$ and the denoised point cloud $\boldsymbol{S}_i' = F(\boldsymbol{S}_i, f_\theta)$. Since the point density of $\boldsymbol{S}_i'$ may slightly affect the consistency, we leverage an inequality to describe the geometric consistency,

$$|f_\theta(\boldsymbol{q})| \le \min_{\boldsymbol{q}'\in F(\boldsymbol{S}_i, f_\theta)} ||\boldsymbol{q} - \boldsymbol{q}'||_2. \qquad (6)$$

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 (5) and (6),

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

where $|\boldsymbol{S}_i|$ is the number of $\boldsymbol{q}$ on $\boldsymbol{S}_i$, E is the difference defined as $(|f_\theta(\boldsymbol{q})| - \min_{\boldsymbol{q}'\in F(\boldsymbol{S}_i, f_\theta)} ||\boldsymbol{q} - \boldsymbol{q}'||_2)$, λ is a balance

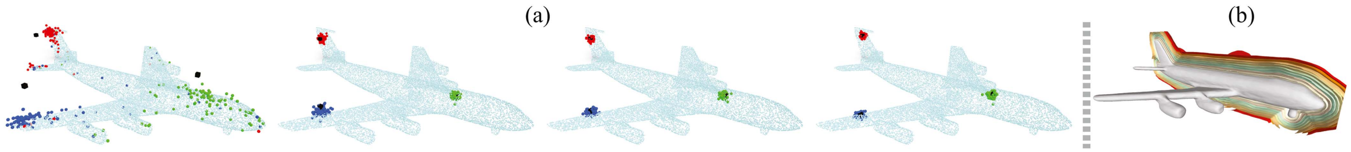

Fig. 5. (a) Visualization of optimization in 4 epochs via noise to noise mapping. 3 queries (black cubes) sampled from one noisy point cloud get pulled onto the surface. For each query, we minimize its distance to all targets (in the same color) matched from another noisy point cloud by the mapping ϕ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

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 $\boldsymbol{S}_i$ using the method introduced in NeuralPull [3]. We randomly sample a batch of B queries as input, and also randomly sample the same number of points from another noisy point cloud $\boldsymbol{S}_j$ 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 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 statistical reasoning in each epoch is that each query will 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 $\boldsymbol{S}_i$ as input and regard $\boldsymbol{S}_i$ 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.

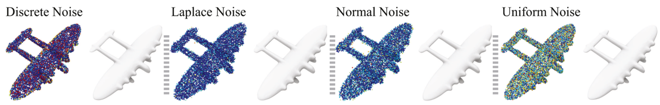

Fig. 6. Reconstruction with different kinds of noises.

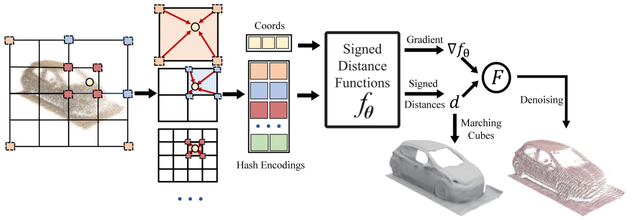

Fig. 7. Overview of our fast learning framework. We leverage the multi-resolution hash encoding as the efficient representation for fast learning SDFs from noisy point clouds.

IV. Fast Learning of Implicit Representations

Review Instant-NGP: The deep coordinate-based MLPs are proven to be slow in learning implicit functions (e.g., NeRF and SDF). To overcome this issue, Instant-NGP [11] leverages a multi-resolution hash encoding to learn radiance fields. Specifically, Instant-NGP divides the bounded space of a 3D shape into multiple voxel grids at different resolutions. The voxel grids at each resolution are then mapped to a hash table with learnable feature vectors.

Fast Learning of SDFs from Noisy Point Clouds: We propose a fast learning schema to learn signed distance fields from noisy point clouds by leveraging the multi-resolution hash encoding [11] as an efficient representation. The overview of our approach is shown in Fig. 7. We first initialize a hash encoding with L levels of resolution, where each level $l = 1, \ldots, L$ has a hash table T_l. Given a point $\boldsymbol{q} \in \mathcal{R}^{1\times 3}$ from one noisy point cloud $\boldsymbol{S}$, we obtain its feature $h_l(\boldsymbol{q})$ at level l using trilinear interpolating on the hash encodings of the eight surrounding corner grids g_i queried from the hash table T_l. The hash encoded feature $\{h_l(\boldsymbol{q})\}_{l=1}^{L}$ of $\boldsymbol{q}$ at all L levels, as well as the coordinate of $\boldsymbol{q}$ are then concatenated as the input to the MLP $f_{\boldsymbol{\theta}}$ to decode the signed distance value d. The similar denoising function as (3) is than leveraged for moving points floating off the surface of 3D shape onto the surface as

$$F_{\text{Fast}}(\boldsymbol{q}, \{h_l(n)\}_{l=1}^{L}, f_{\boldsymbol{\theta}}) = \boldsymbol{q} \\ - d \times \nabla f_{\boldsymbol{\theta}}(\boldsymbol{q}, \{h_l(n)\}_{l=1}^{L}) / ||\nabla f_{\boldsymbol{\theta}}(\boldsymbol{q}, \{h_l(n)\}_{l=1}^{L})||_2. \quad (8)$$

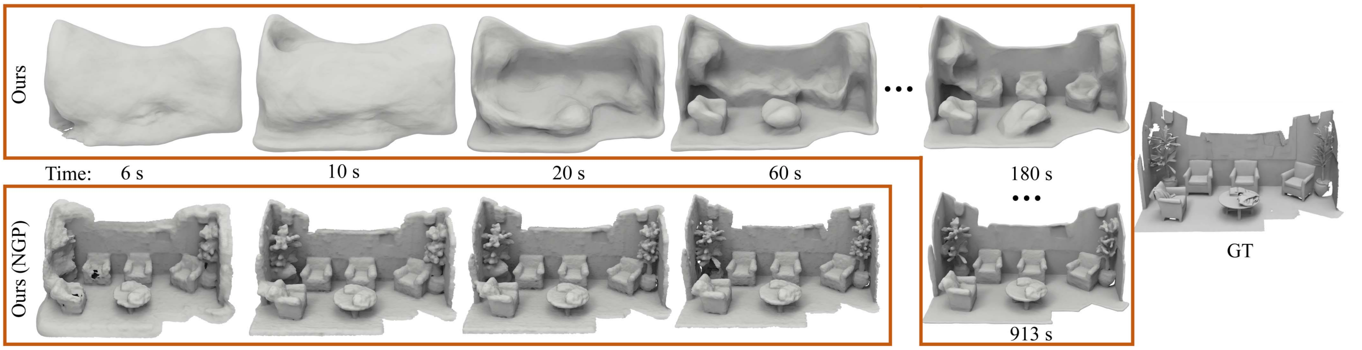

Fig. 8. Comparisons on the reconstructions at different optimization time. The timing results do not include the time for Marching Cubes, and are reported under one single RTX3090 GPU.

The condition here is the hash encoded feature $\{h_l(\boldsymbol{q})\}_{l=1}^L$ of $\boldsymbol{q}$ instead of the feature of entire shape $\boldsymbol{Z}$. We then optimize both the feature vectors in the multi-resolution hash encodings and the parameter $\boldsymbol{\theta}$ of MLP decoder $f_{\boldsymbol{\theta}}$ by minimizing the designed objective function in (7) which combines the EMD loss in (5) and the geometric consistency loss in (6). We provide the visualization of reconstructions at different optimization time with or without our hash encodings in Fig. 8. The results show that our fast learning schema reduces the training time from 15 minutes to one minute with comparable surface qualities, and is able to produce a visual appearing reconstruction within 6 to 10 seconds. All timing results are reported with one single RTX3090 GPU.

More Analyses and Details: Different from the original version of our method [10] which learns deep coordinate-based MLPs for geometry representation, the learning of multi-resolution hash encodings is more difficult to stabilize and requires some other constraints besides the EMD loss in (5). The main problem lies in the instability of the Earth Moving Distance (EMD) loss during optimizations in regions far away from the surface, as establishing a one-to-one correspondence between queries far away from the surface and the point cloud S is challenging. This was not a significant issue in the previous MLP-based version [10], which only focused on the surface's nearby space. However, optimizing the far-space regions is crucial for multi-resolution hash encodings, as it ensures that the hash grids in all spatial spaces are well-optimized, leading to refined representations. To solve this issue, we sample queries $\{Q = q_i, i \in [1, N_q]\}$ at all spacial space and leverage Neural-Pull [94] loss to regularize the gradients and SDFs (especially in areas that is far away from the surface) by pulling each query q_i to its nearest point $\boldsymbol{n'}_i$ in S, formulated as

$$L_{\text{pull}} = \frac{1}{N_q}\sum_{i=1}^{N_q} ||q_i - n'_i||_2^2. \tag{9}$$

The eikonal term [95] is also used to regularize the gradients of SDFs, formulated as

$$L_{\text{reg}}(Q, \boldsymbol{c}) = \frac{1}{N_q}\sum_{i=1}^{N_q} (|\nabla f_{\boldsymbol{\theta}}(q_i, c)| - 1)^2. \tag{10}$$

The finally loss function is formulated as

$$L_{\text{Fast}} = L_{\text{EMD}} + \lambda_1 L_{\text{pull}} + \lambda_2 L_{\text{reg}}, \tag{11}$$

where λ_2 is set to 0.001. λ_1 is set to decrease from 1 to 0 during the first 1,000 iterations and is omitted for the rest iterations, where we train our fast learning framework for 10,000 iterations in total. We only leverage L_{pull} to stable the optimizations in the areas far away from the surface and omit it after a fine initialized fields is achieved, since it will lead to large errors to pull the queries to the noisy inputs in the areas near to the true surface.

V. Learning SDF Priors for Multi-View Reconstruction

Review NeuS: We extend our noise to noise mapping to learn SDF priors for multi-view reconstruction. We use NeuS as our baseline. NeuS [12] is a recent work which focuses on reconstructing surfaces from a set of calibrated images $\{\mathcal{I}_k\}$ of a 3D object. To achieve this, NeuS implicitly learns a signed distance function $f_{mv}(x) : \mathbb{R}^3 \rightarrow \mathbb{R}$ and a radiance function $c(x, v) : \mathbb{R}^3 \times \mathbb{S}^2$, where $x \in \mathbb{R}^3$ is a 3D location and $v \in \mathbb{S}^2$ is a view direction. The surfaces of the 3D object are implicitly represented as the zero-level set of f_{mv} and the appearances are encoded in c. To optimize the two functions, NeuS leverages volume rendering to render novel images from f_{mv} and c and minimize the differences between the rendered images and input images. Specifically, given a pixel of an image, we sample h queries along the ray emitted from this pixel as $\{p(t) = o + tv | t = 0, 1 \ldots h\}$, where o is the camera center and v is the view direction. The color of this pixel is then obtained by accumulating the SDF-based densities and colors at the sampled queries.

Learn SDF Priors for NeuS: The key factor that prevent NeuS [12] from producing high-fidelity and artifact-free geometries is the bias in color rendering [62], [64], [96]. As demonstrated by previous methods [62], [64], there is a gap between the rendered colors and the real colors of the 3D object surfaces, which leads to geometry inconsistency between the implicit represented surfaces indicated as the zero-level set of field $f_{mv}(*)$ and the true surfaces of the 3D object. To solve this issue, a more precise and adaptive prior is required as the

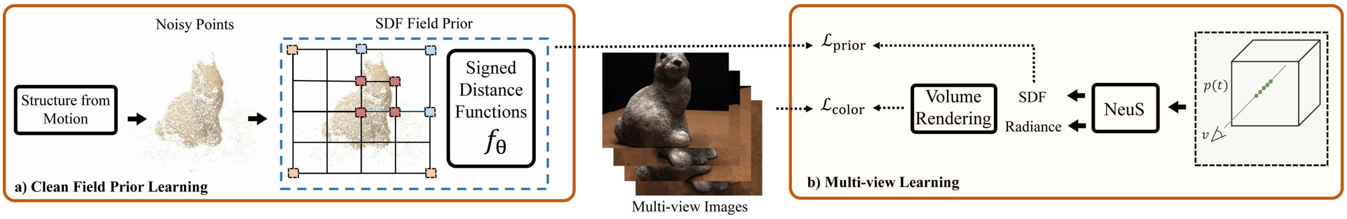

Fig. 9. Overview of our multi-view reconstruction framework. (a) We extract a noisy and sparse point cloud from multi-view images with SfM and learn a clean SDF field prior from it with our fast learning denoising approach. (b) We guide the multi-view learning (NeuS) with the learned clean field prior by introducing $\mathcal{L}_{\text{prior}}$.

supervision for a better optimization on the signed distance function.

Specifically, we propose to learn a signed distance field from the point cloud estimated by Structure from Motion (SfM) as a prior for multi-view reconstruction, as shown in Fig. 9(a). The key insight is that the sparse points P_s generated by SfM is an exact geometry information for the 3D object. SfM matches 2D features at the extracted 2D key points to estimate camera poses and compute the 3D point locations through triangulation which guarantees the geometry consistency of P_s in multi-view images. However, the generated P_s often contains noises due to some wrong key point matchings or errors in camera motion estimation.

We aim to learn a clean and accurate signed distance field from P_s to serve as a prior for multi-view reconstruction, where the noises prevent previous works [49], [94] from predicting an accurate SDF field. To this end, we leverage noise to noise mapping which is robust to noises with different distributions, to learn a SDF prior from the noisy points P_s. The learning process keeps the same as Sections III and IV.

Guide NeuS With SDF Priors: With a learned clean field f_{prior}, we introduce a field constraint to guide the learning of NeuS for improving multi-view reconstruction quality as shown in Fig. 9(b). The key factor is to leverage f_{prior} as a supervision for the signed distance function f_{mv} to be optimized. Specifically, given a pixel of an image and h queries sampled along the ray emitted from this pixel as $\{p(t) = o + tv | t = 0, 1 \ldots h\}$, we minimize the difference between the predicted signed distance value $f_{mv}(q)$ and the pre-learned signed distance prior $f_{prior}(q)$. The designed loss is formulated as

$$L_{\text{prior}} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} ||f_{mv}(q_{ij}) - f_{prior}(q_{ij})||_2^2, \tag{12}$$

where M is the number of pixels and N is the number of points sampled each pixel. We set the weight of $\mathcal{L}_{\text{prior}}$ to gradually decrease during the training process for avoiding overfitting the field to the prior. Specifically, we train our framework for 100k iterations and decrease the weight of $\mathcal{L}_{\text{prior}}$ from 1 to 0 linearly in the first 10k iterations and omit it for the rest iterations.

We further design a novel constraint on the zero level set of f_{mv} with a loss to minimizing the distance errors in the exact surface location. To achieve this, we generate a set of points $\{p_i^{zls}\}_{i=1}^K$ at the zero level set of learned prior f_{prior} by moving queries along the gradient directions with a stride of signed distances as (8). These points $\{p_i^{zls}\}_{i=1}^K$ at the zero level set of the learned prior f_{prior} can serve as another prior of the zero level set of f_{mv} with a simple loss to minimize the distance values of f_{mv} at p^{zls}, formulated as

$$L_{\text{zls}} = \frac{1}{K} \sum_{i=1}^{K} ||f_{mv}(p_k^{zls}) - 0||_2^2, \tag{13}$$

VI. Experiments, Analysis and Applications

We evaluate our performance in point cloud denoising, upsampling and surface reconstruction from point clouds or multi-view images. We first evaluate our method in point cloud denoising in Section VI-A. Next, we go beyond denoising and evaluate our method in point cloud upsampling in Section VI-B. We then evaluate our method in surface reconstruction from point clouds for shapes in Section VI-C and for scenes in Section VI-D. In Section VI-E, we show the efficiency of our fast learning framework in runtime comparison. We further demonstrate the effectiveness of our method to improve the multi-view reconstruction qualities in Section VI-F. Finally, the ablation studies are shown in Section VI-G.

A. Point Cloud Denoising

Dataset and Metric: For the fair comparison with the state-of-the-art results, we follow SBP [92] to evaluate our method under two benchmarks named as PU and PC that were released by PUNet [97] and PointCleanNet [87]. 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 10 K and 50 K points from each shape respectively as the ground truth clean point clouds in two different resolutions. The 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 Table I. The compared methods include Bilateral [98], Jet [99], MR-PCA [100], GLR [101], PCNet [87], GPDNet [88], DMR [91], TTD [89], and SBP [92]. 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

TABLE I
DENOISING COMPARISON

	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

L2CD×10^4 and P2M ×10^4.

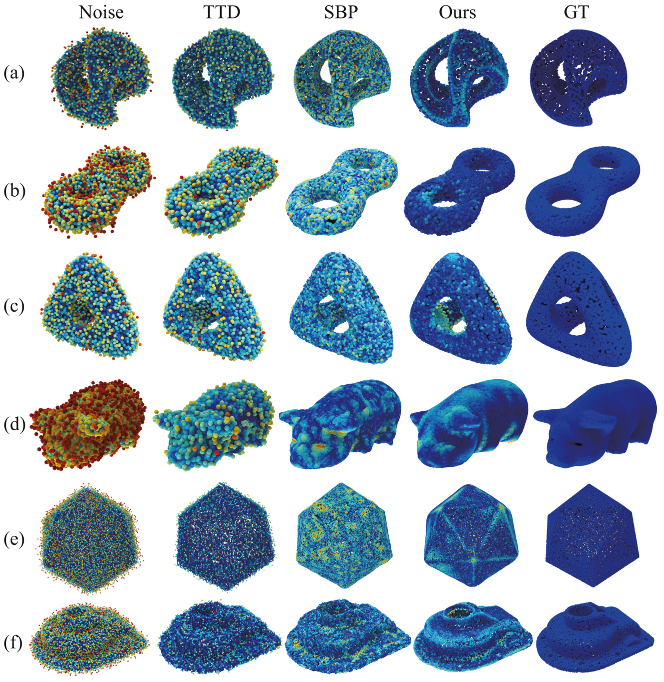

Fig. 10. Visual comparison in point cloud denoising. Error at each point is shown in color. (a) and (b) $10K$ points with 3% noise. (c) $10K$ points with 2% noise. (d) and (e) $50K$ points with 3% noise. (f) $50K$ points with with 2% noise.

TABLE II
UPSAMPLING COMPARISON

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**

L2CD×10^4 and P2M ×10^4.

learning based point cloud denoising methods in both supervised and unsupervised ("-Un") settings. Error map comparison with TTD [89] and SBP [92] in Fig. 10 further demonstrates our state-of-the-art denoising performance.

B. Point Cloud Upsampling

From Denoising to Upsampling: Point cloud upsampling [92], [97] aims at generating dense and uniformly distributed point sets from a sparse point cloud. We justify that our proposed method is also applicable to point cloud upsampling. Specifically, given a sparse point cloud with N points as input, we perturb it with Gaussian noise independently for t times, resulting in a noisy dense point cloud with tN points, which is then feed to our denoising framework to acquire the final upsampled point cloud.

Dataset and Metric: We use the PU dataset mentioned before to evaluate the $f_{\boldsymbol{\theta}}$ learned in our denoising experiments in point cloud upsampling. Following SBP [92], 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 Table II. We compared with PU-Net [97] and SBP [92]. The comparison demonstrates that our method can perform the statistical reasoning to reveal points on the surface more accurately.

C. Surface Reconstruction for Shapes

ShapeNet: We first report our surface reconstruction performance under the test set of 13 classes in ShapeNet [102]. The train and test splits follow COcc [103]. Following IMLS [8], we leverage point clouds with 3,000 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) [37], and F-score [104] with a threshold of 1% as metrics.

TABLE III
L1CD, NC and F-Score Comparison Under ShapeNet

-	L1CD										Normal Consistency										F-Score									
-	PSR	PSG	R2N2	Atlas	COcc	SAP	OCNN	IMLS	POCO	**Ours**	PSR	PSG	R2N2	Atlas	COcc	SAP	OCNN	IMLS	POCO	**Ours**	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**	0.747	-	0.669	0.854	0.931	0.931	0.918	0.937	0.944	**0.960**	0.551	0.476	0.382	0.827	0.965	0.981	0.810	0.992	0.994	**0.995**
bench	0.544	0.128	0.153	0.073	0.035	0.032	0.065	0.030	0.028	**0.025**	0.649	-	0.691	0.820	0.921	0.920	0.914	0.922	0.928	**0.935**	0.430	0.266	0.431	0.786	0.965	0.979	0.800	0.986	0.988	**0.993**
cabinet	0.154	0.164	0.167	0.112	0.047	0.037	0.071	0.035	0.037	**0.034**	0.835	-	0.786	0.875	0.956	0.957	0.941	0.955	0.961	**0.975**	0.728	0.137	0.412	0.603	0.955	0.975	0.789	0.981	0.979	**0.996**
car	0.180	0.132	0.197	0.099	0.075	0.045	0.077	0.040	0.041	**0.037**	0.783	-	0.719	0.827	0.893	0.897	0.867	0.882	0.894	**0.937**	0.729	0.211	0.348	0.642	0.849	0.928	0.747	0.952	0.946	**0.964**
chair	0.369	0.168	0.181	0.114	0.046	0.036	0.066	0.035	0.033	**0.026**	0.715	-	0.673	0.829	0.943	0.952	0.941	0.950	0.956	**0.965**	0.473	0.152	0.393	0.629	0.939	0.979	0.799	0.982	0.985	**0.993**
display	0.280	0.160	0.170	0.089	0.036	0.030	0.066	0.029	0.028	**0.022**	0.749	-	0.747	0.905	0.968	0.972	0.960	0.973	0.975	**0.981**	0.544	0.175	0.401	0.727	0.971	0.990	0.811	0.994	0.994	**0.998**
lamp	0.278	0.207	0.243	0.137	0.059	0.047	0.067	0.031	0.033	**0.027**	0.765	-	0.598	0.759	0.900	0.921	0.911	0.922	0.929	**0.957**	0.586	0.204	0.333	0.562	0.892	0.959	0.800	0.979	0.975	**0.990**
speaker	0.148	0.205	0.199	0.142	0.063	0.041	0.073	0.040	0.041	**0.033**	0.843	-	0.735	0.867	0.938	0.950	0.936	0.947	0.952	**0.977**	0.731	0.107	0.405	0.516	0.892	0.957	0.779	0.963	0.964	**0.977**
rifle	0.409	0.091	0.167	0.051	0.028	0.023	0.062	0.021	0.019	**0.019**	0.788	-	0.700	0.837	0.929	0.937	0.932	0.943	**0.949**	0.938	0.590	0.615	0.381	0.877	0.980	0.990	0.826	0.996	0.998	**0.998**
sofa	0.227	0.144	0.160	0.091	0.041	0.032	0.066	0.031	0.030	**0.027**	0.826	-	0.754	0.888	0.958	0.963	0.949	0.963	0.967	**0.978**	0.712	0.184	0.427	0.717	0.953	0.982	0.801	0.987	0.989	**0.992**
table	0.393	0.166	0.177	0.102	0.038	0.033	0.066	0.032	0.031	**0.028**	0.706	-	0.734	0.867	0.959	0.962	0.946	0.962	0.966	**0.970**	0.442	0.158	0.404	0.692	0.967	0.986	0.801	0.987	0.991	**0.992**
telephone	0.281	0.110	0.130	0.054	0.027	0.023	0.061	0.023	0.022	**0.017**	0.805	-	0.847	0.957	0.983	0.984	0.974	0.984	0.985	**0.987**	0.674	0.317	0.484	0.867	0.989	0.997	0.825	0.998	0.998	**0.999**
vessele	0.181	0.130	0.169	0.078	0.043	0.030	0.064	0.027	0.025	**0.024**	0.820	-	0.641	0.837	0.918	0.930	0.922	0.932	0.940	**0.951**	0.771	0.363	0.394	0.7757	0.931	0.974	0.809	0.987	0.989	**0.997**
mean	0.299	0.147	0.173	0.093	0.044	0.034	0.067	0.031	0.030	**0.026**	0.772	-	0.715	0.855	0.938	0.944	0.932	0.944	0.950	**0.962**	0.612	0.259	0.400	0.708	0.942	0.975	0.800	0.983	0.984	**0.991**

TABLE IV
Comparison Under ABC and Famous Datasets With P2S-CD and L2CD Settings

-	P2S-CD						L2CD				
Dataset	DSDF	Atlas	PSR	P2S	POCO	Ours	NP	IMLS	OnSF	PCP	Ours
ABC var	12.51	4.04	3.29	2.14	2.01	**1.87**	0.72	0.57	3.52	0.49	**0.113**
ABC max	11.34	4.47	3.89	2.76	2.50	**2.42**	1.24	-	4.30	0.57	**0.139**
F-med	9.89	4.54	1.80	1.51	1.50	**1.32**	0.28	0.80	0.59	0.07	**0.033**
F-max	13.17	4.14	3.41	2.52	2.75	**2.31**	0.31	-	3.64	0.30	**0.117**

We compare our methods with methods including PSR [6], PSG [105], R2N2 [106], Atlas [107], COcc [103], SAP [5], OCNN [108], IMLS [8] and POCO [54]. The numerical comparison in Table III 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 [5], COcc [103] and IMLS [8]. We further demonstrate our superiority in the reconstruction of complex geometry in the visual comparison in Fig. 11. More numerical and visual comparisons can be found in our supplemental materials.

FAMOUS and ABC: We further evaluate our method using the test set in FAMOUS and ABC dataset provided by P2S [48]. The clean point cloud is corrupted with noise at different levels. We follow NeuralPull [3] 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 DSDF [31], Atlas [107], PSR [6], P2S [48], NP [3], IMLS [8], PCP [81], POCO [54], and OnSF [80]. Note that the implementation of Chamfer Distance (CD) used in P2S [48] and occupancy network [37] are different. For a comprehensive and fair comparison with the baselines evaluated under different settings, we report the performance of our method in both P2S-CD setting and the L2CD setting used in occupancy network, and make comparisons under both settings to demonstrate the effectiveness of our method. The comparison in Table IV 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. 12.

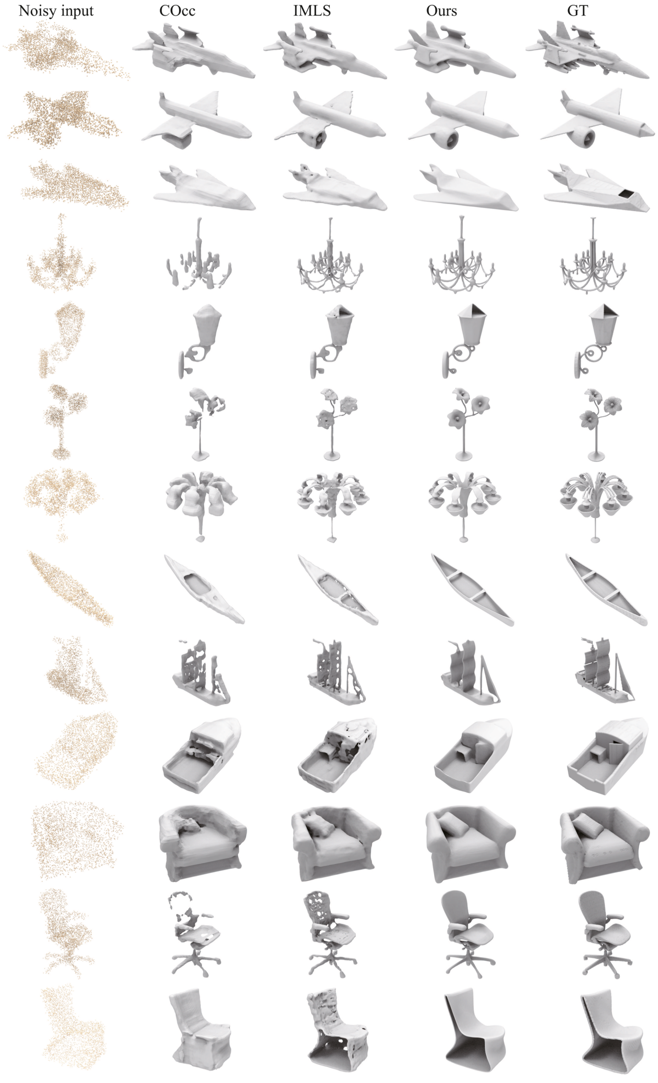

Fig. 11. Comparison in surface reconstruction under ShapeNet.

D-FAUST and SRB: Finally, we evaluate our method under the real scanning dataset D-FAUST [109] and SRB [110]. We follow SAP [5] to evaluate our result using L1CD, NC [37], and F-score [104] 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 $B = 5000$.

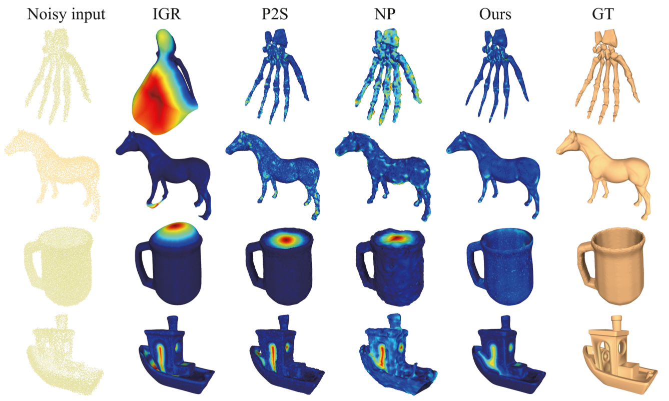

Fig. 12. Visual comparison in surface reconstruction under FAMOUS. Point to surface error at each vertex is shown in color.

TABLE V
COMPARISON IN SURFACE RECONSTRUCTION UNDER D-FAUST

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 VI
COMPARISON IN SURFACE RECONSTRUCTION UNDER SRB

Metrics	IGR	Point2Mesh	PSR	NDF	NP
L1CD×10	0.178	0.116	0.232	0.185	0.106
F-Score	0.755	0.648	0.735	0.722	0.797

Metrics	MLOD	Neural-IMLS	SAP	DiGS	**Ours**
L1CD×10	0.104	0.075	0.076	0.069	**0.067**
F-Score	0.765	0.822	0.830	**0.839**	0.835

We compare our methods with the methods including IGR [1], Point2Mesh [111], PSR [6], NDF [79], NP [94], MLOD [112], Neural-IMLS [113], SAP [5] and DiGS [77]. We report numerical comparisons in Tables V and VI. Although we only do statistical reasoning on a single noisy point cloud and do not require point normals, our method still handles the noise in real scanning well, which achieves much smoother and more accurate structure. The comparison in Figs. 13 and 14 shows that our method can produce more accurate surfaces without missing parts on both rigid and non-rigid shapes.

D. Surface Reconstruction for Scenes

3D Scene: We evaluate our method under real scene scan dataset [114]. We sample 1,000 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 compare our method with SOTA data-driven based methods and overfitting based methods including LIG [4], COcc [49], NP [94], DeepLS [9], POCO [115], ALTO [116], GridFormer [117] and NKSR [118]. We leverage the pretrained models of COcc, LIG, POCO, ALTO, GridFormer and NKSR, and retrain NP and DeepLS to produce their results with the same input. Numerical

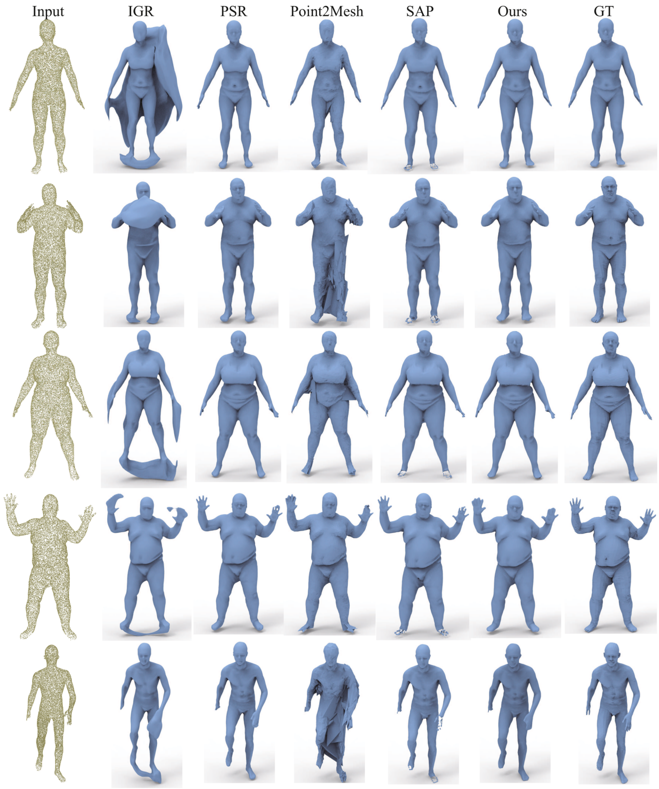

Fig. 13. Comparison in surface reconstruction under DFAUST.

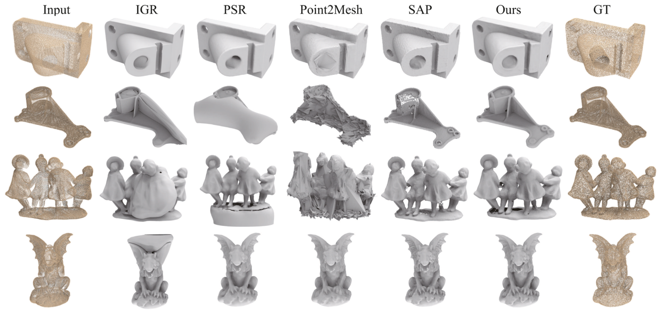

Fig. 14. Comparison in surface reconstruction under SRB.

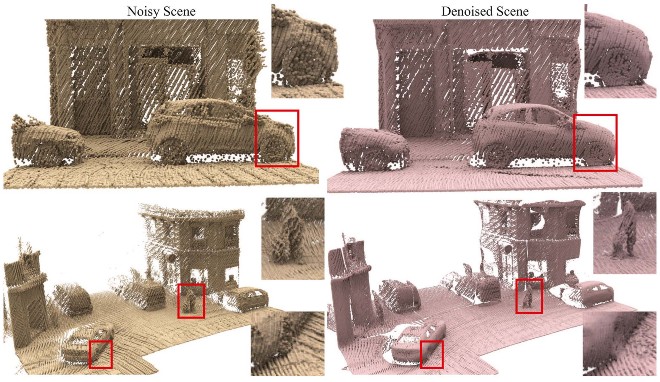

Fig. 15. Demonstration of denoising on real scans.

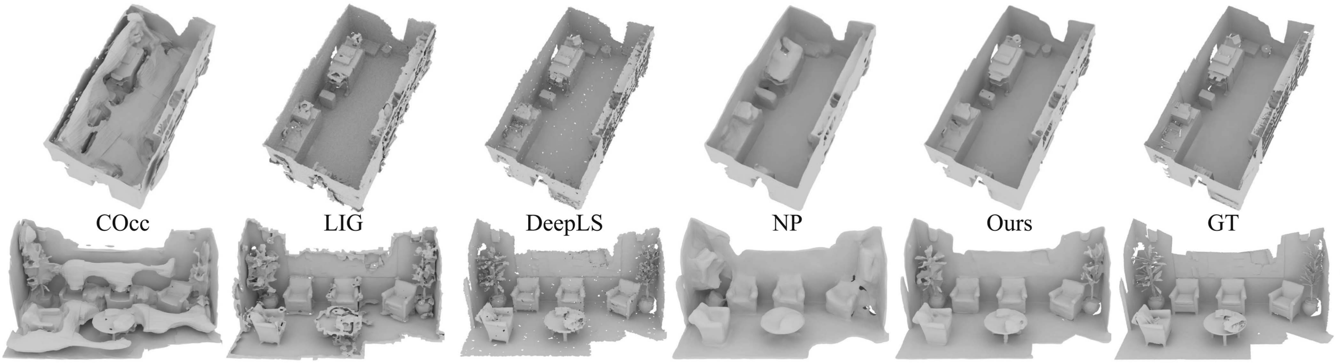

Fig. 16. Visual comparison in surface reconstruction under 3D Scene dataset.

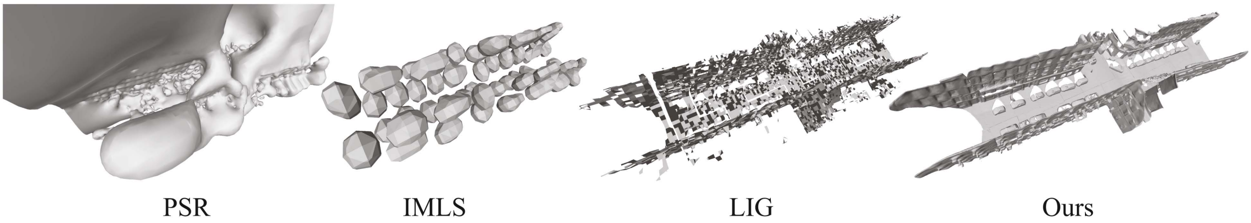

Fig. 17. Comparison in surface reconstruction from real scans.

TABLE VII
SURFACE RECONSTRUCTION UNDER 3D SCENE DATASET

	Lounge			Copyroom		
	L2CD	L1CD	NC	L2CD	L1CD	NC
COcc [103]	9.540	0.046	0.894	10.97	0.045	0.892
LIG [4]	9.672	0.056	0.833	3.61	0.036	0.810
DeepLS [9]	6.103	0.053	0.848	0.609	0.021	0.901
NP [3]	1.079	0.019	0.910	5.795	0.036	0.862
POCO [54]	1.122	0.024	0.912	1.468	0.025	0.883
ALTO [116]	1.716	0.033	0.904	1.095	0.028	0.839
GridFormer [117]	1.345	0.028	0.901	3.485	0.033	0.864
NKSR [118]	0.617	0.019	0.917	0.527	0.017	**0.908**
Ours	**0.602**	**0.016**	**0.923**	**0.442**	**0.016**	0.903

L2-CD×10^3. The unit of error is mm.

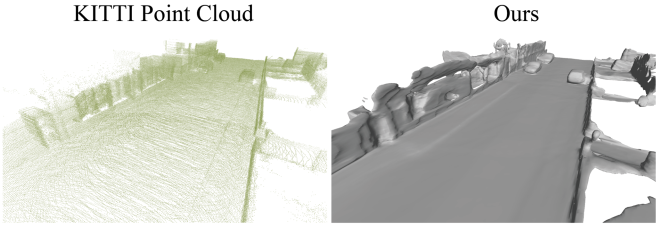

Fig. 18. Reconstruction on a real scan from KITTI.

TABLE VIII
SURFACE RECONSTRUCTION UNDER 3D SCENE DATASET

	Lounge		Copyroom		Training time
	L2CD	L1CD	L2CD	L1CD	
DeepLS [9]	6.103	0.053	0.609	0.021	-
NP [3]	1.079	0.019	5.795	0.036	1150 s
SAP [5]	1.801	0.025	1.992	0.027	288 s
Ours	0.602	**0.016**	**0.442**	0.016	913 s
Ours (+Fast)	**0.496**	0.020	0.457	**0.015**	**63 s**

L2-CD×10^3. The unit of error is mm.

comparison in Table VII demonstrates that our method outperforms the state-of-the-art results. Fig. 16 further demonstrates that we can produce much smoother surfaces with more geometry details. For more analysis and qualitative comparisons under 3DScene dataset, please refer to the supplementary materials.

Paris-rue-Madame: We further evaluate our method under another real scene scan dataset [119]. We only use $N = 1$ noisy point cloud with a batch size of $B = 5000$. We split the $10\,M$ points into 50 chunks each of which is used to learn a SDF. Similarly, we use each chunk to evaluate IMLS [8] and LIG [4] with their pretrained models. Our superior performance over the latest methods in large scale surface reconstruction is demonstrated in Fig. 17. Our denoised point clouds in a smaller scene are detailed in Fig. 15.

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 captured by vehicular radars.

E. Fast Learning Reconstruction and Denoising

We evaluate our fast learning framework in point cloud denoising and scene reconstruction tasks. The metrics and experiment settings keep the same as Sections VI-A and VI-D. All timing results are reported on one single RTX3090 GPU.

Fast Learning Reconstruction: We conduct experiments under 3D Scene dataset for evaluating the surface reconstruction ability of our fast learning framework. Numerical comparisons

TABLE IX
POINT CLOUD DENOISING UNDER PUNET DATASET

	10K(Sparse)		50K(Dense)		Training time
	CD	P2M	CD	P2M	
Ours	**4.221**	1.847	**1.654**	**0.972**	1308 s
Ours (+Fast)	4.537	**1.723**	1.889	1.273	**102 s**

L2CD$\times 10^4$ and P2M $\times 10^4$.

in Table IX show that our fast learning framework can reconstruct complex scenes in about one minute and achieves comparable performance to our origin MLP-based approach for about 15 minutes. Our method also outperforms the state-of-the-art work on fast reconstruction, i.e., SAP [5], in terms of both speed and qualities as shown in Table IX. The visualizations under different optimization time with or without our fast learning framework are shown in Fig. 8, where we can achieve a visual appearing reconstruction in 6 seconds with the multi-resolution hash encodings.

Fast Learning Denoising: We further evaluate our fast learning framework for point cloud denoising under PU dataset. Same as Section VI-A, we report our results under 20 shapes in the test set of PU with different densities of point clouds, i.e., 10K and 50K. We evaluate the results under large Gaussian noise with a standard deviation of 3%. The comparisons in Table VIII show that our fast learning framework (+Fast) achieves comparable performance to our origin MLP-based approach, yet is more than 10 times faster, which demonstrates the effectiveness and efficiency of our proposed fast learning framework.

F. Multi-View Reconstruction

Neural surface reconstruction from multi-view images [12], [13], [16], [26], [61], [96], [120], [121], [122], [123], [124], [125], [126], [127] has been shown to be powerful for recovering dense 3D surfaces via image-based neural rendering. We demonstrate that our method can also improve the performance of multi-view reconstruction by introducing an SDF through noise to noise mapping on point clouds from SfM [15] as a geometry prior.

Dataset and Metrics: To evaluate the effectiveness of our proposed schema to learn SDF priors for multi-view reconstruction, we conduct experiments under the DTU dataset [128] to reconstruct surfaces from multi-view images. Following previous works [12], [13], [16], [61], we report the results under the widely used 15 scenes. Each of these scenes captures 49 to 64 images around a single object with background. For quantitative evaluations under the DTU dataset, we first clean the reconstructed meshes using the provided masks following previous works [12], [13], [16], [61], and then leverage L1 Chamfer distance as the metric to evaluate the errors of the randomly sampled points on the reconstructed surfaces compare to the ground truth point clouds.

Comparison: We compare our method with the traditional method Colmap [15] and the state-of-the-art learning-based methods NeuS [12], UNISURF [16], VolSDF [13], Voxurf [129], MonoSDF [63] and HF-NeuS [64]. The numerical comparisons are reported in Table X, where our method achieves better performance than the state-of-the-art methods. It is worth noticing that our method, which is built upon NeuS [12], brings large performance gain than the baseline NeuS (0.87→0.72). This result demonstrates the effectiveness of our proposed schema which learns a clean field prior from noisy points as priors to serve as a guidance for multi-view learning. We also produce better reconstruction results than MonoSDF [63], which introduces both depth and normal priors from pretrained omnidata models learned using large-scale datasets as additional supervisions. The results demonstrate that our SDF priors perform better compared with the monocular priors achieve by large-scale data.

We further show visual comparisons with the state-of-the-arts before cleaning with GT masks in Fig. 19, where our method not only produces more accurate reconstructions, but also largely remove the artifacts in empty space. The reason is that our SDF priors also serves as a regularization constraint to guide the learning of SDFs at the empty space. More specifically, when getting supervised by our SDF priors, noisy points in the empty space will be removed. Note that the performance gain in removing artifacts is not reflected in the numerical results since the artifacts in empty space has been cleaned using the GT masks following the evaluation protocol.

We then provide a visual comparison on learning SDF priors from SfM noisy point clouds by visualizing the reconstructions of the learned SDF priors in Fig. 20. The results show that our methods can learn a clean and complete SDF prior from the noisy input, where the other methods (e.g., SAP [5]) lead to failures using noisy point clouds. The clean SDF prior helps NeuS [12] to learn the exact geometry information provided by SfM and also serve as a artifact-free field initialization which regularizes the optimization of multi-view reconstruction.

To further demonstrate the efficiency of our proposed method, we make a comparison with NeuS [12] on the reconstructions at different optimization time in Fig. 24. We exclude the time to get points and only show the time for optimization. The result shows that our method achieves much faster convergence than NeuS, where we can obtain a coarse reconstruction in one minute with the clean field prior. And the field prior also helps to learn accurate geometries, i.e., the complete rabbit ears and the clean and closed ground. There are some recent studies [130], [131] focus on accelerating NeuS by improving the framework of NeuS. We justify that our method have no intersection with these methods since we do not have special designs on the framework, but using SDF priors which can also be integrated into these methods to further improve the reconstruction qualities and training efficiency.

G. Ablation Studies

1) Training Settings: 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. Table XI indicates that more points in each batch will slow down the convergence. Table XII demonstrates that more training iterations help perform statistical reasoning

TABLE X
QUANTITATIVE COMPARISONS ON DTU

ScanID	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean
colmap [15]	0.81	2.05	0.73	1.22	1.79	1.58	1.02	3.05	1.4	2.05	1.00	1.32	0.49	0.78	1.17	1.36
IDR∗ [61]	1.63	1.87	0.63	0.48	1.04	0.79	0.77	1.33	1.16	0.76	0.67	0.90	0.42	0.51	0.53	0.90
UNISURF [16]	1.32	1.36	1.72	0.44	1.35	0.79	0.80	1.49	1.37	0.89	0.59	1.47	0.46	0.59	0.62	1.02
VolSDF [13]	1.14	1.26	0.81	0.49	1.25	0.70	0.72	1.29	1.18	0.70	0.66	1.08	0.42	0.61	0.55	0.86
Voxurf [129]	0.71	0.78	0.43	0.35	1.03	0.76	0.74	1.49	1.04	0.74	0.51	1.12	0.41	0.55	0.45	0.74
NeuS [12]	1.37	1.21	0.73	0.40	1.20	0.70	0.72	1.01	1.16	0.82	0.66	1.69	0.39	0.49	0.51	0.87
MonoSDF(MLP) [63]	0.83	1.61	0.65	0.47	0.92	0.87	0.87	1.30	1.25	0.68	0.65	0.96	0.41	0.62	0.58	0.84
MonoSDF(Grid) [63]	0.66	0.88	0.43	0.40	0.87	0.78	0.81	1.23	1.18	0.66	0.66	0.96	0.41	0.57	0.51	0.73
HF-NeuS [64]	0.76	1.32	0.70	0.39	1.06	0.63	0.63	1.15	1.12	0.80	0.52	1.22	0.33	0.49	0.50	0.77
Ours	0.72	0.79	0.58	0.39	0.96	0.64	0.62	1.41	0.89	0.82	0.53	1.13	0.34	0.51	0.48	**0.72**

We compare the proposed method to the state-of-the-art baselines, i.e., NeuS and HF-NeuS, and other methods using their released codes following their best configurations. IDR *means that the ground truth masks are required for IDR, where our method and other baselines do not use masks for training.

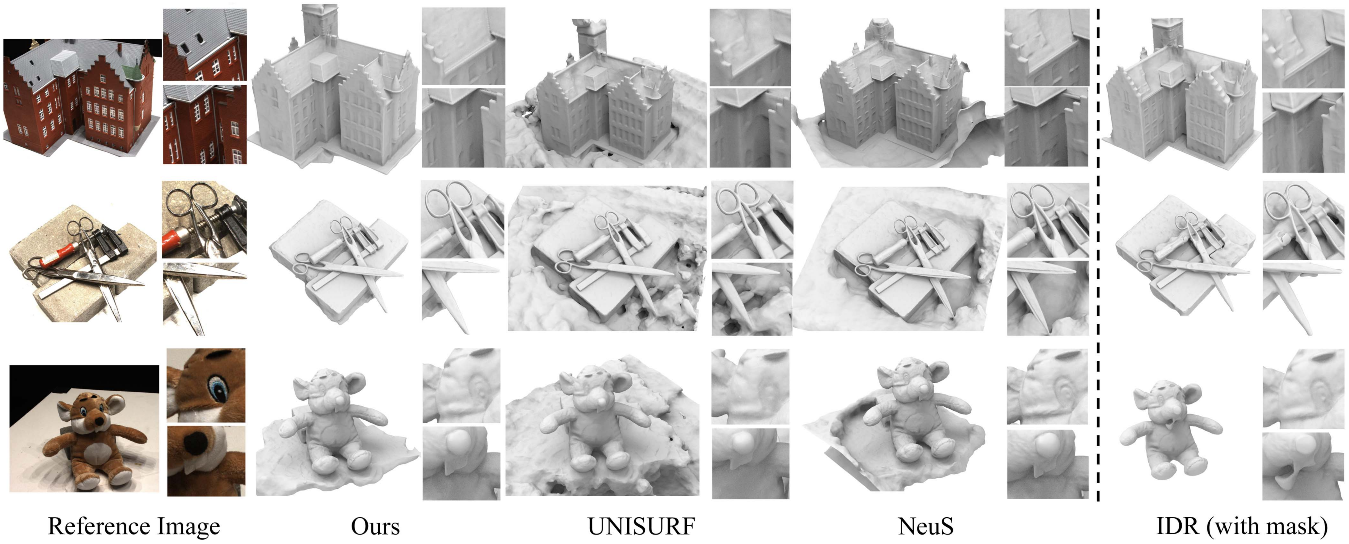

Fig. 19. Qualitative comparisons on DTU dataset. IDR requires masks as additional input.

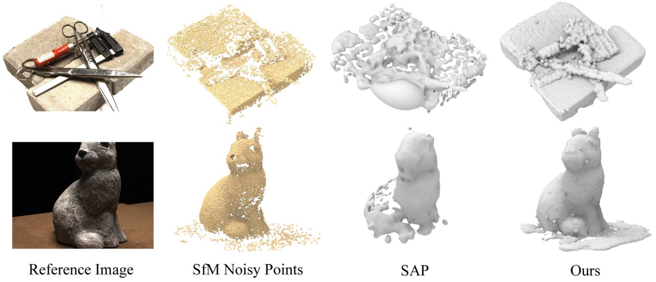

Fig. 20. Comparison in the learned SDF priors from noisy SfM point clouds.

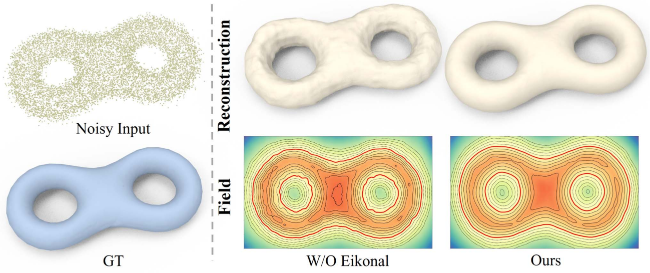

Fig. 21. The effect of Eikonal term in our fast learning framework.

TABLE XI
EFFECT OF BATCH SIZE B UNDER PU

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

TABLE XII
NUMBER OF TRAINING ITERATIONS UNDER PU

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

better to remove noise. Table XIII 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.

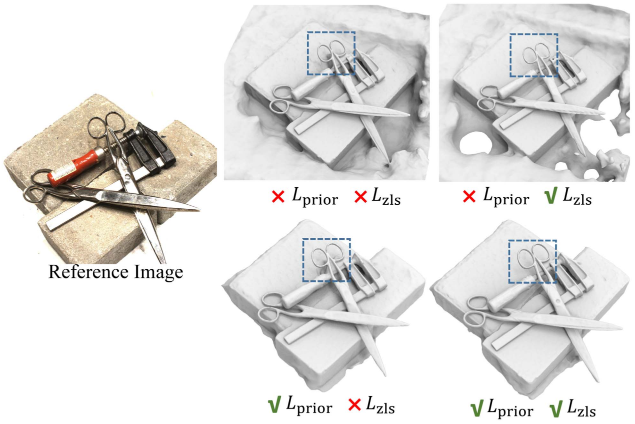
Reference Image
✗ L_{prior} ✗ L_{zls}
✗ L_{prior} ✓ L_{zls}
✓ L_{prior} ✗ L_{zls}
✓ L_{prior} ✓ L_{zls}

Fig. 22. Ablations on the designed losses in Multi-View Reconstruction task.

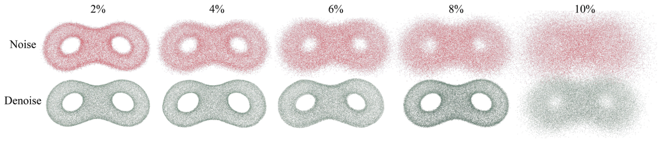

Fig. 23. Point clouds denoising with large noises.

TABLE XIII
EFFECT OF N UNDER PU

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

TABLE XIV
EFFECT OF CD AND EMD AS THE DISTANCE METRIC L AND GEOMETRY CONSISTENCY REGULARIZATION R UNDER PU

	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

L2CD×10^4.

TABLE XV
EFFECT OF NOISY POINT CLOUD NUMBERS L UNDER PU

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

TABLE XVI
EFFECT OF DENSITY D OF POINT CLOUD UNDER PU

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

TABLE XVII
EFFECT OF MIXING MULTIPLE NOISE POINT CLOUDS UNDER PU

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

TABLE XVIII
LOSS DESIGNS IN OUR FAST LEARNING FRAMEWORK

Designs	W/O $\mathcal{L}_{\text{pull}}$	W/O $\mathcal{L}_{\text{reg}}$	Full
L2CD×10^4	8.016	5.255	**4.537**
P2M×10^4	4.209	2.252	**1.723**

2) Loss Designs: We further highlight the effect of EMD as the distance metric L and geometric consistency regularization R in denoising and surface reconstruction in Table XIV. 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 meaningful signed distance fields without R. Moreover, we found the λ weighting R slightly affects our performance.

3) 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 in Table XV. 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.

4) 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 in Table XVI. We learn an SDF from a single noisy point cloud. With more noises, our method can achieve better performance in all the three tasks.

5) One Observation versus Multiple Observations: Since our 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. XVII indicates that there is almost no performance difference with these two training settings.

6) Point Noise Level: Additionally, we visualize our results with larger noises which we use to learn an SDF in point cloud denoising in Fig. 23. 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%. Note that variances larger than 3 percent are not widely used in evaluations in previous studies.

7) Fast Learning Designs: We further conduct ablations for exploring the extra loss designs in our fast learning framework. As shown in Table XVIII, both the introduced pull loss in (9)

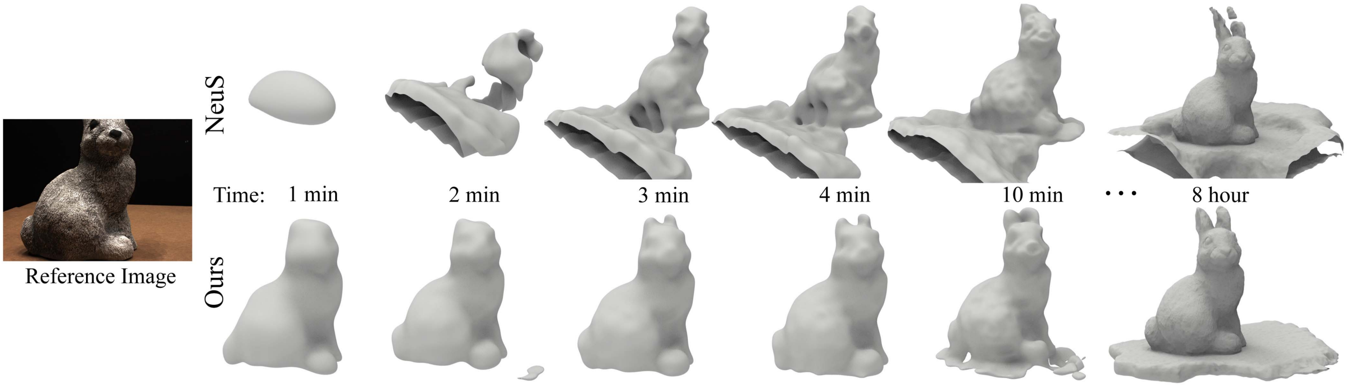

Fig. 24. Visual comparisons on multi-view reconstructions under different optimization time. The timing results do not include the time for Marching Cubes, and are reported under one single RTX3090 GPU.

and the Eikonal term in (10) contribute to the stable optimization of our fast learning framework based on multi-resolution hash grids. The designed pull loss helps the initializing of field in the areas far away from the surface and the Eikonal term stables the SDFs learning by regularizing the gradients.

The Eikonal term is necessary for our fast learning framework based on Instant-NGP [11], especially when learning SDFs from point clouds with large noise levels. We further provide more illustrations and visualizations on the reconstructions and distance fields of the SDFs learned by our framework with or without Eikonal term. The illustrative visualizations are shown in Fig. 21.

The visual comparisons in Fig. 21 demonstrate that the Eikonal term plays the key role in stabilizing and regularizing the SDF learning of our fast learning framework. The reconstructions achieve with Eikonal term is significantly smoother and more accurate than the reconstructions achieved without Eikonal term. We further provide the visualization of the distance fields learned with or without Eikonal term in the bottom of Fig. 21. The field visualization further demonstrates the effectiveness of Eikonal term in learning SDFs from noisy inputs. The level sets of the distance field learned with Eikonal term are much smoother than the ones of the distance field learned without Eikonal term.

8) Multi-View Reconstruction Designs: We then present visualizations of the multi-view reconstruction results under various loss settings to illustrate the effectiveness of our proposed designs. The visualization in Fig. 22 demonstrates that the introduced SDF prior loss L_{prior} in (12) significantly contributes to the reduction of artifacts in empty spaces and slightly improve the reconstruction quality. Conversely, the zero-level set loss L_{zls} defined in (13) has limited impact on removing artifacts but substantially helps to capture intricate geometries. The most optimal outcome is achieved through the combined utilization of both loss functions.

VII. 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 also integrate multi-resolution hash encoding implemented in CUDA into our framework, which reduces our training time by a factor of ten, enabling the convergence within one minute. Furthermore, we introduce a novel schema to improve multi-view reconstruction by using an SDF prior learned from noisy SfM points. 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.

References

[1] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, “Implicit geometric regularization for learning shapes,” in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 3789–3799.

[2] M. Atzmon and Y. Lipman, “SAL: Sign agnostic learning of shapes from raw data,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2562–2571.

[3] B. Ma, Z. Han, Y.-S. Liu, and M. Zwicker, “Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces,” in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 7246–7257.

[4] C. Jiang, A. Sud, A. Makadia, J. Huang, M. Nießner, and T. Funkhouser, “Local implicit grid representations for 3D scenes,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6000–6009.

[5] S. Peng, C. M. Jiang, Y. Liao, M. Niemeyer, M. Pollefeys, and A. Geiger, “Shape as points: A differentiable poisson solver,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 13032–13044.

[6] M. M. Kazhdan and H. Hoppe, “Screened poisson surface reconstruction,” *ACM Trans. Graph.*, vol. 32, no. 3, pp. 29:1–29:13, 2013.

[7] Y. Ohtake, A. G. Belyaev, M. Alexa, G. Turk, and H. Seidel, “Multi-level partition of unity implicits,” *ACM Trans. Graph.*, vol. 22, no. 3, pp. 463–470, 2003.

[8] S.-L. Liu, H.-X. Guo, H. Pan, P. Wang, X. Tong, and Y. Liu, “Deep implicit moving least-squares functions for 3D reconstruction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1788–1797.

[9] R. Chabra et al., “Deep local shapes: Learning local SDF priors for detailed 3D reconstruction,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 608–625.

[10] B. Ma, Y.-S. Liu, and Z. Han, "Learning signed distance functions from noisy 3D point clouds via noise to noise mapping," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 23338–23357.

[11] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," 2022, *arXiv:2201.05989*.

[12] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "NeuS: Learning neural implicit surfaces by volume rendering for multiview reconstruction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 27171–27183.

[13] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 4805–4815.

[14] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 405–421.

[15] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4104–4113.

[16] M. Oechsle, S. Peng, and A. Geiger, "UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 5569–5579.

[17] Z. Han, C. Chen, Y.-S. Liu, and M. Zwicker, "DRWR: A differentiable renderer without rendering for unsupervised 3D structure learning from silhouette images," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 3994–4005.

[18] C. Chen, Z. Han, Y.-S. Liu, and M. Zwicker, "Unsupervised learning of fine structure generation for 3D point clouds by 2D projections matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 12446–12457.

[19] C. Jin, T. Wu, and J. Zhou, "Multi-grid representation with field regularization for self-supervised surface reconstruction from point clouds," *Comput. Graph.*, vol. 114, pp. 379–386, 2023.

[20] P. Xiang et al., "SnowflakeNet: Point cloud completion by snowflake point deconvolution with skip-transformer," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 5479–5489.

[21] T. Takikawa et al., "Neural geometric level of detail: Real-time rendering with implicit 3D shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11358–11367.

[22] J. N. P. Martel, D. B. Lindell, C. Z. Lin, E. R. Chan, M. Monteiro, and G. Wetzstein, "ACORN: Adaptive coordinate networks for neural scene representation," 2021, *arXiv:2105.02788*.

[23] K. Rematas, R. Martin-Brualla, and V. Ferrari, "Sharf: Shape-conditioned radiance fields from a single view," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8948–8958.

[24] W. Feng, J. Li, H. Cai, X. Luo, and J. Zhang, "Neural points: Point cloud representation with neural fields for arbitrary upsampling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18612–18621.

[25] S. Li, J. Zhou, B. Ma, Y.-S. Liu, and Z. Han, "NeAF: Learning neural angle fields for point normal estimation," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 1396–1404.

[26] S. Li, J. Zhou, B. Ma, Y.-S. Liu, and Z. Han, "Learning continuous implicit field with local distance indicator for arbitrary-scale point cloud upsampling," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 3181–3189.

[27] B. Ma, H. Deng, J. Zhou, Y.-S. Liu, T. Huang, and X. Wang, "GeoDream: Disentangling 2D and geometric priors for high-fidelity and consistent 3D generation," 2023, *arXiv:2311.17971*.

[28] C. Chen, Y.-S. Liu, and Z. Han, "GridPull: Towards scalability in learning implicit representations from 3D point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 18322–18334.

[29] C. H. Koneputugodage, Y. Ben-Shabat, and S. Gould, "Octree guided unoriented surface reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 16717–16726.

[30] M. Michalkiewicz, J. K. Pontes, D. Jack, M. Baktashmotlagh, and A. P. Eriksson, "Deep level sets: Implicit surface representations for 3D shape inference," 2019, *arXiv: 1901.06802*.

[31] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 165–174.

[32] A. Ouasfi and A. Boukhayma, "Few 'zero level set'-shot learning of shape signed distance functions in feature space," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 561–578.

[33] B. Ma, J. Zhou, Y.-S. Liu, and Z. Han, "Towards better gradient consistency for neural signed distance functions via level set alignment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 17724–17734.

[34] J. Huang, H.-X. Chen, and S.-M. Hu, "A neural Galerkin solver for accurate surface reconstruction," *ACM Trans. Graph.*, vol. 41, no. 6, pp. 1–16, 2022.

[35] R. Zhu et al., "Semi-signed prioritized neural fitting for surface reconstruction from unoriented point clouds," 2022, *arXiv:2206.06715*.

[36] R. Zhu et al., "SSP: Semi-signed prioritized neural fitting for surface reconstruction from unoriented point clouds," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2024, pp. 3769–3778.

[37] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4460–4470.

[38] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5939–5948.

[39] W. Wang, Q. Xu, D. Ceylan, R. Mech, and U. Neumann, "DISN: Deep implicit surface network for high-quality single-view 3D reconstruction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 490–500.

[40] J. Chibane, T. Alldieck, and G. Pons-Moll, "Implicit functions in feature space for 3D shape reconstruction and completion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6968–6979.

[41] K. Genova, F. Cole, D. Vlasic, A. Sarna, W. T. Freeman, and T. Funkhouser, "Learning shape templates with structured implicit functions," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 7153–7163.

[42] Z. Han, G. Qiao, Y.-S. Liu, and M. Zwicker, "SeqXY2SeqZ: Structure learning for 3D shapes by sequentially predicting 1D occupancy segments from 2D coordinates," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 607–625.

[43] F. Williams, T. Schneider, C. Silva, D. Zorin, J. Bruna, and D. Panozzo, "Deep geometric prior for surface reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10130–10139.

[44] M. Liu, X. Zhang, and H. Su, "Meshing point clouds with predicted intrinsic-extrinsic ratio guidance," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 68–84.

[45] Z. Mi, Y. Luo, and W. Tao, "SSRNet: Scalable 3D surface reconstruction network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 967–976.

[46] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *Comput. Graph.*, vol. 21, no. 4, pp. 163–169, 1987.

[47] M. Jia and M. Kyan, "Learning occupancy function from point clouds for surface reconstruction," 2020, *arXiv: 2010.11378*.

[48] P. Erler, P. Guerrero, S. Ohrhallinger, N. J. Mitra, and M. Wimmer, "Points2Surf: Learning implicit surfaces from point clouds," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 108–124.

[49] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 523–540.

[50] J. Tang, J. Lei, D. Xu, F. Ma, K. Jia, and L. Zhang, "SA-ConvONet: Sign-agnostic optimization of convolutional occupancy networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6484–6493.

[51] E. Tretschk, A. Tewari, V. Golyanik, M. Zollhöfer, C. Stoll, and C. Theobalt, "PatchNets: Patch-based generalizable deep implicit 3D shape representations," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 293–309.

[52] K. Genova, F. Cole, A. Sud, A. Sarna, and T. Funkhouser, "Local deep implicit functions for 3D shape," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4856–4865.

[53] T. Li, X. Wen, Y. Liu, H. Su, and Z. Han, "Learning deep implicit functions for 3D shapes with dynamic code clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12830–12840.

[54] A. Boulch and R. Marlet, "POCO: Point convolution for surface reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6302–6314.

[55] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, "Scene representation networks: Continuous 3D-structure-aware neural scene representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1119–1130.

[56] S. Liu, Y. Zhang, S. Peng, B. Shi, M. Pollefeys, and Z. Cui, "DIST: Rendering deep implicit signed distance function with differentiable sphere tracing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2016–2025.

[57] Y. Jiang, D. Ji, Z. Han, and M. Zwicker, "SDFDiff: Differentiable rendering of signed distance fields for 3D shape optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1248–1258.

[58] Y. Wu and Z. Sun, "DFR: Differentiable function rendering for learning 3D generation from images," *Comput. Graph. Forum*, vol. 39, no. 5, pp. 241–252, 2020.

[59] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3501–3512.

[60] C.-H. Lin, C. Wang, and S. Lucey, "SDF-SRN: Learning signed distance 3D object reconstruction from static images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 11453–11464.

[61] L. Yariv et al., "Multiview neural surface reconstruction by disentangling geometry and appearance," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 2492–2502.

[62] Q. Fu, Q. Xu, Y. Ong, and W. Tao, "Geo-Neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 3403–3416.

[63] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, and A. Geiger, "MonoSDF: Exploring monocular geometric cues for neural implicit surface reconstruction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 25018–25032.

[64] Y. Wang, I. Skorokhodov, and P. Wonka, "HF-NeuS: Improved surface reconstruction using high-frequency details," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 1966–1978.

[65] H. Huang, Y. Wu, J. Zhou, G. Gao, M. Gu, and Y.-S. Liu, "NeuSurf: On-surface priors for neural surface reconstruction from sparse input views," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 2312–2320.

[66] F. Darmon, B. Bascle, J.-C. Devaux, P. Monasse, and M. Aubry, "Improving neural implicit surfaces geometry with patch warping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6260–6269.

[67] J. Wang et al., "NeuRIS: Neural reconstruction of indoor scenes using normal priors," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 139–155.

[68] R. Zhu et al., "AdaFit: Rethinking learning-based normal estimation on point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6118–6127.

[69] J. Zhou, J. Wang, B. Ma, Y.-S. Liu, T. Huang, and X. Wang, "Uni3D: Exploring unified 3D representation at scale," in *Proc. Int. Conf. Learn. Representations*, 2024.

[70] W. Zhao, J. Lei, Y. Wen, J. Zhang, and K. Jia, "Sign-agnostic implicit learning of surface self-similarities for shape modeling and reconstruction from raw point clouds," 2020, *arXiv: 2012.07498*.

[71] M. Atzmon and Y. Lipman, "SALD: Sign agnostic learning with derivatives," in *Proc. Int. Conf. Learn. Representations*, 2021.

[72] X. Wen, J. Zhou, Y.-S. Liu, H. Su, Z. Dong, and Z. Han, "3D shape reconstruction from 2D images with disentangled attribute flow," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3803–3813.

[73] J. Zhou, B. Ma, W. Zhang, Y. Fang, Y.-S. Liu, and Z. Han, "Differentiable registration of images and LiDAR point clouds with voxelpoint-to-pixel matching," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, Art. no. 2227.

[74] Y. Ben-Shabat, C. H. Koneputugodage, and S. Gould, "DiGS: Divergence guided shape implicit neural representation for unoriented point clouds," 2021, *arXiv:2106.10811*.

[75] J. Zhou et al., "3D-OAE: Occlusion auto-encoders for self-supervised learning on point clouds," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 15416–15423.

[76] W. Yifan, S. Wu, C. Oztireli, and O. Sorkine-Hornung, "Iso-Points: Optimizing neural implicit surfaces with hybrid representations," 2020, *arXiv: 2012.06434*.

[77] Y. Ben-Shabat, C. Hewa Koneputugodage, and S. Gould, "DiGS: Divergence guided shape implicit neural representation for unoriented point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 19301–19310.

[78] J. Zhou et al., "Self-supervised point cloud representation learning with occlusion auto-encoder," 2022, *arXiv:2203.14084*.

[79] J. Chibane, A. Mir, and G. Pons-Moll, "Neural unsigned distance fields for implicit function learning," 2020, *arXiv: 2010.13938*.

[80] B. Ma, Y. Liu, and Z. Han, "Reconstructing surfaces for sparse point clouds with on-surface priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6305–6315.

[81] B. Ma, Y. Liu, M. Zwicker, and Z. Han, "Surface reconstruction from point clouds by learning predictive context priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6316–6327.

[82] C. Chen, Y.-S. Liu, and Z. Han, "Latent partition implicit with surface codes for 3D representation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 322–343.

[83] A. Pumarola, A. Sanakoyeu, L. Yariv, A. Thabet, and Y. Lipman, "VisCo grids: Surface reconstruction with viscosity and coarea grids," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 18060–18071.

[84] J. Zhou, B. Ma, L. Yu-Shen, F. Yi, and H. Zhizhong, "Learning consistency-aware unsigned distance functions progressively from raw point clouds," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 16481–16494.

[85] J. Zhou, W. Zhang, B. Ma, K. Shi, Y.-S. Liu, and Z. Han, "UDiFF: Generating conditional unsigned distance fields with optimal wavelet diffusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 21496–21506.

[86] J. Zhou, B. Ma, S. Li, Y.-S. Liu, Y. Fang, and Z. Han, "CAP-UDF: Learning unsigned distance functions progressively from raw point clouds with consistency-aware field optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 22, 2024, doi: 10.1109/TPAMI.2024.3392364.

[87] M. Rakotosaona, V. L. Barbera, P. Guerrero, N. J. Mitra, and M. Ovsjanikov, "PointCleanNet: Learning to denoise and remove outliers from dense point clouds," *Comput. Graph. Forum*, vol. 39, no. 1, pp. 185–203, 2020.

[88] F. Pistilli, G. Fracastoro, D. Valsesia, and E. Magli, "Learning graph-convolutional representations for point cloud denoising," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 103–118.

[89] P. H. Casajus, T. Ritschel, and T. Ropinski, "Total denoising: Unsupervised learning of 3D point cloud cleaning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 52–60.

[90] J. Lehtinen et al., "Noise2Noise: Learning image restoration without clean data," in *Proc. Int. Conf. Mach. Learn.*, J. G. Dy and A. Krause, Eds., 2018, pp. 2971–2980.

[91] S. Luo and W. Hu, "Differentiable manifold reconstruction for point cloud denoising," in *Proc. ACM Int. Conf. Multimedia*, 2020, pp. 1330–1338.

[92] S. Luo and W. Hu, "Score-based point cloud denoising," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 4583–4592.

[93] R. Cai et al., "Learning gradient fields for shape generation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 364–381.

[94] B. Ma, Z. Han, Y.-S. Liu, and M. Zwicker, "Neural-Pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces," 2020, *arXiv: 2011.13495*.

[95] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, "Implicit geometric regularization for learning shapes," 2020, *arXiv: 2002.10099*.

[96] Z. Li et al., "Neuralangelo: High-fidelity neural surface reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 8456–8465.

[97] L. Yu, X. Li, C. Fu, D. Cohen-Or, and P. Heng, "PU-net: Point cloud upsampling network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2790–2799.

[98] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 950–953, 2003.

[99] F. Cazals and M. Pouget, "Estimating differential quantities using polynomial fitting of osculating jets," *Comput. Aided Geometry Des.*, vol. 22, pp. 121–146, 2005.

[100] E. Mattei and A. Castrodad, "Point cloud denoising via moving RPCA," *Comput. Graph. Forum*, vol. 36, no. 8, pp. 123–137, 2017, doi: 10.1111/cgf.13068.

[101] J. Zeng, G. Cheung, M. Ng, J. Pang, and C. Yang, "3D point cloud denoising using graph Laplacian regularization of a low dimensional manifold model," *IEEE Trans. Image Process.*, vol. 29, pp. 3474–3489, 2020.

[102] A. X. Chang et al., "ShapeNet: An information-rich 3D model repository," Stanford University – Princeton University – Toyota Technological Institute at Chicago, 2015, *arXiv:1512.03012*.

[103] S. Peng, M. Niemeyer, L. M. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 523–540.

[104] M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, and T. Brox, "What do single-view 3D reconstruction networks learn?," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3405–3414.

[105] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3D object reconstruction from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2463–2471.

[106] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3D-r2n2: A unified approach for single and multi-view 3D object reconstruction," in *Proc. Eur. Conf. Comput. Vis.*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., 2016, pp. 628–644.

[107] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mâché approach to learning 3D surface generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 216–224.

[108] P.-S. Wang, Y. Liu, and X. Tong, “Deep octree-based CNNs with output-guided skip connections for 3D shape and scene completion,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 266–267.

[109] F. Bogo, J. Romero, G. Pons-Moll, and M. J. Black, “Dynamic FAUST: Registering human bodies in motion,” in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2017, pp. 5573–5582.

[110] M. Berger, J. A. Levine, L. G. Nonato, G. Taubin, and C. T. Silva, “A benchmark for surface reconstruction,” *ACM Trans. Graph.*, vol. 32, no. 2, pp. 1–17, 2013.

[111] R. Hanocka, G. Metzer, R. Giryes, and D. Cohen-Or, “Point2Mesh: A self-prior for deformable meshes,” *ACM Trans. Graph.*, vol. 39, no. 4, 2020, Art. no. 126.

[112] C. Mercier, T. Lescoat, P. Roussillon, T. Boubekeur, and J.-M. Thiery, “Moving level-of-detail surfaces,” *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–10, 2022.

[113] Z. Wang et al., “Neural-IMLS: Self-supervised implicit moving least-squares network for surface reconstruction,” 2021, *arXiv:2109.04398*.

[114] Q. Zhou and V. Koltun, “Dense scene reconstruction with points of interest,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 112:1–112:8, 2013, doi: 10.1145/2461912.2461919.

[115] A. Boulch and R. Marlet, “POCO: Point convolution for surface reconstruction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6292–6304.

[116] Z. Wang et al., “ALTO: Alternating latent topologies for implicit 3D reconstruction,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 259–270.

[117] S. Li, G. Gao, Y. Liu, Y.-S. Liu, and M. Gu, “GridFormer: Point-grid transformer for surface reconstruction,” 2024, *arXiv:2401.02292*.

[118] J. Huang, Z. Gojcic, M. Atzmon, O. Litany, S. Fidler, and F. Williams, “Neural kernel surface reconstruction,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 4369–4379.

[119] A. Serna, B. Marcotegui, F. Goulette, and J. Deschaud, “Paris-rue-madame database - A 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods,” in *Proc. Int. Conf. Pattern Recognit. Appl. Methods*, 2014, pp. 819–824.

[120] B. Cai, J. Huang, R. Jia, C. Lv, and H. Fu, “NeuDA: Neural deformable anchor for high-fidelity implicit surface reconstruction,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 8476–8485.

[121] Y.-T. Liu et al., “NeUDF: Leaning neural unsigned distance fields with volume rendering,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 237–247.

[122] J. Zhou, B. Ma, S. Li, Y.-S. Liu, and Z. Han, “Learning a more continuous zero level set in unsigned distance fields through level set projection,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 3158–3169.

[123] X. Long et al., “NeuralUDF: Learning unsigned distance fields for multi-view reconstruction of surfaces with arbitrary topologies,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 20834–20843.

[124] R. A. Rosu and S. Behnke, “PermutoSDF: Fast multi-view reconstruction with implicit surfaces using permutohedral lattices,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 8466–8475.

[125] P. Xiang et al., “Snowflake point deconvolution for point cloud completion and generation with skip-transformer,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 6320–6338, May 2023.

[126] X. Wen et al., “PMP-Net: Point cloud completion by transformer-enhanced multi-step point moving paths,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 852–867, Jan. 2023.

[127] W. Zhang, R. Xing, Y. Zeng, Y.-S. Liu, K. Shi, and Z. Han, “Fast learning radiance fields by shooting much fewer rays,” *IEEE Trans. Image Process.*, vol. 32, pp. 2703–2718, 2023.

[128] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs, “Large scale multi-view stereopsis evaluation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 406–413.

[129] T. Wu et al., “Voxurf: Voxel-based efficient and accurate neural surface reconstruction,” 2022, *arXiv:2208.12697*.

[130] Y. Wang, Q. Han, M. Habermann, K. Daniilidis, C. Theobalt, and L. Liu, “NeuS2: Fast learning of neural implicit surfaces for multi-view reconstruction,” 2022, *arXiv:2212.05231*.

[131] F. Zhao et al., “Human performance modeling and rendering via neural animated mesh,” *ACM Trans. Graph.*, vol. 41, no. 6, pp. 1–17, 2022.

Junsheng Zhou received the BS degree in software engineering from Xiamen University, China, in 2021, and the MS degree from the School of Software, Tsinghua University, Beijing, China, in 2024. He is currently working toward the PhD degree with the School of Software, Tsinghua University, Beijing, China. His research interests include deep learning, 3D shape analysis and 3D reconstruction.

Baorui Ma received the BS degree in computer science and technology from Jilin University, China, in 2018, and the PhD degree from the School of Software, Tsinghua University, Beijing, China, in 2023. He is currently a researcher with the Beijing Academy of Artificial Intelligence, BAAI. His research interests include deep learning and 3D reconstruction.

Yu-Shen Liu (Member, IEEE) received the BS degree in mathematics from Jilin University, China, in 2000, and the PhD degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2006. From 2006 to 2009, he was a post-doctoral researcher with Purdue University. He is currently an associate professor with the School of Software, Tsinghua University. His research interests include shape analysis, pattern recognition, machine learning, and semantic search.

Zhizhong Han received the PhD degree from Northwestern Polytechnical University, China, 2017. He was a post-doctoral researcher with the Department of Computer Science, University of Maryland, College Park. Currently, he is an assistant professor of computer science with Wayne State University. His research interests include 3D computer vision, digital geometry processing and artificial intelligence.