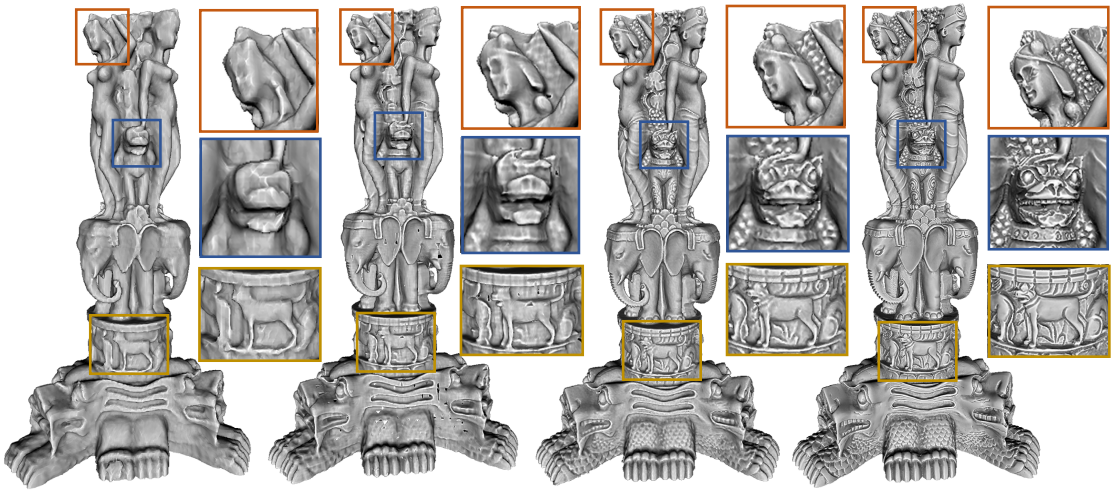
Learning Local Pattern-specific Deep Implicit Function for 3D Objects and Scenes

Anonymous CVPR submission

Paper ID 1449



(a) Global DIF (b) Local DIF (c) Ours (d) Reference

Figure 1. Visual comparison of 3D shape surface reconstruction. Compared with global DIF (e.g. [26]) and local DIF (e.g. [1]), our method can reconstruct the shape with fine-grained geometric details. Compared with previous methods that treat all local regions equally using a single decoder, we regard a shape as clusters of local regions and mine local patterns with different decoders. This alleviates the difficulty of learning caused by diverse local regions.

Abstract

Deep Implicit Function (DIF) has gained much popularity as an efficient 3D shape representation. To capture geometry details, current mainstream methods divide 3D shapes into local regions and then learn each one with a local latent code via a decoder. Such local methods can capture more local details due to less diversity among local regions than global shapes. Although the diversity of local regions has been decreased compared to global approaches, the diversity in different local regions still poses a challenge in learning an implicit function when treating all regions equally using only a single decoder. What is worse, these local regions often exhibit imbalanced distributions, where certain regions have significantly fewer observations. This leads that fine geometry details could not be preserved well. To solve this problem, we propose a novel Local Pattern-specific Implicit Function, named LP-DIF, to represent a shape with clusters of local regions and multiple decoders, where each decoder only focuses on one cluster of local regions which share a certain pattern. Specifically, we first extract local codes for all regions, and then cluster them into multiple groups in the latent space, where similar regions sharing a common pattern fall into one group. After that, we train multiple decoders for mining local patterns of different groups, which simplifies the learning of fine geometric details by reducing the diversity of local regions seen by each decoder. To further alleviate the data-imbalance problem, we introduce a region re-weighting module to each pattern-specific decoder using a kernel density estimator, which dynamically re-weights the regions during learning. Our LP-DIF can restore more geometry details, and thus improve the quality of 3D reconstruction. Experiments demonstrate that our method can achieve the state-of-the-art performance over previous methods.

1. Introduction

Representing 3D shapes is a fundamental problem for many applications in 3D computer vision. Recently, Deep Implicit Function (DIF) [4, 23, 26] has gained popularity for efficiently learning the representation of 3D objects and scenes. In contrast to directly learning explicit 3D representations [15,28,29] (voxels, point clouds or meshes), DIF aims to train a neural network to learn the binary occupancy function [23] or signed distance function (SDF) [26], as given a query location and an input latent code. Such kind of representation is continuous with arbitrary precision and can handle various topology, which has achieved the state-of-the-art results in several shape reconstruction tasks.

Existing DIF methods can be roughly classified into two categories: global and local approaches. Most of the early methods [4, 9, 20, 21, 23, 25–27, 32, 34, 37] fall into global approaches. These methods take advantage of one latent code and a single decoder to represent the whole shape. Global approaches often suffer from long training time and low reconstruction accuracy due to the limited capacity of capturing local geometry details. More recently, local approaches [1,3,5,6,10–12,17,19,31,33,35] divide 3D shapes (often divide by 3D grids) into local regions and then learn each one with a local latent code via a decoder, where the decoder shares the geometric similarities among different local regions. Although such local approaches can capture some local details, a large diversity of different local regions still increase the difficulty of learning an implicit function when treating all regions equally using only a single decoder. In addition, these local regions often exhibit imbalanced distributions, where certain regions have significantly fewer observations, especially in scenes. As a result, fine geometry details of shapes could not be captured well.

To address the above-mentioned problems, we propose a novel Local Pattern-specific Implicit Function, named LP-DIF, for learning 3D shape representation using clusters of local regions with multiple decoders, where each decoder only represents one cluster of local regions which share a certain pattern (geometric features such as facing direction, number of faces, relative positions to the region center). Specifically, we first extract the local latent codes for all local regions divided by 3D grids, and then cluster them into multiple groups in the latent space, where similar regions sharing a common pattern fall into one group. After that, we train a separate pattern-specific decoder for each group of regions, which reduces data-imbalance among different patterns of regions and simplifies the learning of fine geometric details of 3D structures by limiting the diversity of regions seen to each decoder. To further alleviate the region-imbalance problem, we introduce a region re-weighting module to each pattern-specific decoder by kernel density estimator, which dynamically re-weights the regions during learning. Our main contributions can be summarized as follows.

- We propose a novel LP-DIF to learn local pattern-specific deep implicit function of 3D shapes for reconstructing highly detailed geometry. Compared with previous methods that treat all local regions equally using a single decoder, we regard a shape as clusters of local regions and mine local patterns with different decoders. This alleviates the difficulty of learning caused by diverse local regions.

- We introduce a dynamic region re-weighting module, which could provide more focus on less common regions to tackle the data-imbalance problem in each pattern decoder. As a result, the regions with less appearances can be captured more accurately.

- Our method could be applied in multiple objects, single complex objects and large scale of scenes. We improve the state-of-the-art accuracy in surface reconstruction under various benchmarks.

Figure 2 illustrates the main differences between DIF, local DIF and our method. For DIF approaches, one global code and a decoder are used for the whole shape. For Local DIF methods, multiple local codes and a shared decoder are used. For our method, multiple clusters of regions are learned with different decoders.

2. Related Work

In computer vision and graphics, there are two categories of shape representations: explicit representation and implicit representation. Explicit geometric representations [15, 28, 29] such as point clouds, voxels and triangular meshes have been widely used for representing geometries for their simplicity and flexibility. More recently, deep implicit representations [4, 23, 26, 36] have been proposed in the context of shape representation, where the implicit surfaces of geometries are represented as the zero-set of spatial functions with fully connected neural networks. Such kind of representation is continuous with arbitrary precision and can handle different topology structure, which has achieved the state-of-the-art results in several shape reconstruction task. Implicit representations can be categorized into global and local approaches.

Global Deep Implicit Shape Representation. In recent years, implicit functions have been introduced into neural networks [23, 26] and show promising results. Early approaches learn one latent code and one single global implicit network to represent the whole shape. For example, DeepSDF [26] learned an implicit function where the network output represents the signed distance of the input point to its nearest surface, where zero-set of the learned function

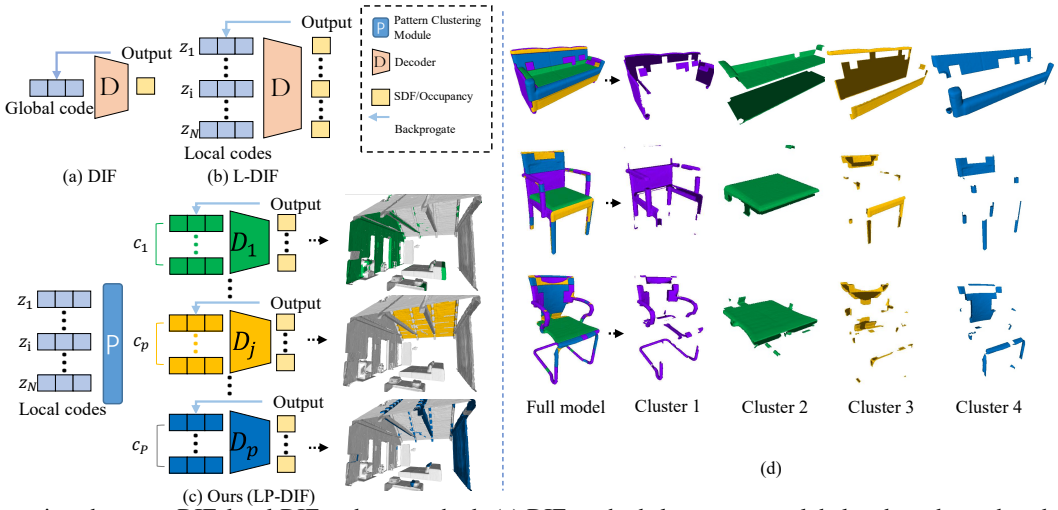


Figure 2. Comparison between DIF, local DIF and our method. (a) DIF methods leverage one global code and one decoder, often suffer from long training speed and low reconstruction quality due to the limited capacity of capturing local geometry details. (b) Local DIF methods use multiple local codes and one shared decoder, a large diversity of different local regions still increase the difficulty of learning an implicit function when treating all regions equally using only a single decoder. (c) Our method, LP-DIF, for representing a shape with some clusters of local regions and multiple decoders, where each decoder only represents one cluster of local regions which share a certain pattern. (d) Visualization of the clustered local regions.

implicitly represents the surface of the shape. Other approaches [23, 27] defined the implicit functions as 3D occupancy probability functions and turned shape representation into a point classification problem. Following these methods, [32, 34] improved the expressiveness of the implicit functions by introducing high frequency signals with periodic activation. [36] proposed to represent the volume density as an implicit function of the signed distance in neural volume rendering for high-quality novel view generation. However, these methods suffer from long training time and low reconstruction accuracy for high resolution data due to the limited capacity of decoder to capture local geometry details.

Local Deep Implicit Shape Representation. To capture more geometry details, current mainstream methods divide 3D shapes into many local regions and then learn each one with a local latent code via a decoder. Some recent works [1, 27] proposed to divide the whole 3D shape uniformly into 3D grids, where they either assign each local grid with a latent code or trilinearly interpolate local latent codes based on querying location. In LIG [17], this idea was applied to 3D scenes. NGLOD [33] leveraged sparse octree instead of uniform grids to increase efficiency. ACORN [22] proposed to adaptively optimize the coordinate decomposition to achieve higher accuracy in large-scale scenes. DCC-DIF [19] optimized the local latent code learning by introducing dynamic grid positions. IFNet [6] utilized multiple level of latent codes in a hierarchical way. LPI [3] divided the global shape into multiple local parts in latent space to allow easier shape parts learning and blending. PatchNet [35] was proposed to divide a shape into local patches with no fixed structure. Points2surf [10] incorporated local information surrounding each point, but it typically takes a lot of time for point-wise distance calculation. Although such local methods can capture some local details, a large diversity of different local regions still poses an challenge for learning an implicit function when treating all regions equally using only a single decoder. In addition, these local regions often exhibit imbalanced distributions, where certain regions have significantly fewer observations. As a result, fine geometry details could not be captured well. Several methods [8, 13] have been proposed to learn multiple decoders for point deformations, however, their methods cannot be applied in learning implicit representations for shapes. MDIF [5] used multiple levels of decoders in a hierarchical way, however, it did not consider the relationship between local regions. In novel view synthesizing, KiloNeRF [30] utilized thousands of tiny MLPs to represent different parts of the scenes. The simple and effective way to introduce multiple decoders is to assign each local region with a decoder, however, it is normally not feasible due to time and space constraint and the prior information of the local regions could not be preserved. In contrast, we introduce clustering for the local regions as a trade-off between reconstruction accuracy and efficiency.

3. Method

Given K discrete points $\mathbf{X} = \{\mathbf{x}_k\}$ and their signed distances $\mathbf{S} = \{\mathbf{s}_k\}$ from a 3D surface, our goal is to regress a continuous function that outputs the closest signed distance

to the 3D surface given a spatial point $\mathbf{x}$. To achieve this goal, we propose to learn a local pattern-specific DIF. The architecture of LP-DIF and differences between our LP-DIF and other methods are illustrated in Figure 2 . In this section, we first introduce latent embedding learning and local region feature extraction in Sec. 3.1. Then we describe our pattern clustering module in Sec. 3.2. After that we present our pattern decoder with density adaptive re-weighting in Sec. 3.3. At last, we illustrate our training process in Sec. 3.4.

3.1. Learning Latent Embedding for Local Regions

At the beginning, we choose to train a local DIF network [1, 17] to learn latent embedding for local regions to extract coarse local features. Local DIF [1, 17] is a special category of DIF, which is defined as a signed distance function $f(\mathbf{x}, \mathbf{z})$ modeled with several Multilayer Perceptrons (MLPs), mapping each query point $\mathbf{x}$ near a surface to the signed distance domain continuously with condition code $\mathbf{z}$. A shape $\mathcal{S}$ is defined as the zero level set of $f(\mathbf{x}, \mathbf{z})$:

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 | f_\theta(\mathbf{x}, \mathbf{z}) = 0\}, \tag{1}$$

where θ is the parameter of the MLPs. To be able to represent multiple shapes in the same time, the condition $\mathbf{z}$ is optimized for each shape, and the parameter θ is shared for all the shapes. Local DIF is defined on partitioned space of shape S. A normalized 3D space is usually partitioned into $d \times d \times d$ regions $\{\mathbf{B}_i\} \subseteq \mathbb{R}^3, i = 0...N-1$. All the regions have their separate codes $\mathbf{z}_i$, sharing the same parameter θ. The entire surface is defined as:

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 | \sum f_\theta(T(\mathbf{x}), \mathbf{z}_i) = 0\}, \tag{2}$$

where $T(\mathbf{x})$ is the re-centering function of $\mathbf{x}$, which transforms the points in each region to their local coordinates. After training of local DIF, all local regions have their codes $\mathbf{z}_i$, which serve as the local features of corresponding regions.

Code constraint. For DIF methods, it is important to constrain the latent codes of all the regions during training. Traditional Local DIF methods [1, 26] use L2 norm loss on the learned latent codes. However, in our method, we found that L2 norm loss could lead to latent codes being spread randomly in space, which is not good for subsequent clustering and latent code learning. Therefore, we constrain all local latent codes to be distributed within a hyper-sphere. The intuition is to process the latent codes in a way similar to normalizing them before entering the network. We update $\mathbf{z}$ according to Eq. (3) after every step of learning of $\mathbf{z}$.

$$\mathbf{z}_i = \begin{cases} \mathbf{z}_i & if \quad ||\mathbf{z}_i|| \leq 1 \\ \frac{\mathbf{z}_i}{||\mathbf{z}_i||} & \text{otherwise} \end{cases} \tag{3}$$

3.2. Pattern Clustering

After training of the local DIF, each region from the whole set of shapes will be encoded by the condition $\mathbf{z_{ij}}$ for region $\mathbf{B}_i$ and shape $S_j, j = 0...M-1$. A shape normally consists of many empty or near-empty regions. To alleviate the influence of these regions, we first remove regions with SDF positive rate (the percentage of positive SDF samples within one local region) out of $[0.01, 0.99]$. The intuition is that if one region has almost all SDF samples to be positive or negative, this region will be far away from any shape surfaces and need no training. After learning of local DIF, there are regions $\mathbf{B}_i$ that are learned well with low reconstruction errors, we remove these easy-to-learn regions with error threshold ϵ. The remaining regions are aggregated into P clusters by finding the cluster center $\mathbf{c}_k$ of their condition codes with K-means by optimizing 4. We empirically select $P = 4$ to balance between training speed and accuracy. The clustered local regions are visualized in Figure 2 (d).

$$\underset{z}{\operatorname{argmin}} \sum_{p=0}^{P} \sum_{i=0}^{N} \sum_{j=0}^{M} ||\mathbf{z_{ij}} - \mathbf{c_k}||^2. \tag{4}$$

Region border consistency. We found that the division of shape and clustering lead to inconsistent surface estimates near the boundaries of local regions. To alleviate this problem, we utilize overlap between close local regions. The $f_\theta(T(\mathbf{x}, \mathbf{z}_i))$ near the boundaries (distance less than 0.05) will be averaged by the nearby local regions.

3.3. Pattern Decoder

We have multiple pattern decoders to cover each cluster. The structure of each pattern decoder consists of 9 MLPs and a skip connection. Specifically, local code and query point coordinate are concatenated and fed into MLPs, and point coordinate is concatenated again after 4 MLPs. The output of the network is signed distance to the nearest surface.

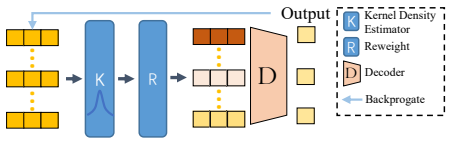


Figure 3. Pattern decoder with density-adaptive re-weighting. To further alleviate the region-imbalance problem, we introduce a region re-weighting module to each pattern-specific decoder by kernel density estimator, which dynamically re-weights the regions during learning.

Density-adaptive re-weighting. Among the learned local codes, certain regions may have less appearances than others, and therefore they are usually neglected by the decoder. To recover more details, it is important to focus more

on these regions. First we estimate the space density D of local codes based on their density distributions using kernel density estimation. Gaussian kernel is selected with bandwidth $h = 0.35$. Now we can re-weight region $\mathbf{B}_i$ with weight $\omega_i = \frac{1}{D}$ during loss calculation. There are two ways of re-weighting the regions, fixed and dynamic. In fixed way, we only update the weight once with pre-extracted local codes. In the dynamic way, the re-weighting is applied every time after local codes being updated by the pattern decoder.

3.4. Training

Our LP-DIF is trained in three stages. In the first stage, we extract the features from local regions by learning a coarse local DIF. In the second stage, we cluster the local regions based on their local features. At last, we train multiple pattern decoders on clustered regions. Note that for each pattern decoder, we use simple decoders so we have similar total number of decoder parameters with local DIF. During training, we optimize decoders and latent codes, while during testing, like other auto-decoder methods, we fix all decoders and only optimize latent codes.

Point sampling. We sample data pairs $(\mathbf{x}_k, \mathbf{s}_k)$ from $\mathcal{S}_j$ as supervision. $\mathcal{S}_j$ is a mesh or point cloud with normals $\mathbf{n}_k$. First we sample points along normal directions with normal distribution $N(0, \theta)$. Then we sample random points $\{\mathbf{x}_k\}$ and calculate their signed distances $\{\mathbf{s}_k\}$ based on the closest distances.

Smooth L1 loss. To increase the weight in regions with larger errors, we introduce smooth L1 loss [24], defined by

$$L_k^{sdf} = \begin{cases} \frac{1}{2}(f_\theta(\mathbf{x_k}) - \mathbf{s_k})^2 & if\ |(f_\theta(\mathbf{x_k}) - \mathbf{s_k})| < 1 \\ ((f_\theta(\mathbf{x_k}) - \mathbf{s_k}) - \frac{1}{2}) & \text{otherwise} \end{cases} \tag{5}$$

Gradient loss. It is important to train a DIF with gradient consistency. We apply constraint on the gradient of each sampled points, denoted as

$$L_k^{grad}(\mathbf{x_k}) = e^{-(100\mathbf{s_k})^2}(1 - \frac{\langle \nabla f(\mathbf{x_k}), \mathbf{n_k} \rangle}{||f(\mathbf{x_k})||\ ||\mathbf{n_k}||}), \tag{6}$$

We use cosine similarity of $\mathbf{x_k}$ and the normal of its nearest sampled surface point, and make $\mathbf{x_k}$ closer to the surface have larger influence.

The total loss function could be formulated as

$$L = L_{sdf} + \lambda L_{grad}, \tag{7}$$

where $\lambda = 1.0$ for all our experiments.

4. Experiments

We evaluate the effectiveness of our method compared with the state-of-the-art methods on several different aspects: (1) the representation performance on 3D objects dataset, (2) the performance on large-scale synthetic and real scenes data, and (3) reconstruction power on single complex object reconstruction.

4.1. 3D Objects Dataset Reconstruction

Dataset and metric. We run the experiments on the ShapeNet dataset [2] with the same settings with [5]. The dataset contains a subset of 13 categories in ShapeNet with train/test splits from [7]. We evaluate geometric reconstruction quality with L2 Chamfer Distance (CD) and F-Score. For L2 Chamfer Distance, we take exactly the same settings as [5]. We sampled 100K signed distances near the surface and 100K signed distances uniformly around the space. We estimate CD using 100,000 randomly sampled points on the ground truth and the reconstructed meshes. Additionally, in all experiments, we compute the F-Score at a threshold of τ , as F-Score is a metric less sensitive to outliers. F-Score is the mean of recall (percentage of reconstruction to target distances under τ) and precision (vice versa). For object reconstruction we use τ = 0.01. To be consistent with [5], we divide the whole space into $16 \times 16 \times 16$ regions. We also compare with NGLOD [33] in ShapeNet150 dataset following their evaluation protocol.

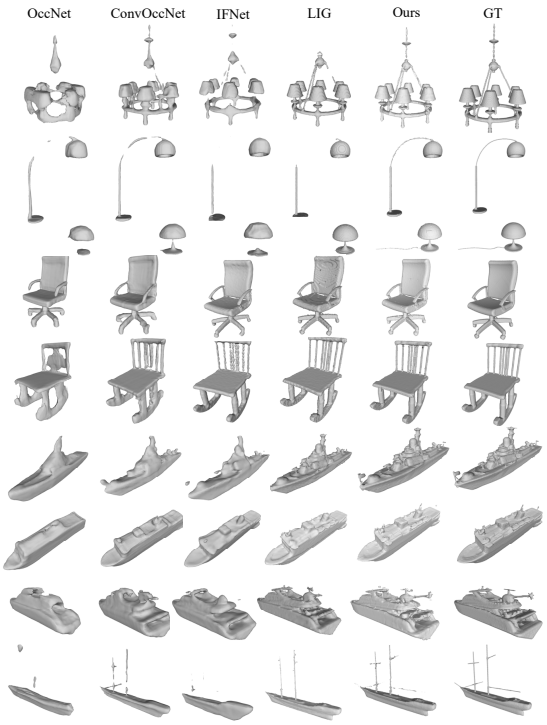


Figure 4. Visual comparison with OccNet [23], ConvOccNet [27], IFNet [6], and LIG [17], our method could reconstruct more geometry detail on ShapeNet.

Quantitative comparison. The comparison results between LP-DIF and other state-of-the-art methods are shown

Category	CD Mean ↓ Occ. [27]	IMNet [4]	L-DIF [1]	MDIF [5]	DCC-DIF [19]	Ours	F-score ↑ Occ [27]	IMNet [4]	L.DIF [1]	MDIF [5]	DCC-DIF [19]	Ours
airplane	0.25	0.13	0.044	0.028	**0.011**	0.0115	89.8	91.7	98.5	98.6	99.7	**99.86**
bench	0.34	0.22	0.121	0.052	0.017	**0.0211**	85.2	88.6	96.0	96.0	99.5	**99.64**
cabinet	0.32	0.23	0.063	0.051	0.131	**0.0405**	83.2	89.2	96.6	96.6	96.4	**98.42**
car	0.58	0.26	0.09	0.088	0.218	**0.0332**	69.3	82.7	93.1	93.0	92.7	**98.84**
chair	0.38	0.43	0.042	0.035	0.037	**0.0294**	80.2	82.5	97.7	97.6	99.1	**99.43**
display	0.35	0.20	0.043	**0.019**	0.028	0.0256	82.3	89.4	98.6	98.7	99.4	**99.64**
lamp	1.47	2.76	0.795	0.795	0.327	**0.0180**	62.9	73.8	93.5	93.5	97.3	**99.58**
rifle	0.39	0.55	0.060	0.057	0.007	**0.0062**	86.1	81.1	96.9	96.9	99.9	**99.97**
sofa	0.31	0.16	0.208	0.037	0.036	**0.0264**	85.2	89.3	98.3	98.4	99.1	**99.64**
speaker	0.38	0.17	0.065	0.044	0.146	**0.0424**	78.1	89.4	97.3	97.3	96.1	**97.86**
table	0.31	0.30	0.107	0.046	0.029	**0.0285**	87.2	88.6	96.5	97.6	99.3	**99.50**
telephone	0.19	0.11	0.043	0.027	**0.010**	0.0202	88.9	96.5	**99.6**	99.3	**99.6**	99.58
watercraft	0.35	0.39	0.075	0.067	0.042	**0.0156**	80.3	84.7	97.4	97.2	98.3	**99.70**
mean	0.43	0.46	0.135	0.102	0.081	**0.0245**	81.4	86.7	96.9	97.0	98.2	**99.36**

Table 1. Surface reconstruction comparison on test set of shapenet in terms of mean L2 chamfer distance (multiplied by 1e3) and F-score.

Methods	LOD3 [33]	LOD4 [33]	LOD5 [33]	LP-DIF
CD Mean ↓	0.112	0.069	0.062	**0.025**

Table 2. Comparison in ShapeNet150 with NGLOD [33].

Methods	Sofa	Lamp	Bookshelf
MDIF [5] (seen/unseen) (CD↓)	0.0370/ -	0.7950/ -	-/ 0.0910
LP-DIF (seen/unseen) (CD↓)	**0.0264**/ 0.0265	**0.0180**/ 0.0185	-/ **0.0490**

Table 3. Cross-category performance in ShapeNet.

in Table 1. LP-DIF achieves the best performance across all categories in terms of average L2 Chamfer Distance and F-score. Compared with the second best method MDIF (auto-decoder branch) [5], LP-DIF achieves the best performance in 11 out of 13 categories, which shows the representational power of our method for representing multiple objects. As in Table 2, our method performs better than the deepest level of [33]. To test the generalization ability of our method, We train our model on one category (chair) and tested on another (unseen). A subset of the cross-category performance is shown in Table 3. Our method has more accuracy than MDIF [5] in unseen categories (bookshelf). In addition, as local regions share similar feature across categories, the accuracy for seen and unseen categories is almost the same for our method.

Qualitative comparison. In Figure 4, we visually compared 3D surface reconstruction on Shapenet with other DIF methods [6, 17, 23, 27], from which we find that LP-DIF reconstructs more geometry details while other method tends to lose details on thin structures.

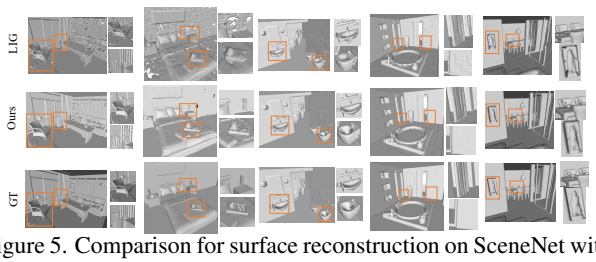


Figure 5. Comparison for surface reconstruction on SceneNet with sparse sampling.

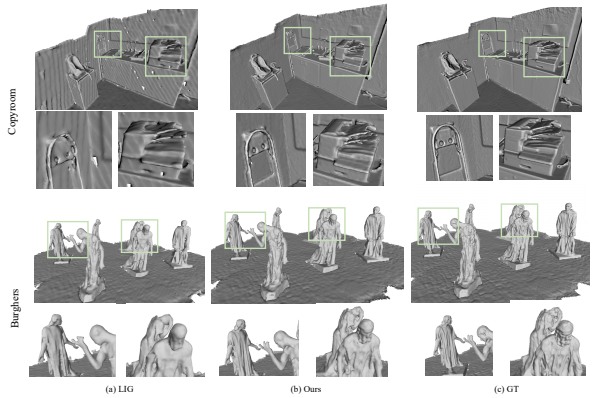


Figure 6. Comparison for surface reconstruction on 3D Scene dataset with dense sampling.

4.2. 3D Scene Reconstruction

Dataset and metric. We conduct our experiments on a synthetic dataset: SceneNet [14], and a real scanned dataset: 3D Scene dataset [38]. As scenes in SceneNet dataset are not watertight, and have many double-sided faces creating conflicting normals, we pre-process SceneNet to watertight using methods from [16]. We drop scenes with empty inner spaces after the watertighting process. We sample a constant density of points according to the area of faces (1000 points per m^2 for SceneNet and 10000 points per m^2 for 3D Scene dataset). For all the scene-level experiments, we randomly sample 2 million points on the generated mesh and the ground truth mesh when estimating L1 CD, and we use $\tau = 0.025$ (2.5cm) for F-score. Since data is provided in a physically-meaningful scale, we use world coordinate (meters) for computing CD. Note that these two metrics are not always consistent as CD is more sensitive to outliers. For scene-level experiments, we directly optimize latent codes and decoders on the scenes, and we divide the whole space into $32 \times 32 \times 32$ regions.

Quantitative comparison. The comparison results between LP-DIF and other state-of-the-art methods are shown in Table 4. LP-DIF achieves the best performance across

Category	CD Mean ↓ LIG [17]	Ours	F-score ↑ LIG [17]	Ours	Scene	CD Mean ↓ LIG [17]	Ours	F-score ↑ LIG [17]	Ours
Bathroom	0.00825	**0.00633**	98.56	**99.66**	Burghers	0.0122	**0.0039**	90.25	**99.67**
Bedroom	0.00978	**0.00730**	96.41	**99.24**	Lounge	0.0131	**0.0031**	85.49	**99.72**
Office	0.01452	**0.00929**	93.92	**96.97**	CopyRoom	0.0130	**0.0034**	84.68	**99.97**
Livingroom	0.01191	**0.00987**	96.39	**98.65**	StoneWall	0.0110	**0.0031**	89.14	**99.90**
Kitchen	0.00988	**0.00825**	96.36	**98.55**	TotemPole	0.0120	**0.0041**	89.39	**99.23**
Mean	0.01065	**0.00816**	96.40	**98.67**	Mean	0.0122	**0.0035**	87.79	**99.70**

Table 4. Surface reconstruction from sparse point cloud on SceneNet and dense point cloud on 3D SceneNet dataset in terms of mean L1 chamfer distance and F-score.

all categories in terms of average L1 Chamfer Distance and F-score both in sparse (Scenenet) and dense (3D SceneNet dataset) sampling settings, which proves better representational power of our method. Compared with scenenet with sparse sampling setting, our method improves more on 3D scenenet dataset with dense sampling settings.

Qualitative comparison. In Figure 5 and 6, we visually compared 3D surface reconstruction of scene dataset with other DIF methods, from which we find that LP-DIF reconstructs more geometry details and less missing parts than other method.

Methods	DeepSDF [26]	LIG [17]	L-DIF [1]	Ours
CD Mean ↓	0.740	0.603	0.211	**0.176**
F-score ↑	1.3	13.6	21.8	**63.6**

Table 5. Surface reconstruction comparison on Thai statue in terms of mean L1 chamfer distance and F-score.

4.3. Complex 3D Object Reconstruction

Dataset and metric. For complex 3D object reconstruction, we use Thai Statue, obtained from the Stanford 3D Scanning Repository [18]. It is an intricate statue, which consists of 10 million polygons. Fine details of the statue are challenging to be captured with existing methods. We randomly sample 2 million points on the generated and the ground truth meshes when estimating L1 CD, and we use τ = 0.025 for F-score. For complex 3D object, we directly optimize latent codes and decoders to fit the shape. For ACORN [22], we train using their official code and extract the reconstructed mesh. [22] is trained for several hours until max octants are reached.

Quantitative comparison. The comparison results between LP-DIF and other state-of-the-art methods are shown in Table 5. LP-DIF achieves the best performance in terms of average L1 Chamfer Distance and F-score for complex 3D object reconstruction.

Qualitative comparison. In Figure 1, we visually compare 3D surface reconstruction of Thai statue with other DIF methods, where we find that LP-DIF reconstructs much more fine geometry details than other methods. In Figure 7, we compare with ACORN [22]. Our result is comparable with ACORN with slightly better details.

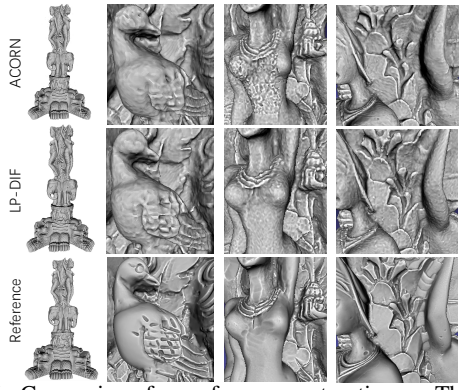


Figure 7. Comparison for surface reconstruction on Thai statue. Our result is comparable with ACORN [22] with slightly better details.

Methods	# Feature Param	# Network Param	Total # Param
LOD5 [33]	10122528	23685	10146213
LDIF [1]	524288	460993	985281
MDIF [5]	44032	18359301	18403333
LIG [17]	131072	227745	358817
Ours	53248 (coarse) + 249856	3745 (coarse) + 463108	769957

Table 6. Network capacity comparison.

Methods	# Feature Param	# Inference Param	Total # Param	CD (Airplane)↓	CD (Bench)↓
LDIF [1]	524288	460993	985281	0.044	0.121
Ours	303104 (53248 + 249856)	466853 (3745 + 463108)	769957	**0.012**	**0.021**
MDIF [5]	44032	18359301	18403333	0.028	0.052
Ours-abl-mdif	303104 (53248 + 249856)	17952293 (3745 + 17948548)	18255397	**0.0145**	**0.031**
Ours-coarse	53248	2689	55937	0.062	0.167

Table 7. Ablation with similar total number of parameters.

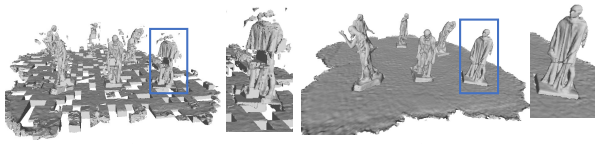
Methods	Stage1	Stage 2	Stage 3	Total
NGLOD [33]	-	-	-	~4000s
L-DIF	-	-	-	165s (1.33s $\times 124 epochs$)
LP-DIF	20s	1s	141s (2.43s $\times 58 epochs$)	**162s**

Table 8. Computation efficiency comparison.

4.4. Ablation Study

Network capacity and efficiency comparison. We have summarized the number of parameters used in NGLOD (LOD5) [33], LDIF [1], MDIF [5], LIG [17], and our method in Table 6. As in the table, different number of parameters are used and our method utilized the second least total number of parameters. We also tested the performance under the same total number of parameters in Table

7. As in the first two rows of the table, LDIF [1] and our method have the similar total number of parameters, where our method outperforms LDIF [1]. After that, we increase our MLP to 800-dimensional hidden layers to have the similar total number parameters with MDIF [5]. As in the 3-4 rows of the table, the performance of our ablation method is still better than MDIF [5], but is worse than our original decoders. The reason could be that the increased parameters of network makes the optimization of local latent code unstable. The performance of our coarse decoder is also shown in the last row for comparison. We have also compared the computation time on 8 shapes of ShapeNet in Table 8. It takes much more time for NGLOD [33] to train. For L-DIF and LP-DIF, we reported the time needed to reach the same accuracy level. It takes less time for LP-DIF than L-DIF.



(a) Random clustering (b) LP-DIF

Figure 8. Comparison with random clustering. (a) Replacing the clustering process with random assigning each local region into clusters. (b) Clustering with regards to similarity leads to better performance.

Methods	# of codes	code size	CD Mean ↓	F-score ↑
Baseline-32	32K	7.63 MB	0.00557	96.78
Baseline-128	2097K	488 MB	0.00389	99.65
LP-DIF-32	32K	7.63 MB	0.00394	**99.67**
LP-DIF-128	2097K	488 MB	**0.00387**	99.64

Table 9. Ablation study with different number of local codes. Result of the baseline method with different number of local codes are shown in the first two rows. Result of LP-DIF with different number of local codes are shown in the last two rows. Our method can achieve same level of accuracy with much less local codes.

Methods	CD Mean↓	F-score ↑
Baseline	0.00557	96.78
LP-DIF w/o clustering	0.00446	99.52
LP-DIF w/o re-weighting	0.00419	99.16
LP-DIF	**0.00394**	**99.67**

Table 10. Ablation study on components of our model in terms of mean L1 chamfer distance and F-score.

Ablation on modules. We also conduct a series of experiments on the effectiveness of our modules on Burghers of 3D scenenet dataset. Our baseline is implemented by removing pattern clustering and region re-weighting. First, we separate the whole space into different number of regions (each region has one code), and test the effect of our module under different settings. The results are shown in Table 9, where the baseline model performs better with more local regions and codes, however, more local codes occupy much more space. Our model with 7.63 MB local codes could have similar performance with the baseline method with 488 MB local codes. After that, we evaluate the effectiveness of our clustering module by replacing it with random assigning cluster indices. As in Figure 8 and Table 6, the performance is deteriorated with random clustering. The possible reason is that randomly separating local regions increases the deviation of some local regions from the cluster centers, causing them to fail in training. Next we evaluate the components of our model by removing them separately. We test two variations: (1) removing the clustering module and re-weight on the whole set of regions, and (2) removing the re-weighting module. The result is shown in Table 10 and Figure 9 (a) - (c), and we find that both modules are helpful for achieving good performance. At last, we test the difference between fixed density estimator and dynamic density estimator. As shown in Figure 9 (d) - (f), dynamic density estimator could reconstruct better geometry details than fixed one.

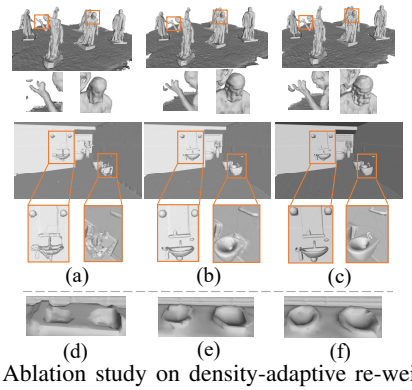


Figure 9. Ablation study on density-adaptive re-weighting. (a) Ours w/o density re-weighting. (b) Ours w/ density re-weighting. (c) Ground truth. (d) Ours w/o density estimator. (e) Ours w/ fixed density estimator. (f) Ours w/ dynamic density estimator.

5. Conclusion and Limitations

In this paper, we present LP-DIF, a local pattern-specific deep implicit function for reconstructing highly detailed geometry. Compared with previous methods that treat all local regions equally using a single decoder, we regard a shape as clusters of local regions and mine local patterns with different decoders. This alleviates the difficulty of learning caused by diverse local regions. In addition, we introduce a dynamic region re-weighting module, which could provide more focus on less common regions to tackle the data-imbalance problem in each pattern decoder. As a result, finer details of the local regions are learned more accurately. We demonstrate that our method outperforms the latest methods under various benchmarks.

Similar to the existing DIF methods, our approach also faces the challenges on 3D scenes with sparse sampling in thin structures. In 3D scenes with sparse sampling on thin structures, there may exist multiple incorrectly sampled SDFs near thin structures, which leads to the discontinuous reconstructed surface.

References

[1] Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local SDF priors for detailed 3d reconstruction. In *European Conference on Computer Vision*, pages 608–625. Springer, 2020. 1, 2, 3, 4, 6, 7, 8

[2] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 5

[3] Chao Chen, Yu-Shen Liu, and Zhizhong Han. Latent partition implicit with surface codes for 3d representation. *arXiv preprint arXiv:2207.08631*, 2022. 2, 3

[4] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 2, 6

[5] Zhang Chen, Yinda Zhang, Kyle Genova, Sean Fanello, Sofien Bouaziz, Christian Hane, Ruofei Du, Cem Keskin, Thomas Funkhouser, and Danhang Tang. Multiresolution deep implicit functions for 3d shape representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13087–13096, 2021. 2, 3, 5, 6, 7, 8

[6] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3D shape reconstruction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6970–6981, 2020. 2, 3, 5, 6

[7] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 5

[8] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Learning elementary structures for 3D shape generation and matching. *Advances in Neural Information Processing Systems*, 2019. 3

[9] Yueqi Duan, Haidong Zhu, He Wang, Li Yi, Ram Nevatia, and Leonidas J Guibas. Curriculum DeepSDF. In *European Conference on Computer Vision*, pages 51–67. Springer, 2020. 2

[10] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J Mitra, and Michael Wimmer. Points2Surf learning implicit surfaces from point clouds. In *European Conference on Computer Vision*, pages 108–124. Springer, 2020. 2, 3

[11] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4857–4866, 2020. 2

[12] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7154–7164, 2019. 2

[13] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3D surface generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 216–224, 2018. 3

[14] Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla. Understanding real world indoor scenes with synthetic data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4077–4085, 2016. 6

[15] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: a network with an edge. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 2

[16] Jingwei Huang, Hao Su, and Leonidas Guibas. Robust watertight manifold surface generation method for shapenet models. *arXiv preprint arXiv:1802.01698*, 2018. 6

[17] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3D scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020. 2, 3, 4, 5, 6, 7

[18] Stanford Computer Graphics Laboratory. Stanford 3D scanning repository, 2014. 7

[19] Tianyang Li, Xin Wen, Yu-Shen Liu, Hua Su, and Zhizhong Han. Learning deep implicit functions for 3d shapes with dynamic code clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12840–12850, 2022. 2, 3, 6

[20] Shi-Lin Liu, Hao-Xiang Guo, Hao Pan, Peng-Shuai Wang, Xin Tong, and Yang Liu. Deep implicit moving least-squares functions for 3D reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1788–1797, 2021. 2

[21] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. *The International Conference on Machine Learning*, 2020. 2

[22] Julien N. P. Martel, David B. Lindell, Connor Z. Lin, Eric R. Chan, Marco Monteiro, and Gordon Wetzstein. ACORN: Adaptive coordinate networks for neural scene representation. *ACM Trans. Graph. (SIGGRAPH)*, 40(4), 2021. 3, 7

[23] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 2, 3, 5, 6

[24] Gregory P Meyer. An alternative probabilistic interpretation of the huber loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5261–5269, 2021. 5

[25] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020. 2

[26] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 1, 2, 4, 7

[27] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020. 2, 3, 5, 6

[28] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 2

[29] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 2017. 2

[30] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14335–14345, 2021. 3

[31] Yawar Siddiqui, Justus Thies, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. RetrievalFuse: Neural 3D scene reconstruction with a database. *International Conference on Computer Vision*, 2021. 2

[32] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 3

[33] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3D shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11367, 2021. 2, 3, 5, 6, 7, 8

[34] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 2020. 2, 3

[35] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Carsten Stoll, and Christian Theobalt. Patchnets: Patch-based generalizable deep implicit 3D shape representations. In *European Conference on Computer Vision*, pages 293–309. Springer, 2020. 2, 3

[36] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34, 2021. 2, 3

[37] Zerong Zheng, Tao Yu, Qionghai Dai, and Yebin Liu. Deep implicit templates for 3D shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1429–1439, 2021. 2

[38] Qian-Yi Zhou and Vladlen Koltun. Dense scene reconstruction with points of interest. *ACM Transactions on Graphics (ToG)*, 32(4):1–8, 2013. 6