Neural Implicit 3D Shapes from Single Images with Spatial Patterns

Yixin Zhuang[1(✉)], Yujie Wang[2,3], Yunzhe Liu[3], and Baoquan Chen[3(✉)]

[1] Fuzhou University
[2] Shandong University
[3] Peking University
{yixin.zhuang,baoquan.chen}@gmail.com

Abstract. Neural implicit representations are highly effective for single-view 3D reconstruction (SVR). It represents 3D shapes as neural fields and conditions shape prediction on input image features. Image features can be less effective when significant variations of occlusions, views, and appearances exist from the image. To learn more robust features, we design a new feature encoding scheme that works in both image and shape space. Specifically, we present a geometry-aware 2D convolutional kernel to learn image appearance and view information along with geometric relations. The convolutional kernel operates at the 2D projections of a point-based 3D geometric structure, called *spatial pattern*. Furthermore, to enable the network to discover adaptive spatial patterns that capture non-local contexts, the kernel is devised to be deformable and exploited by a spatial pattern generator. Experimental results on both synthetic and real datasets demonstrate the superiority of the proposed method.

Keywords: Single Image 3D Reconstruction · Deformable Convolution · Implicit Neural Representation

1 Introduction

3D shape reconstruction from a single image has been one of the central problems in computer vision. Empowering the machines with the ability to perceive the imagery and infer the underlying 3D shapes can benefit various downstream tasks, such as augmented reality, robot navigation, etc. However, the problem is overly ambiguous and ill-posed and thus remains highly challenging due to information loss and occlusion.

In recent years, many deep learning methods have been proposed to infer 3D shapes from single images. These methods rely on learning shape priors from many shape collections and can reason the underlying shape of unseen images. To this end, various learning frameworks have been proposed that exploit different 3D shape representations, including point sets [7, 1], voxels [36, 37], polygonal

Yixin Zhuang and Yujie Wang contributed equally to this work.
The source code can be found at `https://github.com/yixin26/SVR-SP`.

meshes [10, 33], and implicit fields [4, 21, 24]. In particular, implicit field-based models have shown impressive performance compared to the others.

Implicit field-based networks take a set of 3D samplings as input and predict corresponding values under varying representations (e.g., occupancy, signed distance, etc.). Once fitted, 3D shapes are identified as the zero level of the predicted scalar fields using meshing methods such as Marching Cubes [19]. By conditioning the 3D shape generation on the extracted global feature of input image [21, 4], the implicit networks are well-suited to reconstruct 3D shapes from single images. However, this trivial combination often fails to reconstruct fine geometric details and produces overly smoothed surfaces.

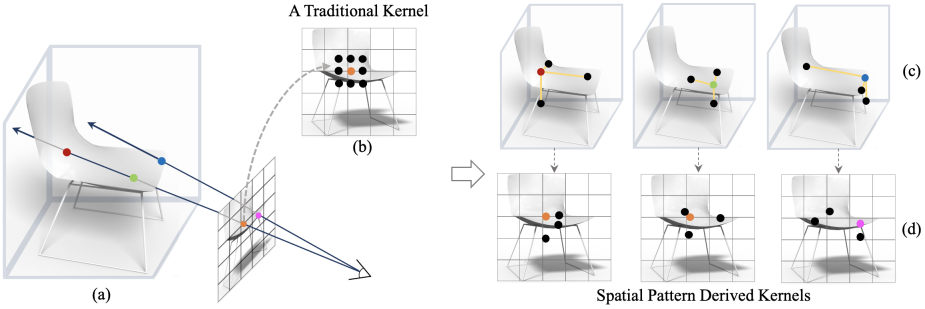


Fig. 1. Illustration of the pipeline of spatial pattern guided kernel. (a) shows that each 3D point sampling (colored differently) of the depicted shape is aligned to a 2D pixel by the given camera pose. Compared with a 2D convolution kernel (b) that only considers neighbors located within a 2D regular local patch, the kernels in (d) derived from the proposed spatial patterns (c) explicitly exploit the underlying geometric relations for each pixel. As a result, the kernels in (d) encode the local image features that capture both image appearance and point relations.

Toward pixel-level accurate reconstruction, DISN [40] proposes a pixel-aligned implicit surface network where individual point sampling is conditioned on a learned local image feature obtained by projecting the point to the image plane according to the camera pose. With local image features, the network predicts a residual field for refinement. Compared to those only acquiring global image features, local features enable the restoration of much finer-level geometry details. However, the strategy of associating 3D samplings with learned local image features would not have intuitive meaning when samples are occluded from the observation view. Hence, to improve the local image feature, Ladybird [41] utilizes the feature extracted from the 2D projection of its symmetric point obtained from the self-reflective symmetry position of the object. The reconstruction is significantly improved upon DISN. Nevertheless, the strategy used in Ladybird is not sufficiently generic as the feature probably would have no intuitive meaning in the situation where the symmetric points are non-visible or the symmetry assumption does not hold. Meanwhile, D^2IM-Net [14] samples training points

based on the scale of geometry feature and includes image laplacian for loss computation, both targets at sharp surface regions. D^{2}IM-Net significantly improves the visual quality but can still fail when the quality of image laplacian is low or dramatic self-occlusion happens. Therefore further exploration of local image features is needed in tackling those challenges.

In this paper, we introduce a new image feature encoding scheme, supported by spatial pattern, to achieve further exploitation of local image features. The spatial pattern may include geometric relationships, e.g., symmetric, co-planar, or other structures that are less intuitive. With the spatial pattern, a 2D kernel operating in image space is derived to encode local image features of 3D point samplings. Specifically, the pattern is formed by a fixed number of affinities around a 3D sampling, for which the corresponding 2D projections are utilized as the operation positions of the kernel. Although a traditional 2D convolution is possible to encode contextual information for the central point, it ignores the underlying geometric relations in the original 3D space between pixels and encounters the limitations brought by the regular local area. A 2D deformable kernel [6] is able to operate in irregular neighborhoods, but it is still not able to explicitly consider the underlying 3D geometric relations, which are important in 3D reconstruction tasks.

Figure 1 shows the pipeline of the 3D spatial pattern guided 2D kernel. As shown in Figure 1 (c-d), the kernels operate on points determined by spatial patterns for different point samplings. Specifically, the proposed kernel finds kernel points adaptively for each pixel, which considers its geometric-related positions (e.g., symmetry locations) in the underlying 3D space, rather than only relying upon the appearance information. Furthermore, the spatial pattern is devised to be deformable to enable the network to discover more adaptive geometric relations for point samplings. In the experiments section, we will explore the learned 3D spatial pattern with visualization and statistics.

To demonstrate the effectiveness of spatial pattern guided kernel, we integrate it into a network based on a deep implicit network [40], and extensively evaluate our model on the large collection of 3D shapes – the ShapeNet Core dataset [3] and Pix3D dataset [28]. The experiments show that our method can produce state-of-the-art 3D shape reconstruction results from single images compared to previous works. Ablation experiments and analyses are conducted to show the performance of different spatial pattern variants and the importance of individual points within the spatial pattern.

In this work, we make the following contributions.

- We present spatial patterns to provide the network with more flexibility to discover meaningful image features that explicitly consider the geometric relationships.
- We extend 2D deformable convolutional kernels with a 3D spatial pattern generator to learn meaningful geometric structures that are crucial for 3D shape reasoning.
- We perform extensive experiments on a real and a synthetic dataset to validate the effectiveness of learned spatial patterns. Our method consistently

outperforms STOA methods on several metrics and shows better visual qualities.

2 Related Work

2.1 Deep Neural Networks for SVR.

There has been a lot of research on single image reconstruction tasks. Recent works involve 3D representation learning, including points [7, 15, 20], voxels [5, 37, 39], meshes [10, 33, 34, 8] and primitives [23, 30, 38]. Those representation can also be learned with differentiable rendering that do not require the ground truth 3D shapes [13, 18, 16, 42, 11, 15].

In this line of research, AtlasNet [10] represents 3D shapes as the union of several surface elements that are generated from the learned multilayer perceptrons (MLPs). Pixel2Mesh [33] generates genus-zero shapes as the deformations of ellipsoid template meshes. The mesh is progressively refined with higher resolutions using a graph convolutional neural network conditioned on the multi-scale image features. 3DN [34] also deforms a template mesh to the target, trained with a differentiable mesh sampling operator pushing sampled points to the target position.

2.2 Implicit Neural Representation for SVR.

The explicit 3D representations are usually limited by fixed shape resolution or topology. Recently, implicit functions for 3D objects have shown the advantages at representing complicated geometry [4, 40, 41, 14, 22, 17, 12, 26, 2, 9, 27, 29]. ImNet [4] uses an MLP-based neural network to approximate the signed distance field (SDF) of 3D shapes and shows improved results in contrast to the explicit surface representations. OccNet [21] generates an implicit volumetric shape by inferring the probability of each grid cell being occupied or not. The shape resolution is refined by repeatedly subdividing the interest cells. While those methods can capture the global shape structure, the geometric details are usually missing. In addition to the holistic shape description, DISN [40] adds a local image feature for each 3D point computed by aligning the image to the 3D shape using an estimated camera pose. With global and local features, DISN recovers much better geometric details and outperforms state-of-the-art methods. The local image feature of each 3D point sampling can be further enriched with its self-symmetry point, as shown in Ladybird [41]. Compared to Ladybird, we investigate a more general point structure, the spatial pattern, along with a deformable 2D kernel derived from the pattern, to encode geometric relationships for local image features.

2.3 Deformable Convolutional Networks.

Deformable convolution predicts a dynamic convolutional filter for each feature position [6]. Compared to locally connected convolutions, deformable convolution enables the exploration of non-local contextual information. The idea was

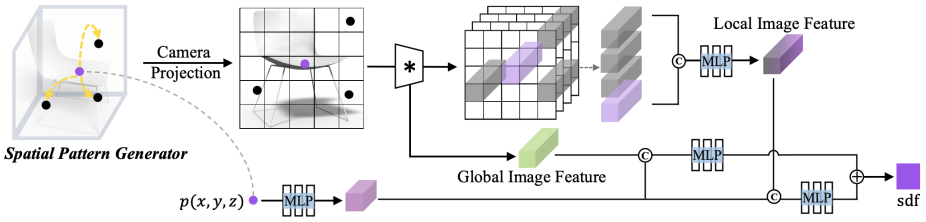


Fig. 2. The overview of our method. Given an image, our network predicts the signed distance field (SDF) for the underlying 3D object. To predict the SDF value for each point p, besides utilizing the global feature encoded from the image and the point feature directly inferred from p, local image features are fully exploited. Particularly, the local feature of a 3D point is encoded with a kernel in the image space whose kernel points are derived from a spatial pattern. *, Ⓒ and $\oplus$ denote convolution, concatenation, and sum operations respectively.

originally proposed for image processing and then extended for learning features from natural language [31], 3D point cloud [35, 32] and depth images [25]. In contrast to existing deformable kernels that generate kernel points within a 'single' domain, the proposed 2D deformable kernel is manipulated by a 3D spatial pattern generator, interacting between the 3D space and the 2D image plane.

3 Method

3.1 Overview

Given an RGB image of an object, our goal is to reconstruct the complete 3D shape of the object with high-quality geometric details. We use signed distance fields (SDF) to represent the 3D objects and approximate the SDFs with a neural network. Our network takes 3D points $p = (x, y, z) \in \mathbb{R}^3$ and an image I as input and outputs the signed distance s at each input location. With an SDF, the surface of an object can be extracted as the isosurface of $SDF(\cdot) = 0$ through the Marching Cubes algorithm. In general, our network consists of a fully convolutional image encoder m and a continuous implicit function f represented as multi-layer perceptrons (MLPs), from which the SDF is generated as

$$f(p, F_l(a), F_g) = s, s \in \mathbb{R}, \tag{1}$$

where $a = \pi(p)$ is the 2D projection for p, $F_l(a) = m(I(a))$ is the local feature at image location a, and F_g represents the global image feature. Feature $F_l(a)$ integrates the multi-scale local image features from the feature maps of m, from which the local image features are localized by aligning the 3D points to the image pixels via camera c.

By integrating with a spatial pattern at each 3D point sampling, the feature $F_l(a)$ of the sampling is modified by the local image features of the pattern

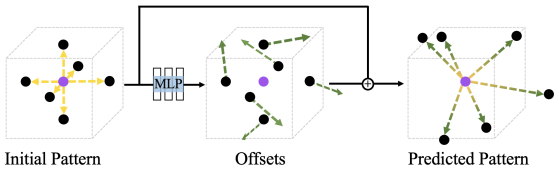


Fig. 3. Illustration of spatial pattern generator. For an input point sampling, a pattern is initialized with n points around it, and the offsets of the surrounding points are predicted by an MLP network. The final pattern is created as the sum of initial points and the corresponding offsets.

points. We devise a feature encoding kernel h attaching to the image encoder m to encode a new local image feature from the features extracted from the image feature map. Then our model is reformulated as

$$f(p, h(F_l(a), F_l(a_1), ..., F_l(a_n)), F_g) = s, \tag{2}$$

where pixels $a_1, ...a_n$ are the 2D projections of the 3D points $p_1, ..., p_n$ belonging to the spatial pattern of the point sampling p. The encoding kernel h is an MLP network that fuses the local image features. n is the number of the pattern points. Points $p_1, ..., p_n$ are generated by a spatial pattern generator which is addressed in the following subsection.

In general, our pipeline is designed to achieve better exploitation of contextual information from local image features extracted according to the predicted 3D spatial patterns, resulting in geometry-sensitive image feature descriptions for 3D point samplings, ultimately improving the 3D reconstruction from single-view images. A schematic illustration of the proposed model is given in Figure 2.

3.2 Spatial Pattern Generator

Our spatial pattern generator takes as input a 3D point sampling p, and outputs n 3D coordinates, i.e., $p_1, ..., p_n$. Like previous 2D or 3D deformable convolution networks [6, 32], the position of a pattern point is computed as the sum of the initial location and a predicted offset. A schematic illustration of the spatial pattern generator is shown in Figure 3.

Initialization. With proper initialization, the pattern can be learned efficiently and is highly effective for geometric reasoning. We consider two different sampling methods for spatial pattern initialization, i.e, uniform and non-uniform 3D point samplings, as shown in Figure 4. For simplicity, the input shapes are normalized to a unified cube centered at the origin. To generate uniform patterns, we uniformly sample n points along input point $p = (x, y, z)$. For example, we place a cube centered at p with the edge length of l, where each pattern point lies at the center of one of its side faces. We set $n = 6$ and $l = 0.2$, then each pattern points p_i can be written as the combinations of $p_i = (x \pm 0.1, y \pm 0.1, z \pm 0.1)$.

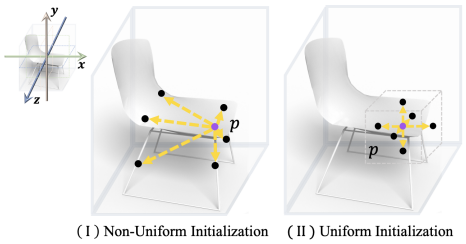
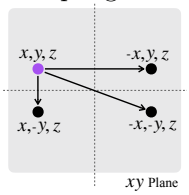


Fig. 4. Examples of spatial pattern initialization obtained by non-uniform sampling (I) and uniform sampling (II) strategies. In (I), a non-uniform pattern is formed by 3D points (in black) that are symmetry to input point p along x, y, z axis, and xy, yz, xz planes; and in (II), a uniform pattern is created by 3D points ling at centers of the side faces of a cube centered at point p.

Unlike the uniform sampling method, the non-uniform sampling method does not have commonly used strategies, except for random sampling. Randomly sampled points always do not have intuitive geometric meaning and are hardly appeared in any kernel point selection methods. As to capture non-local geometric relations, the non-uniform pattern points of p are created at the xy, yz, xz planes that pass through p. For instance, at xy plane, the non-uniform pattern expands at locations $(-x, y, z)$, $(x, -y, z)$ and $(-x, -y, z)$, as shown in the figure. More pattern points are created in the same way in yz, xz planes. Thus the pattern points p_i can be drawn from the combinations of $p_i = (\pm x, \pm y, \pm z)$.



After initialization, the pattern points, along with input sampling, are passed to an MLP network to generate the offsets and the final pattern is the sum of the initial positions and the predicted offsets. By projecting the 3D spatial pattern to the 2D image plane, we obtain a set of 2D pixels as their corresponding 2D pattern, which is the 2D kernel point. These image features derived from such a 2D kernel imply geometry relations.

3.3 Optimization.

Given a collection of 3D shapes and the generated implicit fields from images $\mathcal{I}$, the loss is defined with L_1 distance:

$$L_{SDF} = \sum_{I \in \mathcal{I}} \sum_{p} \omega |f(p, F_l^I, F_g^I) - SDF^I(p)|, \tag{3}$$

where SDF^I denotes the ground truth SDF value corresponding to image I and $f(\cdot)$ is the predicted field. ω is set to ω_1, if $SDF^I(p) < \delta$, and ω_2, otherwise. In practice, the parameters are set to $\omega_1 = 4, \omega_2 = 1$, and $\delta = 0.01$.

4 Experiments

In this section, we show qualitative and quantitative results on single-view 3D reconstruction from our method and comparisons with state-of-the-art methods. We also conduct a study on the variants of spatial patterns to understand the effect of initialization and the number of points. We further investigate the effectiveness of individual points in the spatial pattern with visualization and statistics results.

4.1 Experimental Details

Network Structure. The full network structure is shown in Figure 2. We use DISN [40] as our backbone network, which consists of a VGG-style fully convolutional neural network m as the image encoder. m has six convolutional layers with the dimension of {64, 128, 256, 512, 512, 512}. The spatial pattern generator is a MLPs, six layers with {64, 256, 512, 512, 256, 3} channels, ReLU activation and Tanh on output. Implicit function is also a MLPs, six layers with {64, 256, 512, 512, 256, 1} channels, ReLU activation. The feature aggregation module is a one-layer MLP directly mapping multiple local features to the output.

Dataset and Training Details. We use the ShapeNet Core dataset [3] and Pix3D dataset [28] for evaluation. The ShapeNet Core dataset [3] includes 13 object categories, and for each object, 24 views are rendered with resolution of 137×137 as in [5]. Pix3D Dataset [28] contains 9 object categories with real-world images and the exact mask images. The number of views and the image resolution varies from different shapes. We process all the shapes and images in the same format for the two datasets. Specifically, all shapes are normalized to the range of [-1,1] and all images are scaled to the resolution of 137×137.

In testing, for the ShapeNet dataset, the camera parameters are estimated from the input images, and we use the trained camera model from DISN [40] for fair comparisons. For the Pix3D dataset, ground truth camera parameters and image masks are used.

3D Point Sampling. For each shape, 2048 points are sampled for training. We first normalize the shapes to a unified cube with their centers of mass at the origin. Then we uniformly sample 256^3 grid points from the cube and compute the sign distance field (SDF) values for all the grid samples. Following the sampling process of Ladybird [41], the 256^3 points are downsampled with two stages. In the first stage, 32,768 points are randomly sampled from the four SDF ranges [-0.10,-0.03], [-0.03,0.00], [0.00,0.03], and [0.03,0.10], with the same probabilities. In the second stage, 2048 points are uniformly sampled from the 32,768 points using the farthest points sampling strategy.

In testing, 65^3 grid points are sampled are fed to the network, and output the SDF values. The object mesh is extracted as the zero iso-surface of the generated SDF using the Marching Cube algorithm.

3D-to-2D Camera Projection. Projecting 3D point sampling p to a pixel a is unfolded into two stages. Firstly, the point is converted from the world coordinate system to the local camera coordinate system c based on the rigid transformation matrix A^c, such that $p^c = A^c p$. Then in the camera space, point $p^c = (x^c, y^c, z^c)$ is projected to the 2D canvas via perspective transformation, i.e., $\pi(p^c) = (\frac{x^c}{z^c}, \frac{y^c}{z^c})$. The projected pixel whose coordinate lies out of an image will reset to 0 or 136 (the input image resolution is fixed as 137×137 in our experiment).

Traning Policy. We implement our method based on the framework of Pytorch. For training on the ShapeNet dataset, we use the Adam optimizer with a learning rate of 1e-4, a decay rate of 0.9, and a decay step size of 5 epochs. The network is trained for 30 epochs with a batch size of 20. For training on the Pix3D dataset, we use the Adam optimizer with a constant learning rate 1e-4 and a smaller batch size of 5. For the ShapeNet dataset, at each epoch, we randomly sample a subset of images from each category. Specifically, a maximum number of 36000 images are sampled for each category. The total number of images in an epoch is 411,384 resulting in 20,570 iterations. Our model is trained across all categories.

Evaluation Metrics. The quantitative results are obtained by computing the similarity between generated surfaces and ground truth surfaces. We use the standard metrics including Chamfer Distance (CD), Earth Mover's Distance (EMD), and Intersection over Union (IoU).

Table 1. Quantitative results on the ShapeNet Core dataset for various methods.

Metrics	Methods	plane	bench	cabinet	car	chair	display	lamp	speaker	rifle	sofa	table	phone	watercraft	mean
CD↓	Pixel2Mesh	6.10	6.20	12.11	13.45	11.13	**6.39**	31.41	14.52	4.51	**6.54**	15.61	6.04	12.66	11.28
	OccNet	7.70	6.43	9.36	5.26	7.67	7.54	26.46	17.30	4.86	6.72	10.57	7.17	9.09	9.70
	DISN	9.96	8.98	10.19	5.39	7.71	10.23	25.76	17.90	5.58	9.16	13.59	6.40	11.91	10.98
	Ladybird	5.85	6.12	9.10	5.13	7.08	8.23	21.46	14.75	5.53	6.78	9.97	**5.06**	6.71	8.60
	Ours$_{cam}$	**5.40**	**5.59**	8.43	**5.01**	**6.17**	8.54	14.96	**14.07**	3.82	6.70	**8.97**	5.42	**6.19**	**7.64**
	Ours	3.27	3.38	6.88	3.93	4.40	5.40	6.77	8.48	1.58	4.38	6.49	4.02	4.01	4.85
EMD↓	Pixel2Mesh	2.98	2.58	3.44	3.43	3.52	2.92	5.15	3.56	3.04	2.70	3.52	2.66	3.94	3.34
	OccNet	2.75	2.43	3.05	2.56	2.70	2.58	3.96	3.46	2.27	**2.35**	2.83	2.27	2.57	2.75
	DISN	2.67	2.48	3.04	2.67	2.67	2.73	4.38	3.47	2.30	2.62	3.11	2.06	2.77	2.84
	Ladybird	2.48	2.29	3.03	2.65	2.60	2.61	4.20	3.32	2.22	2.42	2.82	2.06	2.46	2.71
	Ours$_{cam}$	**2.35**	**2.15**	**2.90**	2.66	**2.49**	**2.49**	**3.59**	**3.20**	**2.04**	2.40	**2.70**	**2.05**	**2.40**	**2.57**
	Ours	1.91	1.90	2.58	2.36	2.17	2.08	2.66	2.75	1.52	2.11	2.36	1.77	1.99	2.17
IoU↑	Pixel2Mesh	51.5	40.7	43.4	50.1	40.2	55.9	29.1	52.3	50.9	60.0	31.2	69.4	40.1	47.3
	OccNet	54.7	45.2	**73.2**	73.1	50.2	47.9	37.0	**65.3**	45.8	67.1	**50.6**	70.9	52.1	56.4
	DISN	57.5	52.9	52.3	74.3	54.3	56.4	34.7	54.9	59.2	65.9	47.9	72.9	55.9	57.0
	Ladybird	60.0	53.4	50.8	74.5	55.3	57.8	36.2	55.6	61.0	68.5	48.6	73.6	**61.3**	58.2
	Ours$_{cam}$	**60.6**	**54.4**	52.9	74.7	**56.0**	**59.2**	**38.3**	56.1	**62.9**	**68.8**	49.3	**74.7**	60.6	**59.1**
	Ours	68.2	63.1	61.4	80.7	66.8	67.9	55.9	65.0	75.0	75.2	62.6	81.0	68.9	68.6

4.2 Quantitative and Qualitative Comparisons

We compare our method with the state-of-the-art methods on the single-image 3D reconstruction task. All the methods, including Pixel2Mesh [33], OccNet [21], DISN [40], Ladybird [41], are trained across all 13 categories. The method of

Table 2. Quantitative results on Pix3D dataset.

Categories	**CD**(x1000)↓		**EMD**(x100)↓		**IoU**(%)↑	
	Ladybird	Ours	Ladybird	Ours	Ladybird	Ours
bed	9.84	**8.76**	2.80	**2.70**	70.7	**73.2**
bookcase	**10.94**	14.70	**2.91**	3.32	**44.3**	41.8
chair	14.05	**9.81**	2.82	**2.72**	**57.3**	**57.3**
desk	18.87	**15.38**	3.18	**2.91**	51.2	**60.7**
misc	36.77	**30.94**	4.45	**4.00**	29.8	**44.0**
sofa	4.56	**3.77**	2.02	**1.92**	86.7	**87.6**
table	21.66	**14.04**	2.96	**2.78**	56.9	**58.8**
tool	**7.78**	16.24	3.70	**3.57**	**41.3**	38.2
wardrobe	**4.80**	5.60	**1.92**	2.01	**87.5**	**87.5**
mean	14.36	**13.25**	2.97	**2.88**	58.4	**61.0**

Ours uses ground truth cameras while Ours$_{cam}$ denotes the version of Ours using estimated camera poses.

A quantitative evaluation of the ShapeNet dataset is reported in Table 1 in terms of CD, EMD, and IoU. CD and EMD are evaluated on the sampling points from the generated triangulated mesh. IoU is computed on the solid voxelization of the mesh. In general, our method outperforms other methods. In particular, among DISN, Ladybird, and Ours, which share a similar backbone network, Ours achieves much better performance.

In Figure 9, we show qualitative results generated by Mesh R-CNN [8], OccNet [21], DISN [40] and Ladybird [41]. We use the pre-trained models from the Mesh R-CNN, OccNet, and DISN. For Ladybird, we re-implement their network and carry out training according to the specifications in their paper. All the methods can capture the general structure of the shapes. Shapes generated from DISN, Ladybird, and Ours are more aligned with the ground truth shapes. Specifically, our method is visually better at the non-visible regions and fine-scale geometry features.

The quantitative evaluation of the Pix3D dataset is provided in Table 2. Ours and Ladybird are both trained and evaluated on the same train/test split, during which ground truth camera poses and masks are used. Specifically, 80% of the images are randomly sampled from the dataset for training while the rest images are used for testing. In general, our method outperforms Ladybird on the used metrics. Note that Ladybird already outperforms the other methods shown in Table 1, we only give the results of Ladybird and Ours.

In addition to the quantitative results, we also show the reconstructed shapes in Figure 5. Compared to the synthetic images from the ShapeNet dataset, the real images are more diverse in terms of camera viewpoints, object sizes, and appearances. Our reconstructed shapes are visually more plausible compared to Ladybird.

4.3 Impact of Spatial Patterns

To figure out the influence of different spatial patterns, we designed several variants of the pattern. Specifically, two factors are considered, including the initialization and the capacity, i.e., the pattern point sampling strategy and the number of points in a pattern. As described before, we consider non-uniform

Fig. 5. Qualitative Results on the Pix3D dataset. Ground truth image masks and camera parameters are used.

and uniform sampling methods for pattern initialization and set the number of points to three and six. The variants derived from the combinations of those two factors are denoted as

- Ours$_{uniform-6p}$, in which six points are uniformly sampled on a cube centered at point sampling, such that $p_1 = (x, y, z{+}0.1)$, $p_2 = (x{+}0.1, y, z)$, $p_3 = (x, y{+}0.1, z)$, $p_4 = (x, y, z{-}0.1)$, $p_5 = (x{-}0.1, y, z)$, and $p_6 = (x, y{-}0.1, z)$.
- Ours$_{non-uni-6p}$, in which six points are non-uniformly sampled at the symmetry locations in the shape space along xy, yz and xz planes and x, y and z axes, such that $p_1 = (x, y, -z)$, $p_2 = (-x, y, z)$, $p_3 = (x, -y, z)$, $p_4 = (-x, -y, z)$, $p_5 = (x, -y, -z)$, and $p_6 = (-x, y, -z)$.
- Ours$_{non-uni-3p}$, in which three points are non-uniformly sampled at the symmetry locations in the shape space along xy, yz and xz planes, such that $p_1 = (x, y, -z)$, $p_2 = (-x, y, z)$, and $p_3 = (x, -y, z)$.

In Table 3, we report the numerical results of the methods using ground truth camera pose. In general, Ours$_{non-uni-6p}$ achieves best performance. By reducing the capacity to the number of three points, the performance decreases, as shown by Ours$_{non-uni-3p}$. It indicates that some critical points in Ours$_{non-uni-6p}$ that have high responses to the query point do not appear in Ours$_{non-uni-3p}$. Notably, the sampling strategy is more important. Both Ours$_{non-uni-6p}$ and Ours$_{non-uni-3p}$ outperforms Ours$_{uniform-6p}$ with large margins. Thus, initialization with non-uniform sampling makes the learning of effective spatial patterns easier. It implies that optimizing the pattern position in the continuous 3D space is challenging, and with proper initialization, spatial patterns can be learned more efficiently. To better understand the learned spatial pattern and which pattern points are preferred by the network, we provide analysis with

Table 3. Quantitative results of the variants of our method using different configurations of spatial pattern. Metrics include CD (multiply by 1000, the smaller the better), EMD (multiply by 100, the smaller the better), and IoU (%, the larger the better). CD and EMD are computed on 2048 points.

Metrics	Methods	plane	bench	cabinet	car	chair	display	lamp	speaker	rifle	sofa	table	phone	watercraft	mean
CD↓	$\text{Ours}_{uniform-6p}$	3.72	3.73	7.09	3.93	4.59	4.78	7.77	9.19	2.02	4.64	6.71	3.62	4.17	5.07
	$\text{Ours}_{non-uni-6p}$	3.27	3.38	6.88	3.93	4.40	5.40	6.77	8.48	1.58	4.38	6.49	4.02	4.01	**4.85**
	$\text{Ours}_{non-uni-3p}$	3.33	3.51	6.88	3.87	4.38	4.58	7.22	8.76	3.00	4.45	6.66	3.63	4.11	4.95
EMD↓	$\text{Ours}_{uniform-6p}$	2.07	2.02	2.60	2.38	2.19	2.11	2.86	2.85	1.55	2.16	2.41	1.78	2.01	2.23
	$\text{Ours}_{non-uni-6p}$	1.91	1.90	2.58	2.36	2.17	2.08	2.66	2.75	1.52	2.11	2.36	1.77	1.99	**2.17**
	$\text{Ours}_{non-uni-3p}$	1.96	1.94	2.58	2.35	2.16	2.07	2.81	2.81	1.58	2.13	2.39	1.78	2.00	2.20
IoU↑	$\text{Ours}_{uniform-6p}$	66.1	59.5	59.6	80.0	65.8	66.7	53.8	63.7	74.7	74.1	60.8	79.6	68.0	67.1
	$\text{Ours}_{non-uni-6p}$	68.2	63.1	61.4	80.7	66.8	67.9	55.9	65.0	75.0	75.2	62.6	81.0	68.9	**68.6**
	$\text{Ours}_{non-uni-3p}$	67.4	62.0	60.5	80.5	66.8	67.5	54.1	64.2	73.6	75.1	61.8	80.2	68.7	67.9

visualization and statistics in the next section. Before that, we evaluate the performance of our method by comparing it with several state-of-the-art methods. Specifically, we use $\text{Ours}_{non-uni-6p}$ as our final method.

4.4 Analysis of Learned Spatial Patterns

We have demonstrated the effectiveness of the proposed spatial pattern via achieving better performance than other alternatives, and the experiments on different variants of the spatial pattern show the influence of initialization and capacity. To better understand the importance of individual pattern points, we visualize several learned patterns in Figure 6&7 and calculate the mean offsets of the predicted pattern points visualized in Figure 8.

In Figure 6, we show learned spatial patterns in the 2D image plane. In each row, a spatial pattern is shown in six different images with different views. It implies an explicit constraint on view consistency of image encoding.

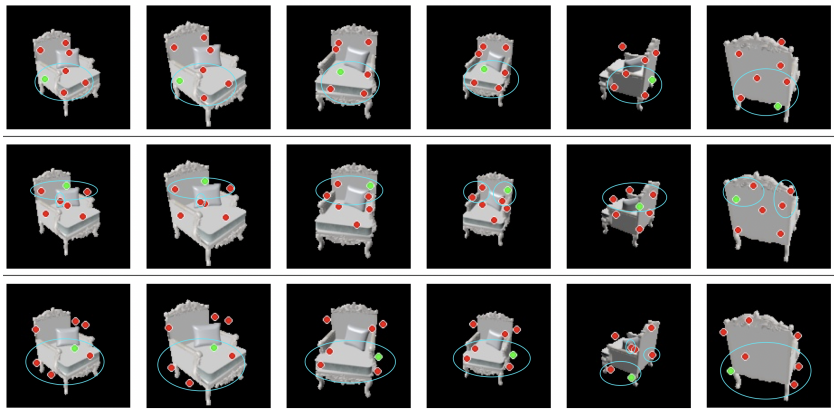


Fig. 6. Visualization of learned spatial patterns in image plane.

Pattern points (colored in red) that have intuitive geometric relationships (e.g., symmetric and co-planar) with the query points (colored in green) are highlighted by cyan circles in Figure 6. Figure 7 provides a better visualization in 3D frame, from which we can see that some learned pattern points from the non-uniform initialization are almost stationary, e.g., points p_1, p_2 and p_6 that are highlighted by dash circles. Also, as shown in Figure 8, the mean offsets of points p_1, p_2 and p_6 are close to zero. To figure out the importance of these stationary pattern points, we train the network using the points p_1, p_2, and p_6 as a spatial pattern and keep their positions fixed during training. As shown in Table 4, the performance of the selected rigid pattern is better than Ours$_{non-uni-3p}$ and Ours$_{uniform-6p}$ and slightly lower than Ours$_{non-uni-6p}$. This reveals that the pattern points discovered by the network are useful, which finally leads to a better reconstruction of the underlying geometry.

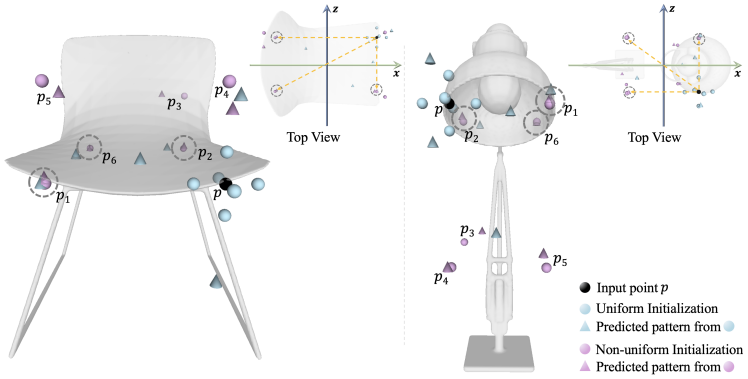


Fig. 7. Visualization of spatial pattern points with different shapes and colors. From the examples in (I) and (II), the learned pattern points (i.e., pink cones) from the non-uniform initialization (i.e., pink balls) are relative stationery, while points (i.e., blue cones) learned from uniform initialization (i.e., blue balls) have much larger deviations from their original positions. Some stationary points p_1, p_2 and p_6 are highlighted in dash circles. (Zoom in for better visualization)

Even though consuming more time and memory, utilizing the auxiliary contextual information brought by other points p_3, p_4, and p_5 only achieve a slight improvement in the performance. The analysis shows that naive selection of more neighboring points is not as effective as the strategy that considers the underlying geometric relationships. Although there is no explicit constraint to guarantee the geometric relations exactly, statistically we found that the network tends to shift the pattern points towards the locations that have geometric relations with the query point, as shown in 7&8. It further proves that encoding geometric relationships with the 2D kernel derived from the proposed spatial pattern are effective for the single-image 3D reconstruction task.

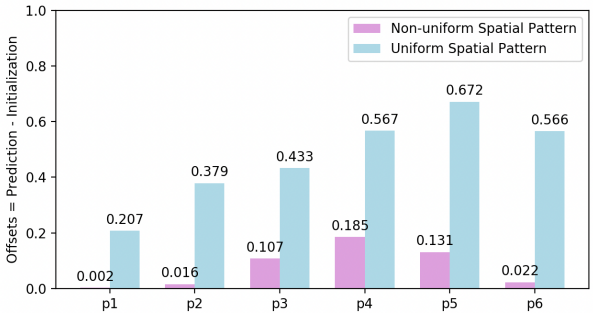


Fig. 8. Statistics on the offsets of spatial pattern points. The offset of individual pattern points is computed as the mean distance between the initial and predicted position. Among all points, p_1, p_2 and p_6 have the smallest learned offsets from the non-uniform initialization (i.e., pink bars), while for uniform initialization (i.e., blue bars), all the predicted points have much larger deviations from their original locations.

Table 4. Quantitative results of a rigid spatial pattern formed by three pattern points selected from the stationary points of the learned spatial pattern.

	plane	bench	cabinet	car	chair	display	lamp	speaker	rifle	sofa	table	phone	watercraft	mean
CD(x1000)	3.38	3.44	7.06	3.87	4.50	4.57	7.30	8.98	1.66	4.53	6.61	3.45	4.17	4.89
EMD(x100)	1.97	1.92	2.58	2.37	2.16	2.07	2.77	2.82	1.52	2.12	2.35	1.80	2.01	2.19
IoU(%)	67.4	62.8	60.5	80.5	66.6	67.4	54.9	64.5	74.9	75.0	62.5	80.1	68.5	68.1

5 Conclusion And Future Work

In this paper, we propose a new neural network that integrates a new feature encoding scheme to the deep implicit surface network for 3D shape reconstruction from single images. We present spatial patterns to allow the 2D kernel to encode local image features with geometric relations. Using spatial pattern enables the 2D kernel point selection explicitly to consider the underlying 3D geometry relations, which are essential in the 3D reconstruction task, while traditional 2D kernels mainly consider the appearance information. To better understand the spatial pattern, we study several variants of spatial pattern designs regarding the pattern capacity and the way of initialization, and we analyze the importance of individual pattern points. Results on large synthetic and real datasets show the superiority of the proposed method on widely used metrics.

A key limitation is that the model is sensitive to camera parameters. As shown in Table 1, when using ground truth camera parameters, the performance is significantly improved. One possible direction to investigate is to incorporate the camera estimation process in the loop of the 3D reconstruction pipeline, such as jointly optimizing the camera pose and the implicit field within a framework with multiple objectives. Another interesting direction is to learn geometric relations with explicit geometric constraints. Restricting the optimization to an optimized subspace could potentially promote performance and interpretation of learned patterns.

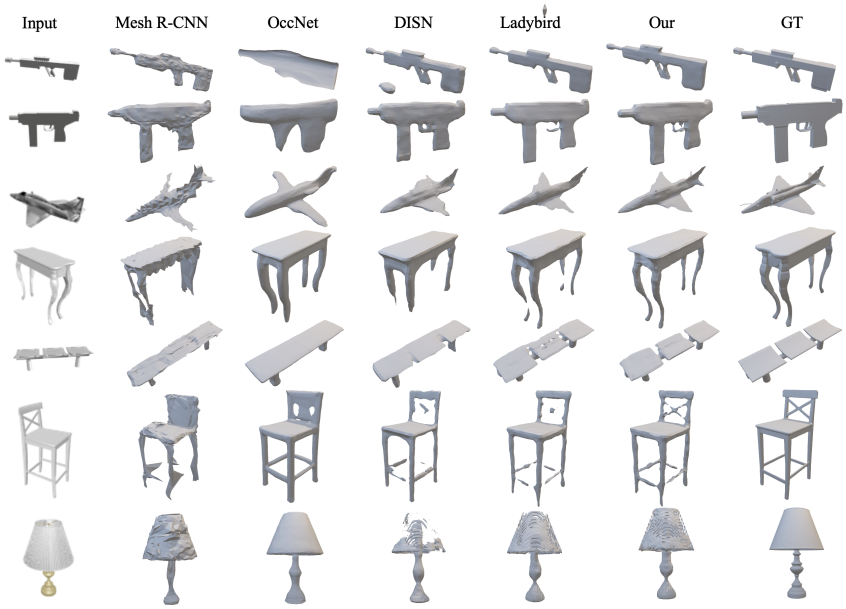


Fig. 9. Qualitative comparison results for various methods.

Acknowledgements We would like to thank the anonymous reviewers for their valuable feedback and suggestions.

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. In: International conference on machine learning. pp. 40–49. PMLR (2018)
2. Atzmon, M., Lipman, Y.: SAL: sign agnostic learning of shapes from raw data. In: CVPR. pp. 2562–2571. Computer Vision Foundation / IEEE (2020)
3. Chang, A.X., Funkhouser, T., J. Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository (arXiv:1512.03012 [cs.GR]) (2015)
4. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
5. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: European conference on computer vision. pp. 628–644. Springer (2016)
6. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)
7. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 605–613 (2017)

8. Gkioxari, G., Malik, J., Johnson, J.: Mesh r-cnn. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9785–9795 (2019)
9. Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y.: Implicit geometric regularization for learning shapes. In: ICML. Proceedings of Machine Learning Research, vol. 119, pp. 3789–3799. PMLR (2020)
10. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A papier-mâché approach to learning 3d surface generation. In: Proc. CVPR. pp. 216–224 (2018)
11. Insafutdinov, E., Dosovitskiy, A.: Unsupervised learning of shape and pose with differentiable point clouds. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. pp. 2807–2817 (2018)
12. Jiang, Y., Ji, D., Han, Z., Zwicker, M.: Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1251–1261 (2020)
13. Kato, H., Ushiku, Y., Harada, T.: Neural 3d mesh renderer. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3907–3916 (2018)
14. Li, M., Zhang, H.: D^2im-net: Learning detail disentangled implicit fields from single images. arXiv preprint arXiv:2012.06650 (2020)
15. Lin, C.H., Kong, C., Lucey, S.: Learning efficient point cloud generation for dense 3d object reconstruction. In: proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
16. Liu, L., Gu, J., Zaw Lin, K., Chua, T.S., Theobalt, C.: Neural sparse voxel fields. Advances in Neural Information Processing Systems **33** (2020)
17. Liu, S., Zhang, Y., Peng, S., Shi, B., Pollefeys, M., Cui, Z.: Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2019–2028 (2020)
18. Liu, S., Chen, W., Li, T., Li, H.: Soft rasterizer: Differentiable rendering for unsupervised single-view mesh reconstruction. arXiv preprint arXiv:1901.05567 (2019)
19. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. ACM siggraph computer graphics **21**(4), 163–169 (1987)
20. Mandikal, P., Navaneet, K., Agarwal, M., Babu, R.V.: 3d-lmnet: Latent embedding matching for accurate and diverse 3d point cloud reconstruction from a single image. arXiv preprint arXiv:1807.07796 (2018)
21. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3D reconstruction in function space. In: Proc. CVPR (2019)
22. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3504–3515 (2020)
23. Niu, C., Li, J., Xu, K.: Im2struct: Recovering 3d shape structure from a single rgb image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4521–4529 (2018)
24. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: DeepSDF: Learning continuous signed distance functions for shape representation. In: CVPR (2019)
25. Park, J., Joo, K., Hu, Z., Liu, C., Kweon, I.S.: Non-local spatial propagation network for depth completion. In: ECCV (13). Lecture Notes in Computer Science, vol. 12358, pp. 120–136. Springer (2020)
26. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In:

Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2304–2314 (2019)
27. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. In: NeurIPS. pp. 1119–1130 (2019)
28. Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J.B., Freeman, W.T.: Pix3d: Dataset and methods for single-image 3d shape modeling. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
29. Tancik, M., Srinivasan, P.P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J.T., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. NeurIPS (2020)
30. Tang, J., Han, X., Pan, J., Jia, K., Tong, X.: A skeleton-bridged deep learning approach for generating meshes of complex topologies from single rgb images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4541–4550 (2019)
31. Thomas, H., Qi, C.R., Deschaud, J., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. In: ICCV. pp. 6410–6419. IEEE (2019)
32. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. Proceedings of the IEEE International Conference on Computer Vision (2019)
33. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: ECCV. pp. 52–67 (2018)
34. Wang, W., Ceylan, D., Mech, R., Neumann, U.: 3dn: 3d deformation network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1038–1046 (2019)
35. Wu, F., Fan, A., Baevski, A., Dauphin, Y.N., Auli, M.: Pay less attention with lightweight and dynamic convolutions. In: ICLR. OpenReview.net (2019)
36. Wu, J., Zhang, C., Xue, T., Freeman, B., Tenenbaum, J.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: Advances in Neural Information Processing Systems. pp. 82–90 (2016)
37. Wu, J., Zhang, C., Zhang, X., Zhang, Z., Freeman, W.T., Tenenbaum, J.B.: Learning shape priors for single-view 3d completion and reconstruction. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 646–662 (2018)
38. Wu, R., Zhuang, Y., Xu, K., Zhang, H., Chen, B.: Pq-net: A generative part seq2seq network for 3d shapes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 829–838 (2020)
39. Xie, H., Yao, H., Sun, X., Zhou, S., Zhang, S.: Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2690–2698 (2019)
40. Xu, Q., Wang, W., Ceylan, D., Mech, R., Neumann, U.: Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. arXiv preprint arXiv:1905.10711 (2019)
41. Xu, Y., Fan, T., Yuan, Y., Singh, G.: Ladybird: Quasi-monte carlo sampling for deep implicit field based 3d reconstruction with symmetry. In: European Conference on Computer Vision. pp. 248–263. Springer (2020)
42. Yan, X., Yang, J., Yumer, E., Guo, Y., Lee, H.: Perspective transformer nets: learning single-view 3d object reconstruction without 3d supervision. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. pp. 1704–1712 (2016)