Unsupervised Learning of Fine Structure Generation for 3D Point Clouds by 2D Projection Matching

Chao Chen*
School of Software, BNRist, Tsinghua University, China
chenchao19@mails.tsinghua.edu.cn

Zhizhong Han*
Wayne State University
h312h@wayne.edu

Yu-Shen Liu
School of Software, BNRist, Tsinghua University, China
liuyushen@tsinghua.edu.cn

Matthias Zwicker
University of Maryland, College Park
zwicker@cs.umd.edu

Abstract

Learning to generate 3D point clouds without 3D supervision is an important but challenging problem. Current solutions leverage various differentiable renderers to project the generated 3D point clouds onto a 2D image plane, and train deep neural networks using the per-pixel difference with 2D ground truth images. However, these solutions are still struggling to fully recover fine structures of 3D shapes, such as thin tubes or planes. To resolve this issue, we propose an unsupervised approach for 3D point cloud generation with fine structures. Specifically, we cast 3D point cloud learning as a 2D projection matching problem. Rather than using entire 2D silhouette images as a regular pixel supervision, we introduce structure adaptive sampling to randomly sample 2D points within the silhouettes as an irregular point supervision, which alleviates the consistency issue of sampling from different view angles. Our method pushes the neural network to generate a 3D point cloud whose 2D projections match the irregular point supervision from different view angles. Our 2D projection matching approach enables the neural network to learn more accurate structure information than using the per-pixel difference, especially for fine and thin 3D structures. Our method can recover fine 3D structures from 2D silhouette images at different resolutions, and is robust to different sampling methods and point number in irregular point supervision. Our method outperforms others under widely used benchmarks. Our code, data and models are available at https://github.com/chenchao15/2D_projection_matching.

*indicates the equal contribution. This work was supported by National Key R&D Program of China (2020YFF0304100), the National Natural Science Foundation of China (62072268), and in part by Tsinghua-Kuaishou Institute of Future Media Data, and NSF (award 1813583). The corresponding author is Yu-Shen Liu.

1. Introduction

It is important to learn to generate 3D point clouds in different 3D computer vision applications, such as single image reconstruction [34, 39, 57, 17, 83] and novel shape generation [30, 6, 93, 63]. The latest supervised methods [12, 64, 65, 48, 30] leverage deep neural networks to learn to generate 3D point clouds from latent codes using 3D ground truth. However, it is expensive and tedious to obtain large scale 3D ground truth data sets, which significantly affects the supervised learning performance.

Unsupervised methods [34, 41, 39, 57, 83, 17] provide a more promising solution for 3D point cloud generation. Similar to unsupervised methods for other 3D representations, such as triangle meshes [42, 43, 36, 44, 9], voxel grids [92, 13, 79, 76], and implicit functions [69, 47, 35, 94, 45], these methods also leverage various differentiable renderers to learn to generate 3D point clouds using 2D images as supervision. In order to recover the 3D structure, the differentiable renderers project the generated 3D point clouds onto a 2D image plane with [34, 41, 39, 57, 83] or without [17] rendering to compare with the 2D supervision to obtain the per-pixel difference in training, such as density [34, 41, 39, 17, 83] or color [34, 57, 83] error. But these methods are still struggling to recover detailed 3D structures, especially for fine structures like thin tubes or planes.

To resolve this issue, we introduce a novel perspective for unsupervised learning of 3D point cloud generation with fine structure. Different from the current methods, which transform the generated 3D point clouds onto a 2D image plane to compare with regular pixel supervision, we discretize the area covered by the silhouette into discrete and irregular 2D points to compare with the 2D projections of the generated 3D point clouds. Without using the per-pixel difference obtained by various differentiable renderers, we cast the learning of 3D point cloud generation from silhou-

ette images as a 2D projection matching problem. Specifically, rather than using an entire 2D silhouette image as a regular pixel supervision, we discretize the silhouette by randomly sampling a 2D point set within it, which we regard as an irregular point supervision. Then, we push the neural network to generate 3D point clouds whose 2D projections on this silhouette image match the irregular point supervision. One advantage of the irregular point supervision is that it can still capture detailed structure information using sampled points even if fine structures are represented by only few pixels in the silhouette, which is hard for other differentiable renderers to leverage. Our irregular point supervision is robust to different sampling methods and the number of sampled points. Our outperforming results under widely used benchmarks show that our method can recover fine 3D structures from 2D silhouette images at different resolutions. Our contributions are as follows:

i) We introduce a method to enable the unsupervised learning of 3D point cloud generation with fine structures by 2D projection matching. Instead of using the per-pixel difference, we introduce irregular point supervision which is sampled from GT silhouettes.

ii) We justify the feasibility of the unsupervised learning of 3D point cloud generation using irregular 2D points rather than widely used regular 2D images. This helps to provide more detailed information for fine 3D structures which are represented by only few pixels.

iii) We demonstrate that our method can significantly improve the state-of-the-art accuracy in 3D point cloud generation applications by recovering finer structures under various benchmarks.

2. Related Work

Deep learning-based 3D shape understanding has achieved very promising results in different tasks [61, 54, 56, 88, 26, 16, 75, 4, 74, 27, 3, 91, 52, 30, 48, 33, 31, 18, 89, 86, 87, 49, 20, 22, 25, 21, 23, 32, 28, 24, 19, 50, 29, 67, 62, 53, 72, 60, 68, 85, 84]. Without 3D supervision, current unsupervised structure learning methods leveraged various differentiable renderers for different 3D raw representations. Differentiable renderers first render a reconstructed 3D shape into 2D silhouette or RGB images, and then, calculate the error between the rendered and GT images to train the neural networks.

Differentiable Renderers for Voxel Grids. By casting perspective rays through voxel grids, some differentiable renderers rendered images using the maximum occupancy values along each ray [92] or the derived ray collision probabilities [79]. While other differentiable renderers employ orthogonal projection using simple projection function [13]. These methods work with known camera poses [92, 79, 13], camera poses estimated from a separate network [76], or in the presence of viewpoint uncertainties [14].

Differentiable Renderers for Triangle Meshes. OpenDR [51] was introduced to approximate gradients with respect to pixel positions in back-propagation. Using handcrafted gradients, Kato et al. [36] were able to adjust faces on 3D meshes. Similarly, [42] and [43] analytically leveraged computed gradients from images to update face normals along with vertex positions via chain rule. By introducing more advanced rasterization, such as probabilistic rasterization [44] or regarding rasterization as interpolation of local mesh properties [9], the pixel value error compared to GT 2D images is used to update mesh reconstruction.

Differentiable Renderers for Implicit Functions. Implicit functions can represent 3D shapes using occupied voxels or signed distance functions in high resolution, which makes them very popular for deep learning models [66, 82, 54, 10, 59, 61, 55, 35, 26, 73, 2]. To reduce the computational cost on sampling points for implicit surface learning, Vincent et al. [69] learned a mapping from world coordinates to a feature representation of local scene properties. Similar to ray marching, various differentiable renderers [47, 35, 94] were proposed to render signed distance functions into images. In addition, ray-based field probing [46] or aggregating detection points on rays [90] was leveraged to mine supervision for 3D occupancy fields. With implicit differentiation, Niemeyer et al. [58] analytically derived in a differentiable rendering formulation for implicit shape and texture representations. Moreover, the naturally differentiable volume rendering was also employed to render a learned implicit radiance fields for view synthesis [56].

Differentiable Renderers for Point Clouds. Compactness is an advantage of 3D point clouds, which, however, brings an issue of sparseness among 2D projections of points in rendering. This issue makes it more difficult for differential renderers to directly compare the images rendered from these 2D projections with the ground truth images.

To resolve this issue, different renderers mainly employed either dense 3D points [41] or various rendering approaches [34, 39, 57, 83, 40, 38, 15, 70, 1] based on rasterization. Specifically, Lin et al. [41] proposed a pseudo-renderer to render dense points by modeling the visibility using pooling. However, it is significantly affected by the number of points. Instead, rendering based methods [34, 39, 57, 83] rasterized point clouds using surface splatting [83], Gaussian functions in 3D space [34] or on 2D images [39, 57]. CapNet [39] also leveraged a loss to match rendered pixels and GT pixels, however, the rendered pixels are interpolated from multiple 3D point locations, which makes the loss not effective to reveal accurate fine structures by adjusting each 3D point location. Without pixel-wise interpolation, visibility handling, or shading in render-

ing, DRWR [17] introduced a loss function to directly infer losses for each 3D point from pixel values and 2D projection relationship.

All these methods leveraged per-pixel difference to calculate the gradient in training. This makes it hard to fully recover fine 3D structures. To resolve this issue, we directly match the 2D projections of the generated 3D point clouds to 2D points randomly sampled from the GT silhouette.

Methods without Differentiable Rendering. Some earlier methods [7, 77, 78] did not leverage the rendering strategy to infer 3D structures from 2D images. However, they require strong priors, such as 3D templates [7, 77] or primitives [78], and the guidance of 2D and 3D key point correspondences obtained by manual annotation [7] or automatic methods [77], which also makes our method much different.

3. Method

Problem Statement. We aim to learn to generate a 3D point cloud $\boldsymbol{M}$ formed by J points $\boldsymbol{p}_j$ only using I ground truth silhouette images $\boldsymbol{v}_i$, without knowing 3D ground truth $\boldsymbol{G}$, where $j \in [1, J]$ and $i \in [1, I]$.

Supervised methods [12, 64, 65, 48, 30] can directly train neural networks by minimizing the Chamfer distance (CD) between the generated 3D points $\boldsymbol{M}$ and the 3D ground truth $\boldsymbol{G}$. Without $\boldsymbol{G}$, previous unsupervised methods leveraged the per-pixel difference from the error [34, 41, 39, 83, 57] between an image $\boldsymbol{v}_i'$ rendered with $\boldsymbol{M}$ from the the i-th view angle and a ground truth image $\boldsymbol{v}_i$ or from a loss [17] evaluated on $\boldsymbol{v}_i$.

Overview. Different from the previous differentiable rendering-based methods, we do not leverage the per-pixel difference. Instead, we first discretize the ground truth silhouette images $\boldsymbol{v}_i$ by randomly sampling K 2D points $\boldsymbol{g}_k^i$, and $k \in [1, K]$ within the silhouette, which we regard as an irregular point supervision. Then, we directly push the 2D projections $\{\boldsymbol{q}_j^i\}$ of the generated 3D point cloud $\boldsymbol{M}$ on the silhouette image $\boldsymbol{v}_i$ to match with the irregular point supervision $\{\boldsymbol{g}_k^i\}$. To leverage the supervision from all I view angles, we conduct this 2D projection matching procedure on each of the views.

We demonstrate our method in Fig. 1. Our purpose is to train a neural network to learn a mapping from an RGB input to a 3D point cloud $\boldsymbol{M}$ formed by a set of 3D points $\{\boldsymbol{p}_j\}$. To achieve this, we project the generated 3D points $\{\boldsymbol{p}_j\}$ onto one silhouette image $\boldsymbol{v}_i$ to get 2D projections $\{\boldsymbol{q}_j^i\}$ of $\boldsymbol{M}$. Then, we push projections $\{\boldsymbol{q}_j^i\}$ to match the irregular point supervision $\{\boldsymbol{g}_k^i\}$, which is randomly sampled within the silhouette on $\boldsymbol{v}_i$.

Advantages. By removing the requirement of the per-pixel difference, our method has several advantages over current differentibale rendering based methods.

Our method is simpler, since we do not require any complex and time consuming rendering procedures [34, 41, 39, 57, 83] such as surface interpolation, visibility handling, and shading.

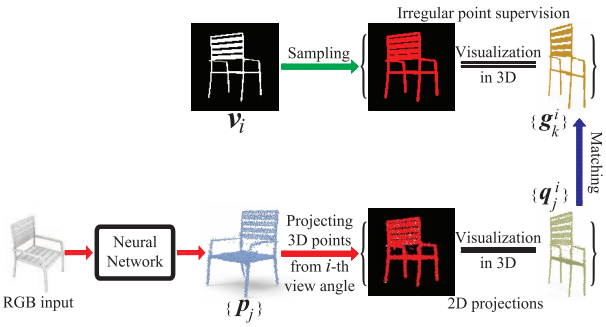

Figure 1. Overview of our method. We train the neural network to learn to generate 3D point clouds $\{\boldsymbol{p}_j\}$ by pushing its projection $\{\boldsymbol{q}_j^i\}$ to match with irregular point supervision $\{\boldsymbol{g}_k^i\}$. We obtain $\{\boldsymbol{g}_k^i\}$ by randomly sampling 2D points to cover a silhouette on ground truth silhouette image.

Although DRWR [17] does not require rendering either, it is hard to force the 2D projections to uniformly cover the silhouettes with fine structures by merely repulsing pairwise projections within the silhouettes. In contrast, our method is more effective by directly representing the fine structures using irregular point supervision.

Rendering based methods [34, 41, 39, 57, 83] are sensitive to the resolution of the ground truth silhouette images since they need to compare in a pixel-by-pixel manner. In contrast, we discretize the silhouette into irregular point supervision that is independent of image resolution. This enables us to formulate a loss based on 2D point matching, which we find to be more effective in our experiments.

We highlight our advantages over current differentiable renderers using an overfitting demonstration in Fig. 2. Current differentiable renderers depend on the per-pixel difference to evaluate how well the 2D projections cover the silhouette by using a rendering based pixel loss [34, 41, 39, 57, 83] in Fig. 2 (a) or a rendering free based point loss [17] in Fig. 2 (b). The projections shown on the top right in each subfigure demonstrate that current differentiable renderers cannot fully cover the fine structures on silhouette images, compared to the ground truth in Fig. 2 (d), which significantly affects the 3D point cloud generation.

Our method resolves this issue by directly matching 2D projections with the explicitly irregular point supervision. As shown in Fig. 3 (a), our irregular point supervision can provide more specific and accurate supervision as dense sampled points in the fine structure represented by merely few pixels, which is more effective than minimizing per-pixel difference using interpolated pixel values shown in Fig. 3 (b). During optimization, our much smaller length (darker color) of gradient for each point in Fig. 3 (c) than the gradient with per-pixel difference in Fig. 3 (d) demonstrates

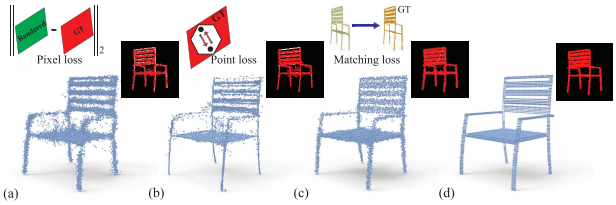

Figure 2. Overfitting experiment using different loss functions. Methods with rendering (a) and without rendering (b) leverage per-pixel difference to the ground truth 2D supervision (d) to infer 3D structures, while our method (c) provides a different perspective using 2D projection matching, where each of 16000 2D projections is shown in red on silhouette images.

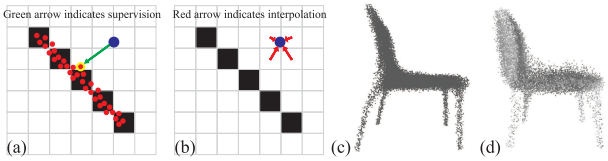

Figure 3. Our dense sampled points (red) in (a) are more effective supervision for projection (blue) to reveal fine structures (black) than per-pixel difference with interpolated pixel values in (b). The length of gradient for each point using our irregular point supervision or per-pixel difference is shown as color in (c) and (d).

that our irregular point supervision can provide a more clear target for each point, which is much easier to achieve.

Projecting 3D Points. We generate a 3D point cloud $\boldsymbol{M}$ in an object centered coordinate system. We leverage perspective projection to project 3D points $\{\boldsymbol{p}_j\}$ on $\boldsymbol{M}$ as 2D projections $\{\boldsymbol{q}_j^i\}$ on each silhouette image $\boldsymbol{v}_i$ from the i-th view angle. We denote $\boldsymbol{C}_i$ as both extrinsic and intrinsic camera parameters of the i-th camera pose, so we can perform the perspective projection as below,

$$[\boldsymbol{q}_j^i \quad 1]^T \sim \boldsymbol{C}_i[\boldsymbol{p}_j \quad 1]^T. \tag{1}$$

Irregular Point Supervision. We aim to provide the training supervision by discretizing a silhouette on image $\boldsymbol{v}_i$ through sampling the silhouette into irregular point supervision $\{\boldsymbol{q}_j^i\}$, which fully covers the silhouette. According to our preliminary results, we found that the consistency of points sampled on different silhouette images affects the performance, since the 2D projections on different silhouette images are from the same generated 3D point cloud. Although this issue can be alleviated by sampling very dense points within each silhouette with many different sampling methods, it would make the loss calculation more costly.

To improve the consistency of points sampled within the silhouette from different view angles, we introduce a *Structure Adaptive Sampling* (SAS) method to sample points within each silhouette. SAS first determines a ratio r to indicate how much area each one of K sampled points can cover within a silhouette. We calculate the area A of the whole silhouette by counting the number of pixels with a value of 1. So, the ratio r equals to A/K. Then, we calculate the sampling step s as $\sqrt{r}$, which is the edge length of the area that each sampled point covers. Finally, we start the sampler from the coordinate of $(0, 0)$ with a stride of s, and sample one point at each step if its interpolated pixel value is larger than 0.5.

Loss Function. Our method casts the learning of 3D point cloud generation into a 2D projection matching problem. Therefore, we push the neural network to generate a 3D point cloud $\boldsymbol{M}$ whose projections $\{\boldsymbol{q}_j^i\}$ on the i-th view can match the corresponding irregular point supervision $\{\boldsymbol{g}_k^i\}$, where we conduct the matching on each one of the I views.

On the i-th view, we leverage the CD to evaluate the distance between $\{\boldsymbol{q}_j^i\}$ and $\{\boldsymbol{g}_k^i\}$ as follows,

$$\begin{aligned} d(\{\boldsymbol{q}_j^i\}, \{\boldsymbol{g}_k^i\}) = & \frac{1}{J} \sum_{q \in \{\boldsymbol{q}_j^i\}} \min_{g' \in \{\boldsymbol{g}_k^i\}} ||q - g'||_2^2 \\ & + \frac{1}{K} \sum_{g \in \{\boldsymbol{g}_k^i\}} \min_{q' \in \{\boldsymbol{q}_j^i\}} ||g - q'||_2^2, \end{aligned} \tag{2}$$

Our preliminary results show that the Earth Mover Distance (EMD) can also get comparable results to CD when evaluating the matching distance. But EMD requires the number of the 2D projections and the number of irregular point supervision to be the same, which may not be necessary to sample many points within each silhouette.

We conduct the 2D projection matching on all I views by minimizing a loss function L below,

$$L = \sum_{i=1}^{i=I} d(\{\boldsymbol{q}_j^i\}, \{\boldsymbol{g}_k^i\}). \tag{3}$$

4. Experiments, Analysis and Discussion

4.1. Experimental Details

Dataset. For the fair comparison with the previous methods [34, 41, 39, 57, 83], we evaluate our method using the same three categories from ShapeNet [8], including chairs, cars, and airplanes. We also keep the same train/test splitting as in [79, 34, 17], and we employ the benchmark released by [34], which is formed by the rendered images and ground truth point clouds. For each 3D shape, there are $I = 5$ rendered views at three different resolutions including 32^2, 64^2, and 128^2, while all of them correspond to the same ground truth point clouds. Note that the ground truth point clouds have different numbers of points.

Metric. We evaluate our results using the CD between the predicted and the ground truth 3D point clouds. To compare with differentiable renderers for different 3D representations such as meshes or voxel grids, we also use volumetric IoU at a resolution of 32^3 to conduct fair comparisons.

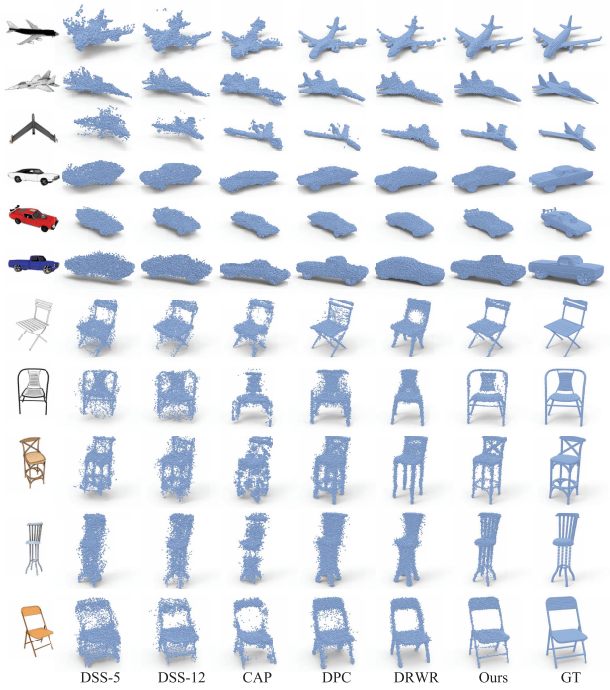

Figure 4. Visual comparison (16000 points) with the state-of-the-art methods using synthetic images.

For better readability, we multiply all CD or IoU values reported in our experiments by 100.

Setup. We realize that the neural network structure may affect the performance. Therefore, we employ the same neural network which was also employed by the previous differentiable renderers [34, 17] in evaluation. We will elaborate on the network structure in our supplemental material.

For training, we use image pairs from the same 3D shape to leverage the supervision from multiple views as the previous methods [34, 17]. For each pair of the $I = 5$ views, we use one RGB image as the input to generate a 3D point cloud $\boldsymbol{M}$ from the network, and then project $\boldsymbol{M}$ onto the other image to push the 2D projections $\{\boldsymbol{q}_j^i\}$ to match with the corresponding irregular point supervision $\{\boldsymbol{g}_k^i\}$.

We project the generated 3D point cloud $\boldsymbol{M}$ using the known camera pose although we can also leverage another network to estimate the camera pose from the input RGB image as [34, 76]. This is because camera pose estimation itself is another challenging problem which may affect the evaluation of 3D structure learning performance.

In addition, we train our network using all the three kinds of image resolutions provided in the benchmark [34] respectively, and accordingly generate point clouds at three different resolutions $J \in [2000, 8000, 16000]$.

We train our network using the Adam optimizer with a learning rate of 0.0001. Each batch contains 16 rendered images which are equally from 4 shapes, where we iterate over 5×10^6 batches in each experiment. We initially sample $K = 5000$ points within each silhouette from each view angle to form the irregular point supervision.

4.2. Single Image Reconstruction

We first evaluate our method in single image reconstruction. We train the neural network using the rendered images from the benchmark, while testing the network using the synthetic image or real image respectively.

Synthetic Image Testing. We compare our method with the latest methods leveraging the per-pixel difference. These methods include Differentiable Surface Splatting (DSS) [83], Differentiable Ray Consistency (DRC) [79], Efficient Point Cloud Generation (EPCG) [41], Continuous Approximation Projection (CAP) [39], Differentiable Point Clouds (DPC) [34], and Differentiable Renderer without Rendering (DRWR) [17]. DRC is voxel-based, and it is only available for voxel grids at a resolution of 32^3 because of the cubic complexity of voxel grids. The other four renderers are point cloud-based.

We first report the numerical comparison in terms of CD in Table 1. Our results are the best under all classes at all three resolutions. Our method shows significant improvement over voxel-based differentiable renderers including DRC and the voxel-based counterpart "DPC-V" of DPC [34]. Moreover, our method also reconstructs much more accurate point clouds compared to differentiable point renderers using per-pixel difference, such as CAP [39], DPC [34], and EPCG [41]. By leveraging 2D projection matching, our method is able to reconstruct 3D structures with high accuracy, especially for fine structures like thin planes and tubes. To demonstrate this, we conduct visual comparison with DSS [83], CAP [39], DPC [34], DRWR [17] in Fig. 4 using the reconstructed shapes in the test set, where DSS reconstructed point clouds with 16000 points from 5 views ("DSS-5") which are the same ones used by other methods or 12 views ("DSS-12"). The comparison shows that our method can reveal more 3D structure details, such as the complex structures on the chairs, cars and airplanes. Please see more point cloud reconstructions in our supplemental material.

Moreover, we also compare our method with the latest supervised 3D point cloud generation method called NOX [71]. We report our results using the same evaluation code and setting released by NOX [71] in Table 2. Specifically, we scale the point clouds reconstructed from input images at a resolution of 64^2 in Table 1, so that the diagonal of the bounding box of each reconstructed point cloud is one. We also resample the ground truth point cloud in the benchmark released by [34] to 8000 points which keeps the number of points the same as NOX [71].

Finally, we conduct a numerical comparison in terms of IoU with other supervised or unsupervised 3D shape generation methods for different 3D shape representations including triangle meshes, voxel grids, point clouds and

Table 1. Numerical comparison with differentiable renderers in terms of CD.

	Image-32^2, Shape-2000						Image-64^2, Shape-8000				Image-128^2, Shape-16000			
	DRC	CAP	DPC-V	DPC	DRWR	Ours	DPC-V	DPC	DRWR	Ours	EPCG	DPC	DRWR	Ours
Plane	8.35	6.34	5.57	4.52	4.01	**3.37**	4.94	3.50	3.18	**2.94**	4.03	2.84	2.66	**2.08**
Car	4.35	6.03	3.88	4.22	3.81	**3.50**	3.41	2.98	2.89	**2.81**	3.69	2.42	2.40	**2.25**
Chair	8.01	6.11	5.57	5.10	4.66	**4.16**	4.80	4.15	4.02	**3.94**	5.62	3.62	3.49	**3.10**
Mean	6.90	6.16	5.01	4.61	4.16	**3.68**	4.39	3.55	3.36	**3.23**	4.45	2.96	2.85	**2.48**

Table 2. Comparison with supervised point generation method.

CD	Cars	Airplanes	Chairs
NOX	0.1569	0.1855	0.3803
Ours	**0.0421**	**0.0492**	**0.0529**

implicit functions. The compared differentiable renderers include Perspective Transform Nets (PTN) [92], Neural Mesh Renderer (NMR) [36], SoftRasterizer (SoftRas) [44], Interpolation-based Differentiable Renderer (DIB-R) [9], Implicit Surface renderer (IMRender) [45], Implicit Function renderer (IMFun) [90], and SDFDiff [35]. The first method is voxel-based, the following two methods are mesh-based, while the last three are implicit function based. We report the results of NMR, SoftR and DIB-R from [9], and the rest from the original papers. The supervised methods in comparison include DISN [82], OccNet [54], IM-NET [10], 3DN [81], Pix2Mesh [80], R2N2 [11], and AtlasNet [16]. To report our IoU results, we voxelize the point clouds predicted from images at a resolution of 128^2 in Table 1 into voxel grids at a resolution 32^3 to compare to the same ground truth as other methods. Our outperforming results in Table 3 demonstrate our advantage in fine structure generation for 3D shapes. Note that although SDFDiff performs a little bit better under the Chair class, it uses RGB images as the supervision signal and requires to know the illumination and surface reflectance model, while our method does not require any such information. Fig. 5 demonstrates that our method can learn more complex structures on chairs than methods for meshes ("AtlasNet","Softras") and implicit functions ("OccNet"), where we produced their results using the trained models released from their papers [16, 44, 54].

Real Image Testing. Next, we evaluate our trained neural network by testing its adaptation to real images containing fine 3D structures. We randomly select some real images from the Internet, and use them as input to generate 3D point clouds at a resolution of 16000 points using the parameters learned in Table 1. As a comparison, we also generate the point clouds at the same resolution from the same images using parameters learned by DRWR [17]. Fig. 6 demonstrates that our method can adapt better to real images and reconstruct high fidelity 3D point clouds with more accurate fine structures than DRWR, such as the thin wings of airplanes and the thin legs of chairs.

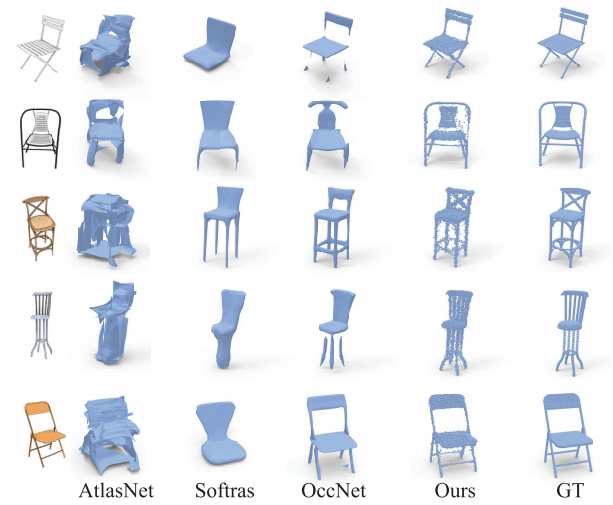

Figure 5. Visual comparison with the state-of-the-art supervised and unsupervised methods for different 3D representations.

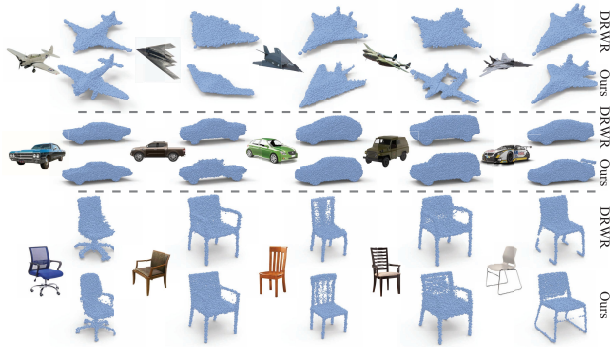

Figure 6. Visual reconstruction comparison using real images.

4.3. Novel Shape Generation

We further evaluate our method in another task of novel shape generation. We conduct this experiment under the Chair class to learn to generate novel point clouds at a resolution of 16000 from 128^2 images. Specifically, we modify our current encoder-decoder network structure by adding a KL loss on the latent code between the image encoder and the point decoder, which makes it very similar to the structure of Variational Auto Encoder (VAE) [37]. We train this network using a loss function formed by the matching loss in Eq. 2 and the KL loss with a balance ratio of $1 : 5 \times 10^{-6}$. In this way, we map the latent code of each input RGB image into a 32 dimensional Gaussian space during training so that we can generate a novel 3D point cloud from a ran-

Table 3. Numerical comparison with supervised and unsupervised 3D shape generation methods in terms of IoU.

	Unsupervised differentiable renderers									Supervised structure learning methods							
	PTN	NMR	SoftRas	DIB-R	IMRender	IMFun	SDFDiff	DRWR	Ours	DISN	OccNet	IMNET	3DN	Pix2Mesh	R2N2	AtlasNet	Ours
Car	71.2	71.3	77.1	78.8	78.2	66.0	80.0	75.3	**80.2**	74.3	73.7	74.5	59.4	50.1	66.1	22.0	**80.2**
Plane	55.6	58.5	58.4	57.0	65.1	53.3	68.7	62.2	**69.9**	57.5	57.1	55.4	54.3	51.5	42.6	39.2	**69.9**
Chair	44.9	41.4	49.7	52.7	54.8	44.4	**64.4**	58.1	62.7	54.3	50.1	52.2	34.4	40.2	43.9	25.7	**62.7**
Mean	57.2	57.1	61.7	62.8	66.0	54.6	**71.0**	65.2	**71.0**	62.0	60.3	60.7	49.4	47.3	50.9	29.0	**71.0**

Figure 7. Visual comparison in novel shape generation.

domly sampled latent code in the Gaussian space using the trained point decoder during inference.

We compare our method with DPC [34] and DRWR [17] by generating 3D point clouds using the randomly sampled latent codes in Fig. 7, where the three point clouds in each column are generated by the same latent code. The visual comparison demonstrates that our method can help the network to capture more fine 3D structures from the 2D supervision during training, which can be further leveraged to generate more reasonable and plausible 3D point clouds with fine structures, such as thin legs of chairs. Please see more point cloud generation in supplemental material.

4.4. Analysis and Discussion

Ablation Studies. We explore the effectiveness of the two terms in the loss function in Eq. 2. We conduct these experiments by merely using either one of the two terms to learn to generate 3D point clouds at a resolution of 2000 from 32^2 images under the Chair class. The degenerated results of "1st" and "2nd" in Table 4 show that the one directional distance loss cannot push the 2D projections to fully match with the irregular point supervision. To explore whether we can better resist the inconsistency among the irregular point supervision from different view angles, we adjust the number of nearest neighbors involved in the two terms in Eq. 2. We tried to leverage more nearest neighbors, such as 5 in the first term and 1 in the second term ("NN(5,1)"), 1 in the first term and 5 in the second term ("NN(1,5)"), or 5 in both terms ("NN(5,5)"). However, we do not observe any improvement compared to our original setting ("NN(1,1)").

Resolution of Irregular Point Supervision. The resolution of irregular point supervision is also important in training. If the number of points in the irregular point supervision is too small, the inconsistency among different view angles will be enlarged, which significantly affects the inference of fine structures on the 3D point cloud. On the other hand, if the number of points in the irregular point supervision is too large, it may bring redundancy without improving the learning performance, which also increases the computational burden when calculating the loss. To explore the trade-off, we sample different numbers of points within the silhouette to form the irregular point supervision, such that $K = \{1000, 3000, 5000, 7000, 9000\}$. Using the irregular point supervision at each resolution, we train a network to learn to generate 3D point clouds at a resolution of 2000 from 32^2 images under the Chair class. The results in Table 5 show that $K = 5000$ achieves the best accuracy and no improvement is observed with a larger K. Therefore, we use $K = 5000$ to establish irregular point supervision under different shape classes. In addition, the obtained results are also robust to different resolutions, since the results just change a little if the resolution is larger than $K = 3000$.

Table 4. Ablation studies in terms of CD.

	1st	2nd	NN(5,1)	NN(1,5)	NN(5,5)	NN(1,1)
CD	24.47	8.49	4.25	4.32	4.29	**4.16**

Table 5. Resolution of irregular point supervision comparison.

K	1000	3000	5000	7000	9000
CD	4.29	4.20	**4.16**	4.20	4.19

Sampling for Irregular Point Supervision. The sampling for establishing irregular point supervision also affects the performance because of the inconsistency among irregular point supervision from different view angles. Our structure adaptive sampling ("SAS") can alleviate the inconsistency to infer more accurate fine structures on 3D point clouds. We produce irregular point supervision with $K = 5000$ to generate shapes in 2000 points from 32^2 images under the Chair class using random sampling ("Rand"), pixel sampling ("Pixel"), pixel and random sampling ("Pix+Ran"), and Poisson-Disk sampling ("Poisson") [5].

Here, random sampling randomly samples a point on the silhouette image, and leverages a threshold of 0.5 to determine whether keeping this point in irregular point supervision if its interpolated pixel value is larger than 0.5, until the number of sampled points reaches $K = 5000$. While pixel sampling just leverages the pixel locations with an interpolated pixel value larger than 0.5. If the number of pixels is

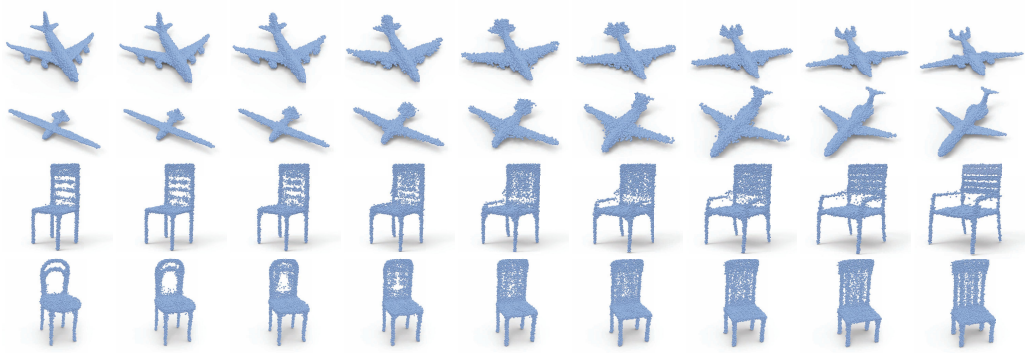

Figure 8. Shape interpolation demonstration to visualize the latent code space.

smaller than $K = 5000$, we will repeat the sampled pixel locations. "Pix+Ran" is similar to pixel sampling, but it uses randomly sampled points to replace the repeating procedure. Moreover, we also tried to do random sampling in a dynamic way, which aims to randomly sample $K = 5000$ points in each epoch to alleviate the impact of inconsistency using the randomness, as shown by the result of "Dynamic".

The comparison in Table 6 shows that our SAS sampling achieves the best inference performance for 3D fine structures during training due to the produced more consistent irregular point supervision. Random sampling in a dynamic way cannot improve the performance either. Moreover, we found that our method is robust to different sampling methods although SAS sampling could help the neural network to learn the most accurate structure. This is because the results with different sampling methods do not change a lot, and more importantly, all of them are better than the state-of-the-art result of 4.66 obtained by DRWR in Table 1.

Table 6. Sampling for irregular point supervision.

	Rand	Pixel	Pix+Ran	Poisson	Dynamic	SAS
CD	4.23	4.20	4.22	4.35	4.36	**4.16**

Latent Code Visualization. We visualize the latent space learned in the network which is trained to produce our results of 16000 points in Table 1. We randomly select two reconstructed point clouds in the test set, and employ their latent codes to interpolate seven new latent codes between them which are further used to generate seven novel shapes by the trained point decoder. We visualize two pairs of shape interpolation under each one of Airplane and Chair classes in Fig. 8. The smooth transformation from one shape to another shape demonstrates that our method can help the network to learn a meaningful latent space.

Robustness to 2D supervision resolution. Without using the per-pixel difference, our method is more robust to the resolution of 2D supervision. To demonstrate this, we train the neural network under the Chair class to generate 3D point clouds at a resolution of 16000 points from 32^2 images, rather than 128^2 images shown in Table 1. Due to the degenerated structure information caused by the lower resolution of 2D supervision, it is reasonable to obtain degenerated results. Therefore, we compare our method with DPC [34] and DRWR [17] in terms of degeneration in Table 7. The degeneration of each method is the difference between the results obtained with 32^2 and 128^2 images. The least degeneration shows that the 2D projection matching can help our method to become more robust to the resolution of 2D supervision, Our degeneration is shown in Fig. 9.

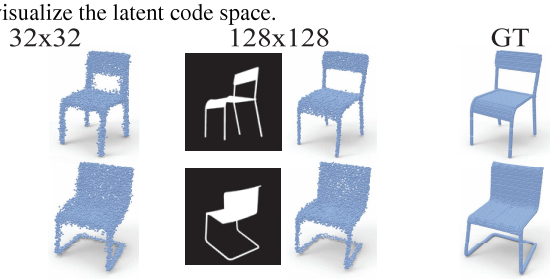

Figure 9. Robustness to the 2D supervision resolution.

Table 7. Robustness to 2D supervision resolution.

Resolution	DPC	DRWR	Ours
32^2	5.01	3.95	3.41
128^2	3.62	3.49	3.10
Degeneration	1.39	0.46	**0.31**

5. Conclusion

We introduce a novel perspective to learn to generate fine structures for 3D point clouds in an unsupervised way. Current differentiable renderers depend on the per-pixel difference to infer 3D structures from 2D supervision, which however cannot fully capture 3D structures, especially for fine structures. Our method successfully resolves this issue by casting this problem into a 2D projection matching problem. By discretizing the continuous area covered by the silhouette into irregular point supervision, our method effectively pushes the neural network to learn to generate 3D point clouds whose 2D projections can match the irregular point supervision as accurately as possible. We also demonstrate that the irregular point supervision can reveal more specific structure information to learn, especially for fine 3D structures. Our outperforming experimental results show that our method can significantly improve the structure generation performance for 3D point clouds.

References

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *European Conference on Computer Vision*, 2020.

[2] Dejan Azinovi, Ricardo Martin-Brualla, Dan B Goldman, Matthias Niener, and Justus Thies. Neural rgb-d surface reconstruction, 2021.

[3] Abhishek Badki, Orazio Gallo, Jan Kautz, and Pradeep Sen. Meshlet priors for 3D mesh reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2020.

[4] Jan Bednarik, Shaifali Parashar, Erhan Gundogdu, and Pascal Salzmann, Mathieu andFua. Shape reconstruction by learning differentiable surface representations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[5] Robert Bridson. Fast poisson disk sampling in arbitrary dimensions. In Marc Alexa and Adam Finkelstein, editors, *International Conference on Computer Graphics and Interactive Techniques*, 2007.

[6] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *European Conference on Computer Vision*, 2020.

[7] T. J. Cashman and A. W. Fitzgibbon. What shape are dolphins? building 3d morphable models from 2d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):232–244, 2013.

[8] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing 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. *CoRR*, abs/1512.03012, 2015.

[9] Wenzheng Chen, Jun Gao, Huan Ling, Edward J. Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3D objects with an interpolation-based differentiable renderer. *CoRR*, abs/1908.01210, 2019.

[10] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[11] Christopher Bongsoo 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, 2016.

[12] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. 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*, 2017.

[13] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3D shape induction from 2D views of multiple objects. In *International Conference on 3D Vision*, pages 402–411, 2017.

[14] Matheus Gadelha, Rui Wang, and Subhransu Maji. Shape reconstruction using differentiable projections and deep priors. In *International Conference on Computer Vision*, 2019.

[15] Artur Grigorev, Karim Iskakov, Anastasia Ianina, Renat Bashirov, Ilya Zakharkin, Alexander Vakhitov, and Victor Lempitsky. Stylepeople: A generative model of fullbody human avatars. In *IEEE Computer Vision and Pattern Recognition*, 2021.

[16] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. A papier-mch approach to learning 3D surface generation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[17] Zhizhong Han, Chao Chen, Yu-Shen Liu, and Matthias Zwicker. DRWR: A differentiable renderer without rendering for unsupervised 3D structure learning from silhouette images. In *International Conference on Machine Learning*, 2020.

[18] Zhizhong Han, Chao Chen, Yu-Shen Liu, and Matthias Zwicker. ShapeCaptioner: Generative caption network for 3D shapes by learning a mapping from parts detected in multiple views to sentences. In *ACM International Conference on Multimedia*, 2020.

[19] Zhizhong Han, Xinhai Liu, Yu-Shen Liu, and Matthias Zwicker. Parts4Feature: Learning 3D global features from generally semantic parts in multiple views. In *IJCAI*, 2019.

[20] Zhizhong Han, Zhenbao Liu, Junwei Han, Chi-Man Vong, Shuhui Bu, and C.L.Philip Chen. Mesh convolutional restricted boltzmann machines for unsupervised learning of features with structure preservation on 3D meshes. *IEEE Transactions on Neural Network and Learning Systems*, 28(10):2268 – 2281, 2017.

[21] Zhizhong Han, Zhenbao Liu, Junwei Han, Chi-Man Vong, Shuhui Bu, and C.L.P. Chen. Unsupervised learning of 3D local features from raw voxels based on a novel permutation voxelization strategy. *IEEE Transactions on Cybernetics*, 49(2):481–494, 2019.

[22] Zhizhong Han, Zhenbao Liu, Junwei Han, Chi-Man Vong, Shuhui Bu, and Xuelong Li. Unsupervised 3D local feature learning by circle convolutional restricted boltzmann machine. *IEEE Transactions on Image Processing*, 25(11):5331–5344, 2016.

[23] Zhizhong Han, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Shuhui Bu, Junwei Han, and CL Philip Chen. Deep Spatiality: Unsupervised learning of spatially-enhanced global and local 3D features by deep neural network with coupled softmax. *IEEE Transactions on Image Processing*, 27(6):3049–3063, 2018.

[24] Zhizhong Han, Honglei Lu, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, Junwei Han, and C.L. Philip Chen. 3D2SeqViews: Aggregating sequential views for 3D global feature learning by cnn with hierarchical attention aggregation. *IEEE Transactions on Image Processing*, 28(8):3986–3999, 2019.

[25] Zhizhong Han, Baorui Ma, Yu-Shen Liu, and Matthias Zwicker. Reconstructing 3d shapes from multiple sketches using direct shape optimization. *IEEE Transactions on Image Processing*, 29:8721–8734, 2020.

[26] Zhizhong Han, Guanhui Qiao, Yu-Shen Liu, and Matthias Zwicker. SeqXY2SeqZ: Structure learning for 3D shapes by sequentially predicting 1D occupancy segments from 2D coordinates. In *European Conference on Computer Vision*, 2020.

[27] Zhizhong Han, Mingyang Shang, Yu-Shen Liu, and Matthias Zwicker. View Inter-Prediction GAN: Unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions. In *AAAI*, pages 8376–8384, 2019.

[28] Zhizhong Han, Mingyang Shang, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, Junwei Han, and C.L. Philip Chen. SeqViews2SeqLabels: Learning 3D global features via aggregating sequential views by rnn with attention. *IEEE Transactions on Image Processing*, 28(2):685–672, 2019.

[29] Zhizhong Han, Mingyang Shang, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. Y2Seq2Seq: Cross-modal representation learning for 3D shape and text by joint reconstruction and prediction of view and word sequences. In *AAAI*, pages 126–133, 2019.

[30] Zhizhong Han, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. Multi-angle point cloud-vae:unsupervised feature learning for 3D point clouds from multiple angles by joint self-reconstruction and half-to-half prediction. In *IEEE International Conference on Computer Vision*, 2019.

[31] Zhizhong Han, Xiyang Wang, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, and C.L. Philip Chen. 3DViewGraph: Learning global features for 3D shapes from a graph of unordered views with attention. In *IJCAI*, 2019.

[32] Tao Hu, Zhizhong Han, Abhinav Shrivastava, and Matthias Zwicker. Render4Completion: Synthesizing multi-view depth maps for 3D shape completion. *ArXiv*, abs/1904.08366, 2019.

[33] Tao Hu, Zhizhong Han, and Matthias Zwicker. 3D shape completion with multi-view consistent inference. In *AAAI*, 2020.

[34] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *Advances in Neural Information Processing Systems*, pages 2807–2817, 2018.

[35] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. SDFDiff: Differentiable rendering of signed distance fields for 3D shape optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[36] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D mesh renderer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3907–3916, 2018.

[37] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[38] Maria Kolos, Artem Sevastopolsky, and Victor Lempitsky. Transpr: Transparency ray-accumulating neural 3D scene point renderer. In *3DV*, 2020.

[39] Navaneet K. L., Priyanka Mandikal, Mayank Agarwal, and R. Venkatesh Babu. Capnet: Continuous approximation projection for 3D point cloud reconstruction using 2D supervision. *AAAI*, 2019.

[40] Christoph Lassner and Michael Zollhfer. Pulsar: Efficient sphere-based neural rendering. In *IEEE Computer Vision and Pattern Recognition*, 2021.

[41] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *AAAI*, 2018.

[42] Hsueh-Ti Derek Liu, Michael Tao, and Alec Jacobson. Paparazzi: Surface editing by way of multi-view image processing. *ACM Transactions on Graphics*, 2018.

[43] Hsueh-Ti Derek Liu, Michael Tao, Chun-Liang Li, Derek Nowrouzezahrai, and Alec Jacobson. Beyond pixel normballs: Parametric adversaries using an analytically differentiable renderer. In *International Conference on Learning Representations*, 2019.

[44] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3D reasoning. *IEEE International Conference on Computer Vision*, 2019.

[45] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3D supervision. In *Advances in Neural Information Processing Systems*. 2019.

[46] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3D supervision. In *Advances in Neural Information Processing Systems*, 2019.

[47] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. DIST: Rendering deep implicit signed distance function with differentiable sphere tracing. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[48] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Point2Sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network. In *AAAI*, pages 8778–8785, 2019.

[49] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Fine-grained 3d shape classification with hierarchical part-view attention. *IEEE Transactions on Image Processing*, 30:1744–1758, 2021.

[50] Xinhai Liu, Zhizhong Han, Wen Xin, Yu-Shen Liu, and Matthias Zwicker. L2G auto-encoder: Understanding point clouds by local-to-global reconstruction with hierarchical self-attention. In *ACM International Conference on Multimedia*, 2019.

[51] Matthew M. Loper and Michael J. Black. OpenDR: An approximate differentiable renderer. In *European Conference on Computer vision*, volume 8695, pages 154–169, 2014.

[52] 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. In *International Conference on Machine Learning*, 2021.

[53] 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. *CoRR*, abs/2105.02788, 2021.

[54] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[55] Mateusz Michalkiewicz, Jhony K. Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders P. Eriksson. Deep level sets: Implicit surface representations for 3D shape inference. *CoRR*, abs/1901.06802, 2019.

[56] 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*, 2020.

[57] K L Navaneet, Priyanka Mandikal, Varun Jampani, and R Venkatesh Babu. DIFFER: Moving beyond 3D reconstruction with differentiable feature rendering. In *CVPR Workshops*, 2019.

[58] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[59] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. 2019.

[60] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: unifying neural implicit surfaces and radiance fields for multi-view reconstruction. *CoRR*, abs/2104.10078, 2021.

[61] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[62] Songyou Peng, Chiyu "Max" Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as points: A differentiable poisson solver. *CoRR*, abs/2106.03452, 2021.

[63] Albert Pumarola, Stefan Popov, Francesc Moreno-Noguer, and Vittorio Ferrari. C-flow: Conditional generative flow models for images and 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[64] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[65] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5105–5114, 2017.

[66] Shunsuke Saito, , Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. *IEEE International Conference on Computer Vision*, 2019.

[67] Vincent Sitzmann, Eric R. Chan, Richard Tucker, Noah Snavely, and Gordon Wetzstein. Metasdf: Meta-learning signed distance functions. In *Neural Information Processing Systems*, 2020.

[68] Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Frédo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *CoRR*, abs/2106.02634, 2021.

[69] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, 2019.

[70] Zhenbo Song, Wayne Chen, Dylan Campbell, and Hongdong Li. Deep novel view synthesis from colored 3D point clouds. In *European Conference on Computer Vision*, 2020.

[71] Srinath Sridhar, Davis Rempe, Julien Valentin, Sofien Bouaziz, and Leonidas J. Guibas. Multiview aggregation for learning category-specific shape reconstruction. In *Advances in Neural Information Processing Systems*. 2019.

[72] 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 *IEEE Computer Vision and Pattern Recognition*, 2021.

[73] 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. 2021.

[74] 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. *NeurIPS*, 2020.

[75] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Carsten Stoll, and Christian Theobalt. PatchNets: Patch-Based Generalizable Deep Implicit 3D Shape Representations. *European Conference on Computer Vision*, 2020.

[76] Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *Computer Vision and Pattern Regognition*, 2018.

[77] S. Tulsiani, A. Kar, J. Carreira, and J. Malik. Learning category-specific deformable 3d models for object reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):719–731, 2017.

[78] Shubham Tulsiani, Hao Su, Leonidas J. Guibas, Alexei A. Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *IEEE Computer Vision and Pattern Regognition*, 2017.

[79] Shubham Tulsiani, Tinghui Zhou, Alexei A. Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 209–217, 2017.

[80] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3D mesh models from single RGB images. In *European Conference on Computer Vision*, pages 55–71, 2018.

[81] Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. 3DN: 3D deformation network. In *IEEE International Conference on Computer Vision*, 2019.

[82] Weiyue Wang, Qiangeng Xu, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. DISN: Deep implicit surface network for high-quality single-view 3D reconstruction. In *NeurIPS*, 2019.

[83] Yifan Wang, Serena Felice, Wu Shihao, Öztireli Cengiz, and Sorkine-Hornung Olga. Differentiable surface splatting

for point-based geometry processing. *ACM Transactions on Graphics*, 38(6), 2019.

[84] Yifan Wang, Lukas Rahmann, and Olga Sorkine-Hornung. Geometry-consistent neural shape representation with implicit displacement fields. *CoRR*, abs/2106.05187, 2021.

[85] Yifan Wang, Shihao Wu, A. Cengiz Öztireli, and Olga Sorkine-Hornung. Iso-points: Optimizing neural implicit surfaces with hybrid representations. *CoRR*, abs/2012.06434, 2020.

[86] Xin Wen, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Cycle4completion: Unpaired point cloud completion using cycle transformation with missing region coding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[87] Xin Wen, Zhizhong Han, Xinhai Liu, and Yu-Shen Liu. Point2spatialcapsule: Aggregating features and spatial relationships of local regions on point clouds using spatial-aware capsules. *IEEE Transactions on Image Processing*, 29:8855–8869, 2020.

[88] Xin Wen, Tianyang Li, Zhizhong Han, and Yu-Shen Liu. Point cloud completion by skip-attention network with hierarchical folding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[89] Xin Wen, Peng Xiang, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Pmp-net: Point cloud completion by learning multi-step point moving paths. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

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

[91] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *IEEE International Conference on Computer Vision*, 2021.

[92] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In *Advances in Neural Information Processing Systems*, pages 1696–1704. 2016.

[93] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge J. Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *IEEE International Conference on Computer Vision*, 2019.

[94] Sergey Zakharov, Wadim Kehl, Arjun Bhargava, and Adrien Gaidon. Autolabeling 3D objects with differentiable rendering of sdf shape priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.