DRWR: A Differentiable Renderer without Rendering for Unsupervised 3D Structure Learning from Silhouette Images

Zhizhong Han [1 2] **Chao Chen** [1] **Yu-Shen Liu** [1] **Matthias Zwicker** [2]

Abstract

Differentiable renderers have been used successfully for unsupervised 3D structure learning from 2D images because they can bridge the gap between 3D and 2D. To optimize 3D shape parameters, current renderers rely on pixel-wise losses between rendered images of 3D reconstructions and ground truth images from corresponding viewpoints. Hence they require interpolation of the recovered 3D structure at each pixel, visibility handling, and optionally evaluating a shading model. In contrast, here we propose a *Differentiable Renderer Without Rendering* (DRWR) that omits these steps. DRWR only relies on a simple but effective loss that evaluates how well the projections of reconstructed 3D point clouds cover the ground truth object silhouette. Specifically, DRWR employs a smooth silhouette loss to pull the projection of each individual 3D point inside the object silhouette, and a structure-aware repulsion loss to push each pair of projections that fall inside the silhouette far away from each other. Although we omit surface interpolation, visibility handling, and shading, our results demonstrate that DRWR achieves state-of-the-art accuracies under widely used benchmarks, outperforming previous methods both qualitatively and quantitatively. In addition, our training times are significantly lower due to the simplicity of DRWR.

1. Introduction

Learning to represent and reconstruct 3D structure is a core problem in 3D computer vision. Supervised deep learning methods (Mescheder et al., 2019; Qi et al., 2017a;b; Wang et al., 2019) have been highly successful by directly learning from 3D data provided as meshes, point clouds, or voxel volumes. However, these methods require large amounts of 3D data in training, which is expensive and time consuming to obtain. In contrast, unsupervised 3D structure learning, that is, 3D structure learning without 3D supervision, is an attractive and promising alternative because it requires only images as training data.

Differentiable renderers (Insafutdinov & Dosovitskiy, 2018; L. et al., 2019; Navaneet et al., 2019; Yifan et al., 2019) are a core component of unsupervised 3D structure learning methods. They can bridge the gap between 3D to 2D by enabling the computation of gradients of 2D loss functions with respect to 3D structure. 2D loss functions are usually defined based on differences in RGB pixel values or pixel-wise silhouette coverage. By rendering a predicted 3D structure from a specific view angle into an image, and then evaluating a loss function based on the difference between the rendered and ground truth image, the parameters of deep learning models can be optimized to recover 3D structures that are consistent with the ground truth image, as illustrated in Fig. 1(a). To evaluate pixel-wise loss functions, previous techniques render images using some form of interpolation of the recovered 3D structure at each pixel, such as rasterization, visibility handling (e.g., z-buffering), and optionally per-pixel shading. For point cloud reconstruction, differentiable renderers have been proposed based on rasterizing Gaussian functions into 3D grids (Insafutdinov & Dosovitskiy, 2018) and on 2D planes (L. et al., 2019; Navaneet et al., 2019), or using surface splatting (Yifan et al., 2019). While pixel-wise interpolation, visibility handling, and shading in these approaches significantly increase the computational cost, one of our main insights here is that these steps do not actually contribute to accurate 3D structure learning.

To show this, we propose a *Differentiable Renderer Without Rendering* (DRWR) for unsupervised 3D point cloud reconstruction from 2D silhouette images. In contrast to pixel-wise losses in previous differentiable renderers, DRWR produces a loss only from the 2D projections of the 3D points, without pixel-wise interpolation, visibility handling, or shading, as shown in Fig. 1(b). Intuitively, the DRWR loss captures how well the projected points cover the object

[1]School of Software, BNRist, Tsinghua University, Beijing 100084, P. R. China [2]Department of Computer Science, University of Maryland, College Park, USA. Correspondence to: Yu-Shen Liu <liuyushen@tsinghua.edu.cn>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

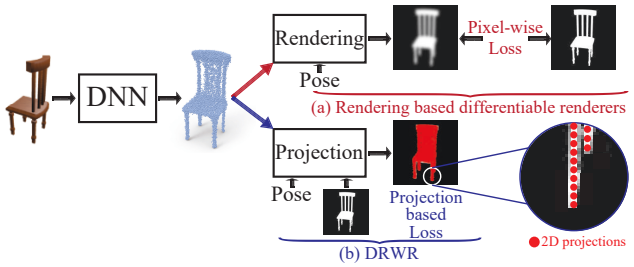

Figure 1. Illustration of the difference between rendering-based differentiable renderers with a pixel-wise loss in (a) and DRWR with a projection-based loss without rendering in (b).

silhouette, or the foreground. More specifically, the loss function includes a unary and a pairwise loss. The unary loss is designed to pull the projection of each 3D point into the foreground, while the pairwise loss pushes each pair of projections that lie inside the foreground far away from each other. This ensures that the entire foreground is covered and it prevents points from clumping. In addition, to avoid getting stuck in a local minima, we construct a smooth silhouette loss as the unary loss, which is designed to produce non-zero gradients for all points until their projections move into the foreground. To make the optimization more efficient and accurate, we also formulate the pairwise loss in a structure-aware manner, where we adaptively take into account the repulsion between each pair of projections only when both of them appear in the foreground. Our experiments show that DRWR outperforms all previous methods in achieving state-of-the-art results under widely used benchmarks. In summary, our main contributions are as follows:

i) We propose DRWR to justify the idea of conducting unsupervised 3D structure learning for point clouds using a differentiable renderer without rendering, that is, without pixel-wise interpolation of 3D structure, visibility handling, or shading.

ii) We demonstrate that DRWR reduces training time while improving the state-of-the-art accuracy of reconstructed point clouds under widely used benchmarks.

iii) We introduce a smooth silhouette loss and a structure-aware repulsion loss based on the projections of 3D points. The resulting model can be trained efficiently and robustly.

2. Related work

Deep learning models have made significant progress in different 3D applications (Han et al., 2016; 2017a;b; 2018; 2019a;b;c;d;e;f;g; 2020a;b; Hu et al., 2019; 2020; Liu et al., 2019d;e; Mescheder et al., 2019; Park et al., 2019; Qi et al., 2017b; Wen et al., 2020a;b). Here, we briefly review differentiable renderers for different 3D representations including voxel grids, meshes, implicit functions and point clouds, which is most related to our methods.

Voxel grids. Yan et al. (Yan et al., 2016) selected the maximum occupancy value along a ray to learn to reconstruct 3D voxel grids from silhouette images. Gadelha et al. (Gadelha et al., 2017) employed orthogonal projection using simple projection function to bridge 3D voxel grids to silhouette images. Tulsiani et al. (Tulsiani et al., 2017b) derived a differentiable formulation by leveraging ray collision probabilities. These methods work with known camera poses. Then, Tulsiani et al. (Tulsiani et al., 2018) extended (Tulsiani et al., 2017b) with an additional network to simultaneously predict camera poses. Similarly, Gadelha et al. extended the projection (Gadelha et al., 2017) in the presence of viewpoint uncertainties.

Meshes. OpenDR (Loper & Black, 2014) is the pioneer of differentiable rendering based renderers, which approximates gradients with respect to pixel positions to back-propagate. With hand-crafted gradients, Kato et al. (Kato et al., 2018) also achieved the adjustment of faces. To analytically compute gradients, (Liu et al., 2018) and (Liu et al., 2019a) back-propagated the image gradients to the face normals which are further used to update vertex positions via chain rule. Liu et al. (Liu et al., 2019b) introduced SoftRas with a probabilistic rasterization and assigned each pixel to all faces of a mesh. Similarly, Chen et al. (Chen et al., 2019) regarded rasterization as interpolation of local mesh properties by computing analytic gradients of foreground pixels.

Implicit functions. Implicit functions have been attracting more research interests as a new 3D shape representation to learn using deep learning models (Chen & Zhang, 2019; Jiang et al., 2020; Mescheder et al., 2019; Michalkiewicz et al., 2019; Oechsle et al., 2019; Park et al., 2019; Saito et al., 2019; Wang et al., 2019), due to its great representative ability with voxels or signed distance function in high resolutions. To reduce the computational cost on sampling implicit surface required in training, Vincent et al. (Sitzmann et al., 2019) learned a mapping from world coordinates to a feature representation of local scene properties. Based on the idea of ray marching rendering, different differentiable renderers (Jiang et al., 2020; Liu et al., 2020; Zakharov et al., 2020) were proposed to render the signed distance function. While Liu et al. (Liu et al., 2019c) proposed a novel ray-based field probing technique to mine supervision for 3D occupancy fields. With the implicit differentiation, (Niemeyer et al., 2020) derived analytically in a differentiable rendering formulation for implicit shape and texture representations.

Point clouds. Due to the compactness, point clouds are also an important 3D representation in deep learning based 3D

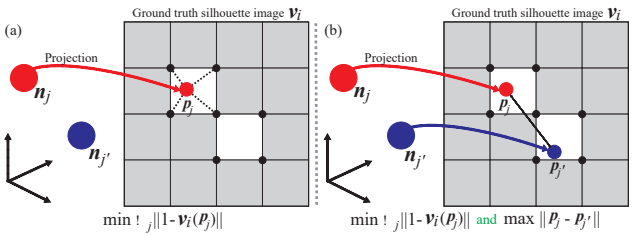

Figure 2. DRWR learns the structure of 3D point clouds from multiple silhouette images. For clarity, we show only two points $\boldsymbol{n}_j$ and $\boldsymbol{n}_{j'}$ and one silhouette image $\boldsymbol{v}_i$. The goal of uniformly locating the projections $\boldsymbol{p}_j$ inside $\boldsymbol{v}_i$ is implemented by two losses: (a) the first loss pulls all projections into the foreground (white areas) by minimizing the error between 1 and the pixel value of each projection $\boldsymbol{v}_i(\boldsymbol{p}_j)$. The second loss in (b) pushes each pair of projections in the foreground far away from each other.

shape understanding. However, this also brings an issue of sparseness among 2D projections of points. This issue makes the images with these projections impossible to directly compare with the ground truth images, which is hard to handle by differential renderers. To resolve this issue, different renderers mainly employed either dense points (Lin et al., 2018) or different rendering approaches (Insafutdinov & Dosovitskiy, 2018; L. et al., 2019; Navaneet et al., 2019; Yifan et al., 2019) based on rasterization. Specifically, Lin et al. (Lin et al., 2018) resorted to dense points and proposed pseudo-renderer to model the visibility using pooling. However, it is significantly affected by the number of points. Instead, rendering based methods (Insafutdinov & Dosovitskiy, 2018; L. et al., 2019; Navaneet et al., 2019; Yifan et al., 2019) approximated the distribution of point clouds using surface splatting (Yifan et al., 2019) or gaussian functions in 3D space (Insafutdinov & Dosovitskiy, 2018) and on 2D images (L. et al., 2019; Navaneet et al., 2019). But rendering adds computational burden, while it is not clear whether it contributes to improving the reconstruction accuracy.

Different from these methods, DRWR can be used to generate point clouds containing arbitrary numbers of points and produces a loss for each point without rendering.

Moreover, DRWR is also much different from the previous methods (Cashman & Fitzgibbon, 2013; Tulsiani et al., 2017; Tulsiani et al., 2017a) which did not leverage the rendering strategy to reveal the 3D structures from 2D images. These methods require strong priors, such as 3D template (Cashman & Fitzgibbon, 2013; Tulsiani et al., 2017) or primitives (Tulsiani et al., 2017a), and the guidance of 2D and 3D key point correspondence obtained by manually annotated (Cashman & Fitzgibbon, 2013) or algorithm (Tulsiani et al., 2017). However, DRWR do not require any of these.

3. Details of DRWR

Overview. Our goal is to learn the structure of 3D point clouds $\boldsymbol{N}$ formed by J points $\boldsymbol{n}_j$ only from I ground truth silhouette images $\boldsymbol{v}_i$, where $j \in [1, J]$ and $i \in [1, I]$. Current differentiable renderers rely on rendering point clouds $\boldsymbol{N}$ into raster images $\boldsymbol{v}_i'$ from the i-th view angle, which would then be used to produce a loss by comparing $\boldsymbol{v}_i'$ and $\boldsymbol{v}_i$ pixel-by-pixel. We argue, however, that rendering with pixel-wise interpolation of 3D structure, visibility handling, and shading adds unnecessary computational cost, and accurate results can be achieved without these steps.

To demonstrate this, we propose DRWR, a differentiable renderer without rendering, providing a novel perspective for unsupervised learning of 3D structure. Denoting the projection of point $\boldsymbol{n}_j$ in view i as $\boldsymbol{p}_j^i$, DRWR computes a loss by evaluating how well the sets of projected points $\{\boldsymbol{p}_j^i | j \in [1, J]\}$ cover the object silhouette. The DRWR loss consists of a unary and a pairwise term, as illustrated in Fig. 2. Given a predicted 3D point cloud and a binary silhouette image, we compute the loss as follows (only two points for illustration): We first project the points $\boldsymbol{p}_j$ (short for $\boldsymbol{p}_j^i$) onto the silhouette images $\boldsymbol{v}_i$, denoting the pixel value of the projection $\boldsymbol{p}_j$ as $\boldsymbol{v}_i(\boldsymbol{p}_j)$. The unary loss penalizes points outside the foreground by simply computing the difference $1 - \boldsymbol{v}_i(\boldsymbol{p}_j)$, assuming that the foreground in the binary silhouette image has value 1. Minimizing this loss will pull all projections into the foreground. In addition, the pairwise loss adjusts the spatial distribution of projected points by pushing pairs of projections in the foreground to be as far from each other as possible, as shown in Fig. 2(b). DRWR adjusts the 3D locations of points $\boldsymbol{n}_j$ through their projections $\boldsymbol{p}_j$ by jointly optimizing these two losses. DRWR can produce this loss and its gradients for any generative neural network for 3D point clouds, enabling unsupervised training as shown in Fig. 1(b).

3D-to-2D Projection. In our approach, we represent 3D point clouds $\boldsymbol{N}$ in an object centered coordinate system. Using the perspective transformation, we start by transforming the coordinate of each point $\boldsymbol{n}_j$ into the projection $\boldsymbol{p}_j^i$ on each silhouette image $\boldsymbol{v}_i$ from the i-th view angle. Using $\boldsymbol{T}_i$ as both extrinsic and intrinsic camera parameters of the i-th camera pose, we conduct the projection below,

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

Smooth Silhouette Loss. We model the unary loss as the difference between 1 and the pixel value $\boldsymbol{v}_i(\boldsymbol{p}_j^i)$ of each projection $\boldsymbol{p}_j^i$ on silhouette image $\boldsymbol{v}_i$. Here, we employ bilinear interpolation to obtain $\boldsymbol{v}_i(\boldsymbol{p}_j^i)$ using the binary pixel values of the nearest pixels around $\boldsymbol{p}_j^i$, as demonstrated in Fig. 2(a). DRWR aims to pull all projections into the foreground on all silhouette images $\boldsymbol{v}_i$ by minimizing the

loss

$$l_1(\boldsymbol{p}_j^i, \boldsymbol{v}_i) = \|1 - \boldsymbol{v}_i(\boldsymbol{p}_j^i)\|. \quad (2)$$

However, we found that it is impossible to pull all the projections into the foreground by minimizing $l_1(\boldsymbol{p}_j^i, \boldsymbol{v}_i)$ in Eq. (2). As an example illustrated in Fig. 3, we optimize a point cloud according to a silhouette image in Fig. 3(a). Starting from randomly initialized points, whose projections are colored as red diamonds in Fig. 3(b), the poorly optimized points are projected in Fig. 3(c).

This problem occurs because the gradient of the loss in Eq.(2) is merely from the pixel intensity difference between 1 and the pixel value $\boldsymbol{v}_i(\boldsymbol{p}_j^i)$ interpolated from the four nearby binary pixel values. This prohibits training if the projections $\boldsymbol{p}_j^i$ are far from the foreground, which would be a local minima. This issue also exists in motion estimation (Bergen et al., 1992; Garg et al., 2016; Godard et al., 2017; Zhou et al., 2017). Since these methods work on real images which contain large variability in texture, a widely used solution to resolve this issue is to employ an explicit multi-scale and smoothness loss that derives gradients from larger spatial regions directly. However, this solution cannot resolve our issue, since silhouette images have no texture variability in the background, which is still impossible to derive gradients even from larger spatial regions.

To resolve this issue, we introduce a smoothing procedure for the pixel values on the ground truth silhouette images $\boldsymbol{v}_i$ that we then use in the unary loss. The key idea behind our silhouette smoothing approach is to establish a progressively varying field in the background on $\boldsymbol{v}_i$, which leads to non-zero gradients anywhere in the background, while the pixel values in the foreground are not changed. We achieve this using the negative distance function as the smoothed values for each pixel $\boldsymbol{x}$ on the background in $\boldsymbol{v}_i$,

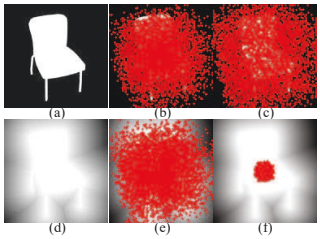

Figure 3. Visualization of the smooth silhouette loss. With binary pixel values in silhouette image $\boldsymbol{v}_i$ in (a), randomly initialized projections in (b) cannot be pulled into the foreground, as shown in (c). Using the smooth silhouette loss based on $\boldsymbol{v}_i^G$ in (d), the projections in (e) can be pulled into the foreground in (f).

$$\boldsymbol{v}_i^G(\boldsymbol{x}) = \begin{cases} 1, & \boldsymbol{x} \in \Omega \\ 1 - d(\boldsymbol{x}, \partial\Omega), & \boldsymbol{x} \in \bar{\Omega} \end{cases} \quad (3)$$

where $\Omega = \{\boldsymbol{x}|\boldsymbol{v}_i(\boldsymbol{x}) = 1\}$ is the foreground, $\bar{\Omega} = \{\boldsymbol{x}|\boldsymbol{v}_i(\boldsymbol{x}) = 0\}$ is the background, $\partial\Omega$ is the boundary of the foreground, and $d(\boldsymbol{x}, \partial\Omega)$ is the L2 distance between $\boldsymbol{x}$ and its nearest $\partial\Omega$, which is normalized by the resolution of $\boldsymbol{v}_i$. We denote the smoothed silhouette images as $\boldsymbol{v}_i^G$ to distinguish them from the original silhouette images $\boldsymbol{v}_i$. Moreover, we normalize the smoothed pixel values in the background to lie in a range of 0 to 1 by minmax normalization, such that $\boldsymbol{v}_i^G(\bar{\Omega}) = minmax(\boldsymbol{v}_i^G(\bar{\Omega}))$. Finally, based on Eq. (2), we define the smoothed silhouette loss as the following unary loss,

$$l_1(\boldsymbol{p}_j^i, \boldsymbol{v}_i) = \|1 - \boldsymbol{v}_i^G(\boldsymbol{p}_j^i)\|. \quad (4)$$

We illustrate this loss using the example in Fig. 3. With the smooth silhouette values in Fig. 3(d), we can pull the projections of all randomly initialized points in Fig. 3(e) into the foreground by minimizing Eq. (4), as shown in Fig. 3(f).

Structure-aware Repulsion Loss. The smooth silhouette loss is far from achieving our goal of uniformly locating projections in the foreground, as shown in Fig. 3(f). To better cover the silhouette and more accurately capture the 3D shape, DRWR includes a pairwise loss to model the spatial relationship between each pair of projections. We design the pairwise loss such that minimizing it pushes projections inside the foreground far away from each other.

However, this pairwise loss is somewhat in conflict with the smooth silhouette loss, which tends to pull all projections close together, especially near the foreground boundaries. In addition, near the foreground boundaries it is harder than deep inside the foreground to push two projections away from each other without pushing them into the background. To resolve this issue, we propose a structure-aware repulsion loss as a pairwise term. This structure-aware repulsion loss adaptively weighs the repulsion between each pair of projections according to the structure around the projections. It increases repulsion for pairs of projections deep inside the foreground, reduces repulsion for pairs of projections around the foreground boundary, and cancels repulsion if any projection is in the background.

Specifically, for each pair of projections $\boldsymbol{p}_j^i$ and $\boldsymbol{p}_{j'}^i$, the L2 distance between them is $d(\boldsymbol{p}_j^i, \boldsymbol{p}_{j'}^i) = \|\boldsymbol{p}_j^i - \boldsymbol{p}_{j'}^i\|_2$, which we further normalize according to the resolution of the silhouette image. DRWR aims to maximize the distance $d(\boldsymbol{p}_j^i, \boldsymbol{p}_{j'}^i)$, and we employ a Gaussian function to obtain a repulsion loss that decreases with increasing distance. Hence we can minimize the repulsion loss along with the smooth silhouette loss. For each projection $\boldsymbol{p}_j^i$, the structure-aware repulsion loss models its spatial relationships to all the other projections

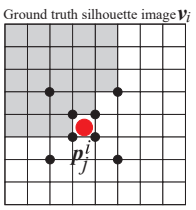

Figure 4. Multi-scale bilinear interpolation for boundary bias δ_j^i.

$\boldsymbol{p}^i_{j'}$ as follows,

$$l_2(\boldsymbol{p}^i_j, \{\boldsymbol{p}^i_{j'}\}, \boldsymbol{v}_i) = \omega^i_j \sum_{j'=1}^{J} \omega^i_{j'} e^{(-d(\boldsymbol{p}^i_j, \boldsymbol{p}^i_{j'})/\sigma + \delta^i_j)}, \quad (5)$$

where ω^i_j and $\omega^i_{j'}$ are indicator weights for projections $\boldsymbol{p}^i_j$ and $\boldsymbol{p}^i_{j'}$, σ is the decay parameter, and δ^i_j is the boundary bias for projection $\boldsymbol{p}^i_j$. The indicator weight ω^i_j represents the degree to which projection $\boldsymbol{p}^i_j$ is located in the background. Small ω^i_j or $\omega^i_j = 0$ decreases or completely removes the repulsion on $\boldsymbol{p}^i_j$, so that the smooth silhouette loss can pull $\boldsymbol{p}^i_j$ into the foreground immediately. The decay parameter σ controls the range of repulsion. The boundary bias δ^i_j controls the distance to the foreground boundary where the repulsion on projection $\boldsymbol{p}^i_j$ is reduced.

Figure 5. Result with both losses.

We compute the indicator weight ω^i_j using bilinear interpolation from nearby binary pixel values on the silhouette image $\boldsymbol{v}_i$, such that $\omega^i_j = \boldsymbol{v}_i(\boldsymbol{p}^i_j)$, as shown in Fig. 2(a). In addition, we employ a multi-scale bilinear interpolation approach to obtain the boundary bias δ^i_j. We use binary pixel values from square neighborhoods at R scales around projection $\boldsymbol{p}^i_j$ to conduct the interpolation, and take the mean of the interpolations from all R scales as δ^i_j. This is demonstrated in Fig. 4, where $R = 2$ scales around a projection (in red) are shown. In this example, some pixels (in black) on the two scales lie in the background (the shaded area), so the boundary bias δ^i_j is small, which accordingly reduces the repulsion on the projection. This approach enables DRWR to progressively decrease repulsion when $\boldsymbol{p}^i_j$ approaches the foreground boundary. As shown in Fig. 5, the structure-aware repulsion loss combined with the smooth silhouette loss successfully pulls all projections into the foreground (as in Fig. 3(f)), while also uniformly covering the entire foreground area.

Loss function. DRWR minimizes the following overall loss function based on the two losses defined in Eq. (4) and Eq. (5) to conduct unsupervised structure learning for 3D point clouds,

$$L = \frac{1}{I}\frac{1}{J}\sum_{i=1}^{I}\sum_{j=1}^{J}(l_1(\boldsymbol{p}^i_j, \boldsymbol{v}_i) + \beta l_2(\boldsymbol{p}^i_j, \{\boldsymbol{p}^i_{j'}\}, \boldsymbol{v}_i)), \quad (6)$$

where β is a weight to balance the two losses and the total loss L averages over all J points from all I views.

4. Experiments and Analysis

Datasets and metrics. We conduct experiments involving 3D shapes in three categories from ShapeNet (Chang et al., 2015), including chairs, cars, and airplanes, which are commonly used for evaluation by our competitors. We also follow the same train/test splitting as in (Insafutdinov & Dosovitskiy, 2018; Tulsiani et al., 2017b), and we employ the five rendered views from each 3D shape and the ground truth point clouds released by (Insafutdinov & Dosovitskiy, 2018). Specifically, we employ rendered views at three different resolutions including 32^2, 64^2, and 128^2, all corresponding to the same set of ground truth point clouds with different numbers of points.

We conduct numerical comparisons using the Chamfer distance (CD) between predicted and ground truth point clouds. For differentiable renderers for meshes or voxel grids, we also use volumetric IoU to conduct fair comparisons. Note that all reported CD or IoU values reported in our experiments are multiplied by 100 for better readability.

Details. For fair comparison, we employ exactly the same neural network architecture as the one introduced by Insafutdinov et al. (Insafutdinov & Dosovitskiy, 2018), however, replacing the differentiable renderer (Insafutdinov & Dosovitskiy, 2018) with DRWR. The approach by Insafutdinov et al. (Insafutdinov & Dosovitskiy, 2018) implements structure learning of 3D point clouds using pairs of RGB images. For each pair, the network first generates a point cloud from the first RGB image, and then renders the predicted point cloud from the view angle of the second image. Their differentiable renderer produces a rendered silhouette image as its output, as shown in Fig. 1(a), and the neural network is trained by minimizing the pixel-wise error between the rendered silhouette image and the silhouette of the second input image. In our approach, we replace their rendering-based differentiable renderer by DRWR, as shown in Fig. 1(b). We omit rendering and simply leverage the projected positions of the generated point clouds to produce the loss and gradient required in the training. At test time, the trained network generates a 3D point cloud from a single RGB image.

We evaluate DRWR using this network with ground truth camera poses during projection. In addition, we employ RGB images with three different resolutions to train, and accordingly evaluate the generated point clouds in three resolutions including 2000, 8000, and 16000 points. We train the network using the Adam optimizer with a batch size of 16 rendered images (4 views of 4 shapes), where we iterate over 1×10^5 batches in each experiment. We set $R = 5$ to calculate the boundary bias δ^i_j for all projections during the optimization and set the decay parameter $\delta = 1$ to decrease the repulsion between each pair of projections. We balance the two losses using $\beta = 3$ in all our experiments.

4.1. Comparison with the State-of-the-art

CD comparisons. We first compare DRWR with rendering-based differentiable renderers in terms of CD. All compared renderers produce silhouettes of the predicted shapes to compute their loss with respect to ground truth silhouettes. We conduct the comparison by training the networks using silhouette images at three different resolutions as mentioned previously.

We compare DRWR with Differentiable Ray Consistency (DRC) (Tulsiani et al., 2017b), Efficient Point Cloud Generation (EPCG) (Lin et al., 2018), Continuous Approximation Projection (CAP) (L. et al., 2019), and Differentiable Point Clouds (DPC) (Insafutdinov & Dosovitskiy, 2018). The first renderer is voxel-based, and it is only available for voxel grids with a resolution of 32^3 because of the cubic complexity of voxel grids. The other three renderers are point cloud-based, which is the same as DRWR.

We report the quantitative comparison in Table 1. Our results outperform all compared methods under all classes at all three resolutions. DRWR shows obvious advantages over voxel-based differentiable renderers including DRC and the voxel-based counterpart "DPC-V" of DPC, where DRWR recovers more geometry details in a more memory-efficient manner. In addition, by omitting rendering, DRWR also achieves higher accuracies of the reconstructed point clouds compared to rendering-based differentiable point renderers, such as CAP, DPC, and EPCG. These results further demonstrate that DRWR is robust to changes in image resolutions and number of points.

Fig. 8 shows a qualitative comparison with the point cloud renderers used in CAP and DPC at a resolution of 2000 points. We find that DRWR generalizes better for rarely seen shapes and achieves higher accuracies by more uniformly distributing the recovered 3D points. In addition, we visualize more high fidelity shapes at resolutions of 2000, 8000, and 16000 points in Fig. 9 (a), (b) and Fig. 7, respectively. We find that DRWR can train networks to generate plausible point clouds with different numbers of points from images, while DRWR would recover more geometry details when using more points to represent a shape.

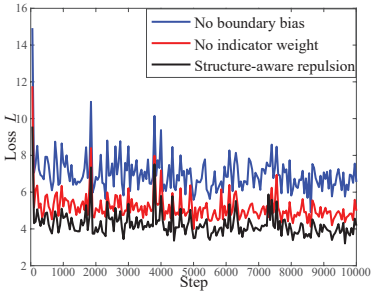

Figure 6. The efficiency of structure-aware repulsion.

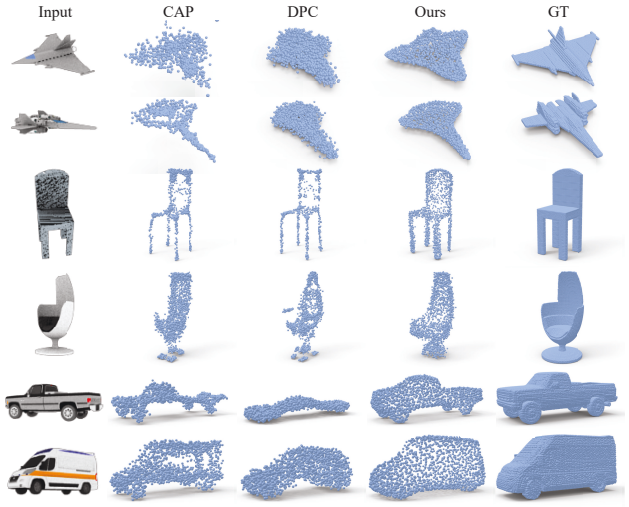

Figure 8. Qualitative comparison with differentiable renderers for point clouds.

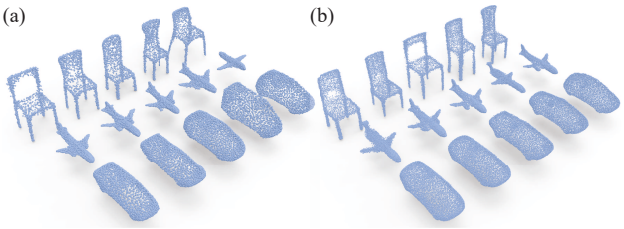

Figure 9. Randomly selected shapes with 2000 points in (a) and 8000 points in (b) from reconstructed shapes producing Table 1.

IoU comparisons. We further compare DRWR in terms of IoU with rendering-based differentiable renderers for other 3D representations, such as meshes and voxel grids. The comparison includes Perspective Transform Nets (PTN) (Yan et al., 2016), Neural Mesh Renderer (NMR) (Kato et al., 2018), SoftRasterizer (SoftR) (Liu et al., 2019b), and Interpolation-based Differentiable Renderer (DIB-R) (Chen et al., 2019). The former two methods are voxel-based, while the latter two methods are mesh-based. We report the results of NMR, SoftR and DIB-R from (Chen et al., 2019). To produce our IoU, we voxelize the point clouds predicted from images with a resolution of 128^2 in Table 1 into voxel grids with resolution 32^3 to compare to the same ground truth as other methods.

The quantitative comparison is shown in the "unsupervised" part of Table 2. Our results significantly outperform the state-of-the-art differentiable renderers in terms of mean IoU, where we achieve the best under airplanes and chairs. For cars, our results are better than voxel-based renderers but only comparable to mesh-based renderers. This is because

Table 1. Comparison with point clouds renderers in terms of CD.

	Resolution 32 (2000)					Resolution 64 (8000)			Resolution 128 (16000)		
	DRC	CAP	DPC-V	DPC	Ours	DPC-V	DPC	Ours	EPCG	DPC	Ours
Plane	8.35	6.34	5.57	4.52	**4.01**	4.94	3.50	**3.18**	4.03	2.84	**2.66**
Car	4.35	6.03	3.88	4.22	**3.81**	3.41	2.98	**2.89**	3.69	2.42	**2.40**
Chair	8.01	6.11	5.57	5.10	**4.66**	4.80	4.15	**4.02**	5.62	3.62	**3.49**
Mean	6.90	6.16	5.01	4.61	**4.16**	4.39	3.55	**3.36**	4.45	2.96	**2.85**

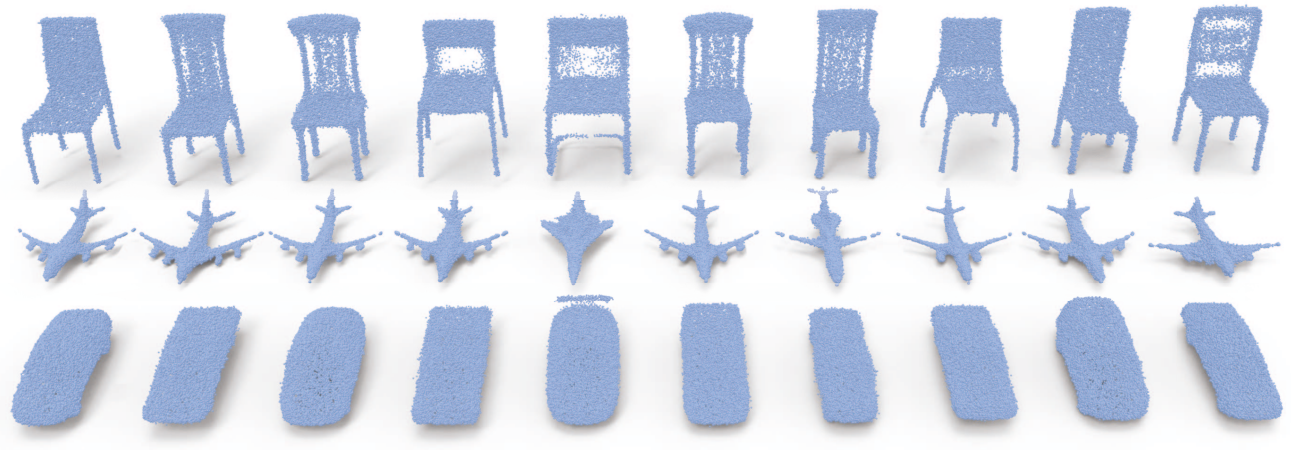

Figure 7. Visualization of randomly selected shapes with 16000 points in single image reconstruction in Table 1.

the mesh-based renderers are good at representing large areas of flat surfaces, such as cars. Mesh-based approaches are limited to a fixed (usually spherical) mesh topology, however. This leads to inaccuracies when representing more complex surfaces, such as chairs, which often exhibit non-spherical topology.

We further conduct qualitative comparisons with the latest differentiable renderers including point cloud-based DSS (Yifan et al., 2019) and mesh-based SoftR in Fig. 10. For fair comparison, we also produce results of DSS from four views which keeps the same as DRWR in training, where DSS uses its own camera system to generate rendered images from ground truth point clouds. In addition, we show our results at a resolution of 16000 points and ground truth involved in Table 1. Compared to DSS and SoftR, DRWR can reveal more geometry details.

Supervised methods. Finally, we compare DRWR with the latest supervised 3D structure learning methods. In the first experiment, we compare with NOX (Sridhar et al., 2019) for point clouds, where we report our results using the evaluation code released by NOX (Sridhar et al., 2019). Following the same setting, we employ the point clouds reconstructed from input images with a resolution of 64^2 in Table 1, scale each predicted point cloud such that the diagonal of its bounding box is one, and resample a ground truth point cloud to 8000 points if there are more than 8000 points in it. Table 3 shows that our results significantly outperform NOX under all three classes.

The "supervised" part in Table 2 reports the numerical comparison with the state-of-the-art supervised methods, where we used the same approach as before to obtain our IoU results. We significantly outperform supervised methods under all three shape categories. Fig. 10 also shows a visual comparison with DISN, the best supervised method, illustrating that we obtain similar quality.

Table 3. Comparison (CD) with latest supervised method NOX.

	Cars	Airplanes	Chairs
NOX	0.3331	0.2795	0.4637
Ours	**0.0446**	**0.0527**	**0.0540**

4.2. Ablation studies and analysis

Ablation studies. We conduct ablation studies to justify the effectiveness of each element in DRWR under airplanes at a resolution of 2000 points in Table. 1. In Table 4, we report our results with only the smooth silhouette loss in Eq. (4) ("l_1"), with only structure-aware repulsion loss in Eq. (5) ("l_2"), with binary pixel loss in Eq. (2) and structure-aware

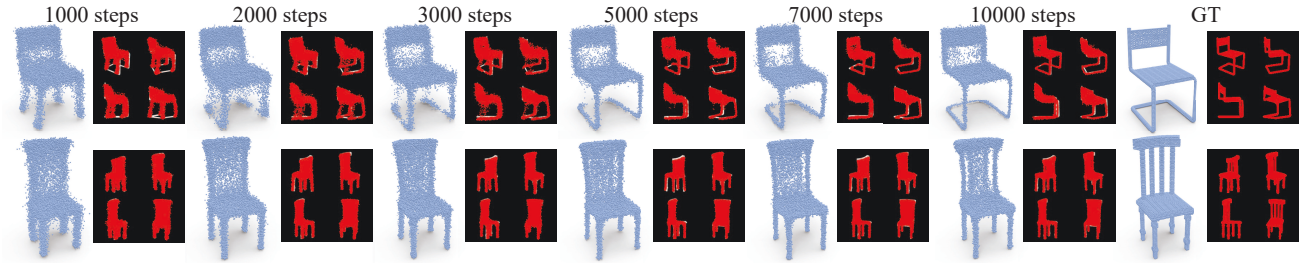

Figure 11. A shape reconstructed from a test image using network parameters in different steps.

Table 2. Quantitative comparison with differentiable renderers for different 3D representations and supervised methods in terms of IoU.

	Unsupervised differentiable renderers					Supervised structure learning methods							
	PTN	NMR	SoftR	DIB-R	Ours	DISN	OccNet	IMNET	3DN	Pix2Mesh	R2N2	AtlasNet	Ours
Car	71.2	71.3	77.1	**78.8**	75.3	74.3	73.7	74.5	59.4	50.1	66.1	22.0	**75.3**
Plane	55.6	58.5	58.4	57.0	**62.2**	57.5	57.1	55.4	54.3	51.5	42.6	39.2	**62.2**
Chair	44.9	41.4	49.7	52.7	**58.1**	54.3	50.1	52.2	34.4	40.2	43.9	25.7	**58.1**
Mean	57.2	57.1	61.7	62.8	**65.2**	62.0	60.3	60.7	49.4	47.3	50.9	29.0	**65.2**

Figure 12. Interpolated shapes in the learned 3D feature space.

repulsion loss ("Pixel+l_2"), with smooth silhouette pixel loss in Eq. (4) and repulsion loss without indicator weights ("l_1+no w_j^i"), and with smooth silhouette loss in Eq. (4) and repulsion loss without boundary bias ("l_1+no δ_j^i"). We also report results with fewer views of each shape in training, such as $I = 2$ and $I = 3$ views.

Table 4. Ablation studies in terms of CD.

	l_1	l_2	Pixel+l_2	l_1+no w_j^i	l_1+no δ_j^i	$I = 2$	$I = 3$	$I = 4$
CD	19.50	139.10	24.59	4.58	4.41	4.79	4.54	**4.01**

The ablation studies show that DRWR cannot learn the structure of shapes using only l_1 or l_2, nor without the smooth silhouette loss because of the local minimum issue. The structure awareness brought by indicator weights and boundary bias also contributes to the reconstruction accuracy and optimization efficiency, which is also demonstrated by the loss L comparison in the first 10000 steps in Fig. 6. The loss comparison shows that the structure awareness significantly decreases the conflict with the smooth silhouette loss, which leads to lower loss and faster convergence. In addition, using fewer views than our $I = 4$ views in training also degenerates the structure learning performance.

Table 5. Efficiency comparison in terms of training time.

	Modality	Rendering	32^2 image 2000 points/ 32^3 voxels	64^2 image 8000 points/ 64^3 voxels	128^2 image 16000 points/ 128^3 voxels
DRC	Voxel	Yes	≈14h	≈60h	Out of memory
DPC	Points	Yes	≈14h	≈24h	≈72h
Ours	Points	No	≈**7**h	≈**12**h	≈**36**h

Efficiency. We highlight the efficiency of DRWR by comparing our network training time with state-of-the-art differentiable renderers for 3D shapes. The voxel-based method of DRC (Tulsiani et al., 2017b) suffers from a huge computation burden due to the cubic complexity of voxel grids, which limits it to work only in low resolutions such as 32^3 and 64^3 with slow convergence. Although the point cloud-based method of DPC (Insafutdinov & Dosovitskiy, 2018) does not require 3D convolution layers as DRC, which improves the efficiency and enables to work in higher resolution, the rendering procedure still requires intensive computation with discrete 3D grids. Therefore, DPC requires more time (6×10^5 mini-batch iterations) during training than DRWR (1×10^5 mini-batch iterations), thanks to the removal of rendering.

Optimization. We visualize the optimization process in

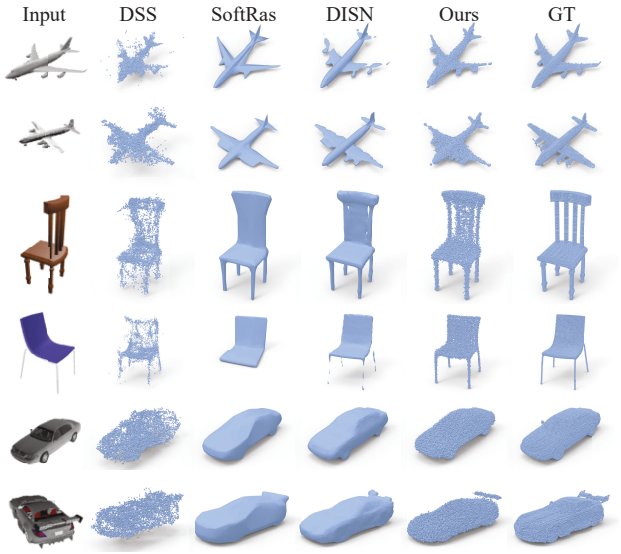

Figure 10. Qualitative comparison with differentiable renderers for different 3D representations and supervised learning methods.

Figure 13. Qualitative demonstration of shapes reconstructed from real images.

Fig. 11. We use the parameters learned in different steps during training to reconstruct a shape from a corresponding image in the test set. In addition, we show the 2D projections on four views. Using an image from the test rather than training set can better demonstrate the generalization ability learned in optimization, justifying our effectiveness.

Adaptation to real images. We evaluate the adaptation ability to real images in the network trained using DRWR in Fig. 13. Using the parameters learned in Table 1, we reconstruct shapes at a resolution of 16000 points from real images selected from the Internet. The high fidelity of reconstructed shapes demonstrates that DRWR can train networks to adapt to real images very well.

Latent space. We visualize the latent space learned in the network that we train using DRWR. We employ a trained network to reconstruct shapes using 1024-dimensional latent codes that are interpolated from two known codes of two shapes. As shown in Fig. 12, the interpolated shapes in the smooth transformation show that DRWR helps the network to learn a meaningful latent space.

5. Conclusion

We propose DRWR for unsupervised 3D structure learning using point clouds. DRWR successfully removes the rendering step that is commonly used in state-of-the-art differentiable renderers. While rendering requires additional computation, a key observation from our experiments is that it does not contribute to improving accuracy in 3D structure learning. DRWR achieves this by minimizing a unary and a pairwise loss. The unary loss uses a smooth silhouette loss to pull all projections into the foreground by effectively resolving the severe local minimum issue, while the pairwise loss uses structure-aware repulsion to efficiently push pairs of projections in the foreground away from each other by adaptively weighting the repulsion according to the 2D structure. The effectiveness of DRWR is justified by superior experimental results over the state-of-the-art.

Acknowledgements

We thank the anonymous reviewers for reviewing our paper and providing helpful comments. This work was supported by National Key R&D Program of China (2018YFB0505400), in part by Tsinghua-Kuaishou Institute of Future Media Data, and NSF (award 1813583).

References

Bergen, J. R., Anandan, P., Hanna, T. J., and Hingorani, R. Hierarchical model-based motion estimation. pp. 237–252, 1992.

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

Chang, A. X., Funkhouser, T. A., Guibas, L. J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. ShapeNet: An information-rich 3D model repository. *CoRR*, abs/1512.03012, 2015.

Chen, W., Gao, J., Ling, H., Smith, E. J., Lehtinen, J., Jacobson, A., and Fidler, S. Learning to predict 3D objects with an interpolation-based differentiable renderer. *CoRR*, abs/1908.01210, 2019.

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

Gadelha, M., Maji, S., and Wang, R. 3D shape induction

from 2D views of multiple objects. In *International Conference on 3D Vision*, pp. 402–411, 2017.

Garg, R., Kumar, B. V., Carneiro, G., and Reid, I. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pp. 740–756, 2016.

Godard, C., Mac Aodha, O., and Brostow, G. J. Unsupervised monocular depth estimation with left-right consistency. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Han, Z., Liu, Z., Han, J., Vong, C.-M., Bu, S., and Li, X. Unsupervised 3D local feature learning by circle convolutional restricted boltzmann machine. *IEEE Transactions on Image Processing*, 25(11):5331–5344, 2016.

Han, Z., Liu, Z., Han, J., Vong, C.-M., Bu, S., and Chen, C. 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, 2017a.

Han, Z., Liu, Z., Vong, C.-M., Liu, Y.-S., Bu, S., Han, J., and Chen, C. BoSCC: Bag of spatial context correlations for spatially enhanced 3D shape representation. *IEEE Transactions on Image Processing*, 26(8):3707–3720, 2017b.

Han, Z., Liu, Z., Vong, C.-M., Liu, Y.-S., Bu, S., Han, J., and Chen, C. P. 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.

Han, Z., Liu, X., Liu, Y.-S., and Zwicker, M. Parts4Feature: Learning 3D global features from generally semantic parts in multiple views. In *IJCAI*, 2019a.

Han, Z., Liu, Z., Han, J., Vong, C.-M., Bu, S., and Chen, C. Unsupervised learning of 3D local features from raw voxels based on a novel permutation voxelization strategy. *IEEE Transactions on Cybernetics*, 49(2):481–494, 2019b.

Han, Z., Lu, H., Liu, Z., Vong, C.-M., Liu, Y.-S., Zwicker, M., Han, J., and Chen, C. P. 3D2SeqViews: Aggregating sequential views for 3D global feature learning by cnn with hierarchical attention aggregation. *IEEE Transactions on Image Processing*, 28(8):3986–3999, 2019c.

Han, Z., Shang, M., Liu, Z., Vong, C.-M., Liu, Y.-S., Zwicker, M., Han, J., and Chen, C. P. SeqViews2SeqLabels: Learning 3D global features via aggregating sequential views by rnn with attention. *IEEE Transactions on Image Processing*, 28(2):685–672, 2019d.

Han, Z., Shang, M., Wang, X., Liu, Y.-S., and Zwicker, M. Y2Seq2Seq: Cross-modal representation learning for 3D shape and text by joint reconstruction and prediction of view and word sequences. In *AAAI*, pp. 126–133, 2019e.

Han, Z., Wang, X., Liu, Y.-S., and Zwicker, M. 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 *ICCV*, 2019f.

Han, Z., Wang, X., Vong, C.-M., Liu, Y.-S., Zwicker, M., and Chen, C. P. 3DViewGraph: Learning global features for 3D shapes from a graph of unordered views with attention. In *IJCAI*, 2019g.

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

Han, Z., Qiao, G., Liu, Y.-S., and Zwicker, M. SeqXY2SeqZ: Structure learning for 3D shapes by sequentially predicting 1D occupancy segments from 2D coordinates. *ArXiv*, abs/2003.05559, 2020b.

Hu, T., Han, Z., Shrivastava, A., and Zwicker, M. Render4Completion: Synthesizing multi-view depth maps for 3D shape completion. *ArXiv*, abs/1904.08366, 2019.

Hu, T., Han, Z., and Zwicker, M. 3D shape completion with multi-view consistent inference. In *AAAI*, 2020.

Insafutdinov, E. and Dosovitskiy, A. Unsupervised learning of shape and pose with differentiable point clouds. In *Advances in Neural Information Processing Systems*, pp. 2807–2817, 2018.

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

Kato, H., Ushiku, Y., and Harada, T. Neural 3D mesh renderer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3907–3916, 2018.

L., N. K., Mandikal, P., Agarwal, M., and Babu, R. V. Capnet: Continuous approximation projection for 3d point cloud reconstruction using 2d supervision. *AAAI*, 2019.

Lin, C.-H., Kong, C., and Lucey, S. Learning efficient point cloud generation for dense 3d object reconstruction. In *AAAI Conference on Artificial Intelligence*, 2018.

Liu, H.-T. D., Tao, M., and Jacobson, A. Paparazzi: Surface editing by way of multi-view image processing. *ACM Transactions on Graphics*, 2018.

Liu, H.-T. D., Tao, M., Li, C.-L., Nowrouzezahrai, D., and Jacobson, A. Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer. In *International Conference on Learning Representations*, 2019a.

Liu, S., Li, T., Chen, W., and Li, H. Soft rasterizer: A differentiable renderer for image-based 3D reasoning. *The IEEE International Conference on Computer Vision*, 2019b.

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

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

Liu, X., Han, Z., Liu, Y.-S., and Zwicker, M. Point2Sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network. In *AAAI*, pp. 8778–8785, 2019d.

Liu, X., Han, Z., Xin, W., Liu, Y.-S., and Zwicker, M. L2G auto-encoder: Understanding point clouds by local-to-global reconstruction with hierarchical self-attention. In *ACMMM*, 2019e.

Loper, M. M. and Black, M. J. OpenDR: An approximate differentiable renderer. In *European Conference on Computer vision*, volume 8695, pp. 154–169, 2014.

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

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

Navaneet, K. L., Mandikal, P., Jampani, V., and Babu, R. V. DIFFER: Moving beyond 3D reconstruction with differentiable feature rendering. In *CVPR Workshops*, 2019.

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

Oechsle, M., Mescheder, L., Niemeyer, M., Strauss, T., and Geiger, A. Texture fields: Learning texture representations in function space. 2019.

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

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. PointNet: Deep learning on point sets for 3D classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017a.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pp. 5105–5114, 2017b.

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

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

Sridhar, S., Rempe, D., Valentin, J., Bouaziz, S., and Guibas, L. J. Multiview aggregation for learning category-specific shape reconstruction. In *Advances in Neural Information Processing Systems*. 2019.

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

Tulsiani, S., Su, H., Guibas, L. J., Efros, A. A., and Malik, J. Learning shape abstractions by assembling volumetric primitives. In *IEEE Computer Vision and Pattern Regognition*, 2017a.

Tulsiani, S., Zhou, T., Efros, A. A., and Malik, J. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 209–217, 2017b.

Tulsiani, S., Efros, A. A., and Malik, J. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *Computer Vision and Pattern Regognition*, 2018.

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

Wen, X., Han, Z., Youk, G., and Liu, Y.-S. CF-SIS: Semantic-instance segmentation of 3D point clouds by context fusion with self-attention. In *ACM International Conference on Multimedia*, 2020a.

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

Yan, X., Yang, J., Yumer, E., Guo, Y., and Lee, H. Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In *Advances in Neural Information Processing Systems*, pp. 1696–1704. 2016.

Yifan, W., Serena, F., Wu, S., Öztireli, C., and Sorkine-Hornung, O. Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics*, 38(6):1–14, 2019.

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

Zhou, T., Brown, M., Snavely, N., and Lowe, D. G. Unsupervised learning of depth and ego-motion from video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6612–6619, 2017.