DIFFER: Moving Beyond 3D Reconstruction with Differentiable Feature Rendering

K L Navaneet[1], Priyanka Mandikal[1], Varun Jampani[2], and R. Venkatesh Babu[1]

[1]Video Analytics Lab, CDS, Indian Institute of Science, [2]NVIDIA

Abstract

Perception of 3D object properties from 2D images form one of the core computer vision problems. In this work, we propose a deep learning system that can simultaneously reason about 3D shape as well as associated properties (such as color, semantic part segments) directly from a single 2D image. We devise a novel depth-aware differentiable feature rendering module (DIFFER) that is used to train our model by using only 2D supervision. Experiments on both synthetic ShapeNet dataset and the real-world Pix3D dataset demonstrate that our 2D supervised DIFFER model performs on par or sometimes even outperforms existing 3D supervised models.

1. Introduction

The world we live in is composed of illuminated physical objects with diverse shapes, sizes, textures, and surface information. We, as humans, are capable of processing the retinal image of an object to decipher the underlying 3D structure. Our 3D perception capabilities go beyond mere reconstruction of structural information. We are highly adept at capturing a variety of other 3D properties such as texture, part information, surface normals, etc.

Like humans, machines require 3D perception to perform real world tasks. The 3D perception of machines need to go beyond just the shape reconstruction from 2D images. For instance, semantic understanding of the perceived 3D object is particularly advantageous in tasks such as robot grasping, object manipulation, etc. Further, the ability to effectively colorize a 3D model has applications in creative tasks such as model designing, texture mapping, etc. Thus, an ideal machine would have the capacity to infer both the three-dimensional structure as well as associated features given a single 2D image (Fig. 1).

In this work, we aim to design a deep learning system that can simultaneously predict 3D shape (in the form of point cloud) of an object while also predicting important 3D point characteristics such as color and part segmentation. However, training systems capable of performing a multitude of 3D perception tasks poses several challenges: **(1)** 3D data required for training such systems is not easy to acquire. There is a lack of large-scale ground truth 3D annotations for in-the-wild images. Existing datasets with accurate 3D annotations are either synthetically created [1] or are captured in constrained environments requiring elaborate procedures using multiple sensors and scanners [18]. **(2)** Models trained on synthetic datasets do not generalize well to the real-world images due to differences in the input data distributions. These challenges necessitate learning techniques that rely on easily available 2D images as supervision instead of 3D ground truth.

Utilizing 2D data as supervision for 3D perception network requires a differentiable rendering module that can effectively propagate gradients from the rendered 2D image back to the predicted 3D model. Since our task is to learn both 3D structure and features, this module would need to be generic enough to render any feature that is associated with a 3D model. Towards this end, we design a *depth-aware* feature expectation formulation, where 3D point features are effectively rendered onto a 2D surface based on the depth value of the corresponding points. Such a mechanism allows us to obtain accurate projections of the predicted 3D features.

In summary, our contributions are as follows:

- We propose a differentiable point feature rendering module named DIFFER to train single-view 3D point cloud reconstruction and feature prediction using only 2D supervision. Being depth-aware, DIFFER can effectively render a diverse set of features such as color, part segmentation and surface normals, thus enabling the training of 3D feature learning systems using weak supervision.
- We benchmark our approach on both synthetic (ShapeNet [1]) and real-world (Pix3D [18]) datasets. Extensive quantitative and qualitative evaluations show that DIFFER performs comparably or even better than approaches that use full 3D supervision.

2. Related Works

3D Reconstruction Existing approaches to 3D reconstruction from single-view images predominantly use full 3D supervision. Voxel based methods predict a full 3D occupancy grid using 3D CNNs [4, 2, 21]. However, voxel formats are information-sparse since meaningful structural information is mainly provided by the surface voxels. 3D CNNs are also compute heavy and add considerable overhead during training and inference. More recent works have introduced techniques for predicting unordered 3D point clouds [3, 10]. Point clouds offer the advantage of being information-rich, since points are sampled only on the surface, and require lighter compute units for processing. In this work, we compare against [3], which introduced framework and loss formulations tailored for training point cloud generators using 3D ground truth supervision, and obtained superior single-view reconstruction results compared to volumetric approaches [2]. We show competitive performance using only 2D data as supervision. Works such as [22, 19, 20, 24, 9, 13, 5, 8] explore ways to reconstruct 3D shapes from 2D projections such as silhouettes and depth maps. Yan et al. [22] obtain 2D masks by performing perspective transformation and grid sampling of voxel outputs. Tulsiani et al. [19] use differentiable ray consistency to train on 2D observations like foreground mask, depth and color images. Lin et al. [9] pre-train a network by directly regressing depth maps from eight fixed views, which are fused to obtain the point cloud. This is followed by a network fine-tuning via a depth projection loss. The works of [13] and [5] project reconstructed 3D point clouds using a differentiable point cloud renderer to obtain 2D masks during supervision. While existing differentiable point cloud rendering modules are able to render masks or depth maps, our proposed module is capable of rendering arbitrary features associated with the 3D model. Contrasting to [5], which predicts color along with shape reconstruction, our network jointly predicts shape, parts and color reconstruction and we show quantitative results on all of them.

3D Feature Prediction 3D feature learning involves predicting 3D features such as semantics or color. Semantic segmentation using neural networks has been explored by several works [16, 14, 15, 6, 12, 11, 17]. [16] estimate voxel occupancy as well as part labels for 3D scenes from depth maps. [14, 15] introduce networks that perform point cloud classification and segmentation. [11] train a network that jointly estimates shape and part segmentation. While these works require 3D part labels as ground truth, we show competitive performance using only 2D annotations.

3. Approach

We develop a deep learning framework for joint 3D point cloud reconstruction and general feature prediction that uses only 2D supervision The predicted 3D point features can be color (RGB), part segmentation labels or surface normals. To this end, we propose a novel depth-aware differentiable renderer to obtain the corresponding 2D feature projections from the 3D predictions of the network (Fig. 1). The network training objectives for each feature are formulated in the 2D domain. We extend the 2D mask projection formulation provided by Navaneet et al. [13] (CAP-Net) to general feature projection of 3D point cloud from a given viewpoint. Consider an input image I. We predict (x, y, z) co-ordinates of point cloud $P' \in \mathbb{R}^{N\times 3}$ along with $k-$dimensional features $\hat{F} \in \mathbb{R}^{N\times k}$ using an encoder-decoder architecture based network (Fig. 1). Assuming the knowledge of intrinsic camera parameters and view-point v, a perspective transformed point cloud $\hat{P} = (\hat{x}, \hat{y}, \hat{z}) \in \mathrm{R}^{N\times 3}$ is obtained. Let $\hat{M}^v$ be the mask obtained by orthogonally projecting $\hat{P}$ from view point v. Then the value of mask at pixel index (i, j) is obtained as

$$\hat{M}^v_{i,j} = \tanh\left(\sum_{n=1}^{N} \phi(\hat{x}_n - i)\phi(\hat{y}_n - j)\right), \qquad (1)$$

where $\phi(\cdot)$ is an un-normalized Gaussian kernel. The above differentiable rendering formulation is proposed in CAP-Net [13] and has no occlusion reasoning. It can only be used to obtain mask supervision where self-occlusions do not matter. Renderings of GT parts and color using CAP-Net shown in fig. 2 indicate that the feature projections do not account for occlusions. This makes it unsuitable for training general feature prediction networks.

Depth-aware general feature projection The above projection formulation (Eq. 1) is independent of the depth of the points. However, for a general feature associated with the points, their relative depths determine which of the points is projected to a particular 2D location. For a given 2D location, the point with the lowest depth value would be visible while the rest of the points in the same line of sight would be occluded and hence, not projected onto the 2D map. Thus, it is necessary to obtain a depth map in order to project any feature value. While the points corresponding to the minimum depth values can directly be used to acquire the depth maps, the resulting method is not differentiable. In this work, we propose a differentiable approximation to obtain the depth values and subsequently project features from a point cloud in a differentiable manner. Let $\hat{d}^{n,v}_{i,j}$ be the depth value obtained at location (i, j) by projecting point n (Eq. 2).

$$\hat{d}^{n,v}_{i,j} = \psi(\hat{x}_n - i)\psi(\hat{y}_n - j)\hat{z}_n \qquad (2)$$

The kernel function ψ for depth projection is defined as:

$$\psi(k) = \begin{cases} 1, & -r \leq k \leq r \\ 10, & \text{elsewhere} \end{cases} \qquad (3)$$

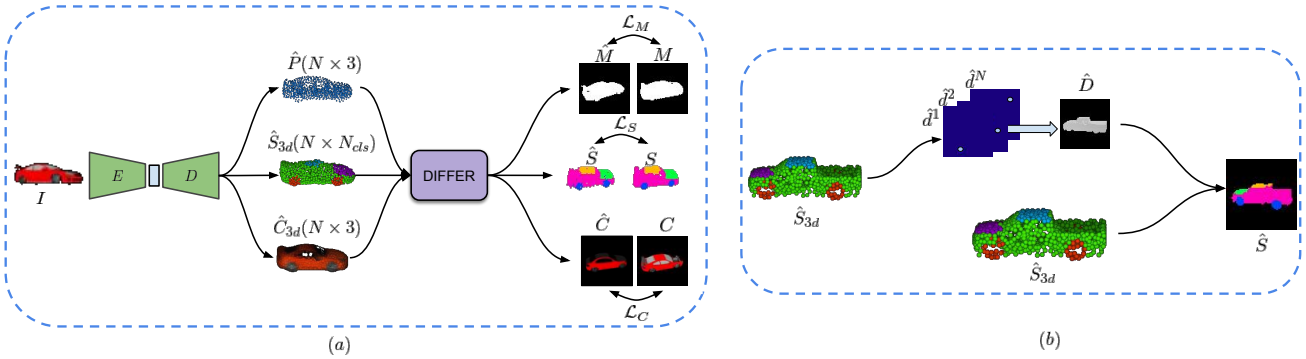

Figure 1: DIFFER module for feature reconstruction. We propose a differentiable point feature renderer for reconstructing point clouds with associated features from just a single input image. (a) The network predicts features like part-segmentation and point color in addition to the 3D shape. DIFFER is used to obtain 2D projection maps(eg. mask, color image and part-segmentation map) from the predicted point cloud. The network is trained with 2D supervisory data. (b) DIFFER predicts projection probability values as a function of depth for each point in the prediction. The 2D feature map is obtained as an expectation of point feature values.

where r is the width of the kernel, referred hereafter as "well-radius". The kernel determines the points in the vicinity of the projected pixel and the point with the least depth amongst them is selected as the point to be projected. The well-radius regulates the smoothness and accuracy of the depth maps. While a low value results in sparse projections, a very high value results in inaccurate outputs.

We use the depth values obtained by the above formulation to project any general 3D point features onto 2D images. We define the probability of the point n being projected on to the pixel (i, j), $\hat{p}_{i,j}^{n,v}$, as:

$$\hat{p}_{i,j}^{n,v} = \exp\left(\frac{1}{\hat{d}_{i,j}^{n,v}}\right) \Big/ \left(\sum_{k=1}^{N} \exp\left(\frac{1}{\hat{d}_{i,j}^{k,v}}\right)\right). \tag{4}$$

The probability of a point being projected depends on the depth of the point and the presence of other points in the same line-of-sight. Lower the depth value of a point, higher is its probability of projection. To model this, we consider the probability of projection to be inversely proportional to the depth value of the point. The softmax normalization approximately models the influence of other points. Once the point projection probabilities are determined, the final feature projection at a specific pixel is obtained as the expected feature value at that location, $\hat{F}_{i,j}^{v} = \sum_{n=1}^{N} \hat{p}_{i,j}^{n,v} \hat{f}^{n}$.

We refer to this differentiable feature renderer as ''DIFFER''. In the case of DIFFER, a simple *depth-aware* rendering (Eqns. 2- 4) can mimic complex occlusion reasoning resulting in an effective differentiable renderer for general feature projection. Fig. 2 shows that DIFFER part/color projections closely resemble GT parts/colors demonstrating the importance of depth-aware rendering for feature projection. The above formulation can be extended to other general features. We show experimental results on surface normal prediction in the supplementary.

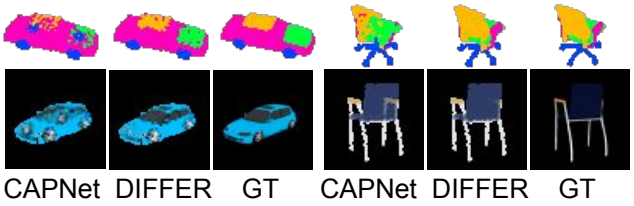

Figure 2: Importance of depth-aware rendering: Projected part segmentation and color maps for CAPNet [13] and DIFFER.

3.1. Loss Formulation

Mask Projection The per-pixel binary cross-entropy loss between ground truth mask M^v and projection $\hat{M}^v$ from view-point v is obtained as:

$$\mathcal{L}_M^v = \frac{1}{HW} \sum_{i,j} -M_{i,j}^v \log \hat{M}_{i,j}^v - (1 - M_{i,j}^v) \log(1 - \hat{M}_{i,j}^v) \tag{5}$$

where H, W are the height and width of the projected image respectively.

Color Projection The point cloud color is represented as a 3-channel RGB value associated with each point, i.e $P' = (X, Y, Z, R, G, B)$. Once the network predicts the 3D point locations along with their color, we use the DIFFER module to project 3D point colors on to the 2D image grid. We use the mean squared error between the ground truth C^v and the projected color image, $\hat{C}^v$, as a loss function to train our network:

$$\mathcal{L}_C^v(C^v, \hat{C}^v) = \frac{1}{HW} \sum_{i,j} ||C_{i,j}^v - \hat{C}_{i,j}^v||^2. \tag{6}$$

Semantic Part Projection A part label is associated with every point in the point cloud. The label values are discrete, and hence cannot be directly used to obtain projections. We

represent the labels using one-hot encoding and our 3D network is trained to predict the probability of a point belonging to each of the classes. We treat the probability values as features and project them onto 2D using DIFFER. During inference, the label with the maximum probability is treated as the prediction. We use pixel-wise softmax cross-entropy loss between the ground truth S^v and the projected class probabilities $\hat{S}^v$ for the training:

$$\mathcal{L}_S^v = \frac{1}{HW} \sum_{i,j} S_{i,j}^v \log(\hat{S}_{i,j}^v) + (1 - S_{i,j}^v) \log(1 - \hat{S}_{i,j}^v) \quad (7)$$

The total loss is obtained as a combination of feature and mask loss averaged over all viewpoints V,
$\mathcal{L}_{tot} = (1/V) \sum_{v=1}^{V} (\mathcal{L}_M^v + \lambda \mathcal{L}_{feat}^v)$ where $\mathcal{L}_{feat}^v$ can be either $\mathcal{L}_C^v$ or $\mathcal{L}_S^v$ and λ is the relative weight for feature loss. Details for loss formulation for surface normal prediction are provided in the supplementary.

4. Experiments

4.1. Implementation details

We use four random viewpoint projections (for 2D supervision) in all the experiments. The viewpoints are randomly selected as in [13]. We consider three object categories - chair, car and airplane - and set the variance of the Gaussian kernel in Eq. 1 to $[0.4, 0.4, 0.1]$ and depth well radius r in Eq. 3 to $[1.0, 1.0, 0.75]$ respectively. The weight for feature loss, λ, is set to 1. We use Adam optimizer with a learning rate of $5e^{-5}$ to train the network. Network architecture details are provided in the supplementary.

4.2. Evaluation Methodology

Reconstruction We evaluate 3D reconstruction performance using two distance metrics – Chamfer distance and Earth Mover's Distance (EMD). Chamfer distance between two point clouds P' and P is defined as $d_{Chamfer}(P', P) = \sum_{\alpha \in P'} \min_{\beta \in P} ||\alpha - \beta||_2^2 + \sum_{\alpha \in P} \min_{\beta \in P'} ||\alpha - \beta||_2^2$. A low Chamfer error indicates more faithful reconstructions. EMD between two point sets P' and P is given by: $d_{EMD}(P', P) = \min_{\phi: P' \to P} \sum_{\alpha \in P'} ||\alpha - \phi(\alpha)||_2$ where $\phi : P' \to P$ is a bijection from P' to P. Since it enforces a point-to-point mapping between the two sets, EMD ensures uniformity in point predictions. The ground truth point cloud is obtained by randomly sampling 16,384 points on the surface of the object and performing farthest point sampling to obtain 1024 points. For computing the metrics, we renormalize both the ground truth and predicted point clouds within a bounding box of length 1 unit. We report Chamfer and EMD metrics after scaling them by 100.

Part Segmentation We formulate part segmentation as a per-point classification problem. Evaluation metric is mIoU on points. For each shape S of category c, we calculate the shape mIoU as follows: For each part type in category c, we compute IoU between ground truth and prediction. If the union of ground truth and prediction points is empty, then we count part IoU as 1. We then average IoUs for all part types in category c to get mIoU for that shape. Since there is no correspondence between the ground truth and predicted points, we compute forward and backward mIoUs, before averaging them out to get the final mIoU as follows: $mIoU(P_c, P_c') = \frac{1}{2C} \sum_i \frac{N_{ii}}{\sum_j N_{ij} + \sum_j N_{ji} - N_{ii}} + \frac{1}{2C} \sum_i \frac{N'_{ii}}{\sum_j N'_{ij} + \sum_j N'_{ji} - N'_{ii}}$ where N_{ij} is the number of points in category i in P_c predicted as category j in P_c' for forward point correspondences between P_c and P_c'. Similarly N'_{ij} is for backward point correspondences. C is the total number of categories.

Color Prediction Similar to part-segmentation, we consider the average of forward and backward squared Euclidean distance between the predicted RGB values by obtaining point correspondences.

4.3. Baselines

We compare our approach against state-of-the-art 3D reconstruction and feature prediction networks that utilize full 3D data as supervision. We consider two baselines:

PSGN [3]+PointNet [14]: We train two separate networks for reconstruction and feature prediction. PSGN is trained to predict the ground truth point cloud, and PointNet is trained to predict features given a ground truth point cloud. During inference, the predicted point cloud by PSGN is passed through PointNet to obtain features.

3D-PSRNet [11]: We train a single network to perform both structure and feature prediction. Since there is no correspondence between the ground truth and predicted points, we compute forward and backward feature losses between the two sets [11].

For color prediction, it would not be possible to first obtain reconstructions and then independently regress the colors for them since the color is dependent on the input image. Hence, we compare only against 3D-PSRNet, which jointly regresses shape and color.

4.4. Part Segmentation

Dataset We train all our networks on synthetic models from the ShapePFCN dataset [7] which consists of part segmented ground truth meshes from ShapeNet [1]. For obtaining the 2D ground truth part segmentation, we render these meshes using the mesh label information and threshold the rendered images to obtain the ground truth part segmented images. The corresponding part annotated ground truth point clouds are taken from [23]. We use the same train/test split provided by [7] to train category-specific models in all our experiments.

Results Table 1 presents the results on the ShapePFCN dataset [7] with comparison against the 3D-supervised base-

Category	Metric	PSGN [3] + PointNet [14]	3D-PSRNet [11]	DIFFER
Supervision		3D	3D	2D
Chair	Chamfer ↓	8.15	8.06	9.10
	EMD ↓	11.49	11.80	13.49
	mIoU ↑	75.29	75.98	73.21
Car	Chamfer	5.29	5.26	5.49
	EMD	6.52	5.97	5.59
	mIoU	58.43	61.05	59.67
Airplane	Chamfer	4.21	4.29	4.79
	EMD	6.66	6.23	7.42
	mIoU	65.38	66.89	67.26
Mean	Chamfer	5.88	5.87	6.46
	EMD	8.22	8.00	8.83
	mIoU	66.37	67.97	66.71

Table 1: Reconstruction and Part Segmentation metrics on ShapePFCN dataset [7]. DIFFER performs comparably to the baselines that use full 3D supervision.

lines PSGN [3]+PointNet [14] and 3D-PSRNet [11]. We note that we perform comparably or sometimes even better than the baselines that are trained with full 3D supervision. Fig. 3 shows qualitative results. We observe that we are better able to capture the overall shape and different parts present in the input image (back of chairs, parts of cars, wings of airplanes). Our reconstructions also display uniformity in points.

4.5. Color

Dataset We use the ShapeNet dataset [1] for colorized point cloud reconstruction. Points are sampled on the mesh surfaces and the points are associated with the corresponding face colors. Input image rendering is performed as in the case of part segmentation. The images from various views form the 2D ground truth data.

Results Table 2 provides quantitative comparison with the jointly trained 3D supervision based model. We observe that both the reconstruction and color prediction performances of DIFFER are comparable to that of 3D supervised 3D-PSRNet. Fig. 4 provides qualitative results for colored point cloud reconstruction. We obtain predictions that match the input image features. Clear part distinctions are observed, for e.g., window panes and wheels in cars are clearly identifiable with a different color from that of the body. While 3D-PSRNet, being a 3D supervised network, obtains better reconstructions, DIFFER has higher visual color correspondence to the input image.

Pix3D To show the adaptability of our approach, we also provide results on the real world Pix3D dataset[18]. A random 80%-20% train-test split is utilized. PSGN-joint[3] is initially trained on the ShapeNet dataset and is fine-tuned on the 2D data using DIFFER. To account for the difference in the input image domains of ShapeNet and Pix3D, we train the baseline network by overlaying the synthetic images on

Category	Metric	3D-PSRNet [11]	DIFFER
Chair	Chamfer ↓	7.76	8.44
	EMD ↓	9.47	14.38
	RGB-$\mathcal{L}_2$ ↓	0.12	0.19
Car	Chamfer	4.81	5.05
	EMD	4.35	4.79
	RGB-$\mathcal{L}_2$	0.20	0.26
Airplane	Chamfer	4.18	5.99
	EMD	5.69	8.47
	RGB-$\mathcal{L}_2$	0.15	0.19
Mean	Chamfer	5.58	6.49
	EMD	6.50	9.21
	RGB-$\mathcal{L}_2$	0.16	0.21

Table 2: Reconstruction and Color metrics on ShapeNet [1]. DIFFER performs comparably to the baselines that use full 3D supervision in terms of Chamfer distance.

Approach	Chamfer	Forward	Backward	EMD
PSGN Joint [3]	16.05	7.5	8.55	16.8
DIFFER	**14.29**	**6.98**	**7.31**	**14.46**

Table 3: Reconstruction metrics for color on Pix3D [18]. Note that it is not possible to report RGB metrics due to absence of GT 3D data.

random natural background images [19, 13]. In Table 3, we observe significant boost in the reconstruction performance compared to the baseline. The ability of DIFFER to train using 2D color images facilitates the effective use of such real world datasets.

Category	Chair	Car	Airplane	Mean
CAPNet [13]	9.64	6.71	6.58	7.64
DIFFER	**9.55**	**6.38**	**5.87**	**7.27**

Table 4: Chamfer metrics on ShapeNet [1]. Single view supervision based reconstruction performance. Addition of depth in DIFFER improves reconstruction.

4.6. Role of Depth in Reconstruction

While we predict depth values for each point at each pixel (Eq. 2), the minimum depth across all points for each pixel would yield the depth maps. Such depth maps can be used for supervision to obtain better reconstructions. We consider single-view supervised reconstruction with and without depth supervision. Since minimal information from hidden regions is available to the network, reconstruction quality suffers with mask loss alone (CAPNet [13]). Quantitative metrics in Table 4 and qualitative results (in supplementary material) suggest that additional depth supervision

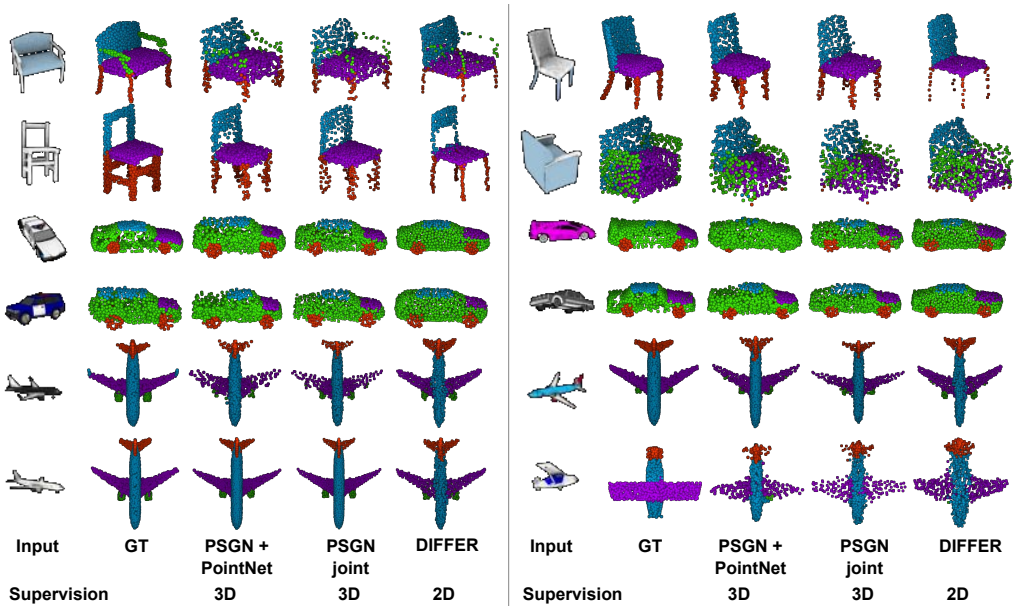

Figure 3: Qualitative results for part-segmented reconstruction on ShapePFCN [7]. While the density of points in DIFFER outputs suffers in quality, there is improved correspondence in shape to the input image. The network is better able to reconstruct and predict parts like chair handles and legs, whereas the baseline tends to predict parts not present in the input.

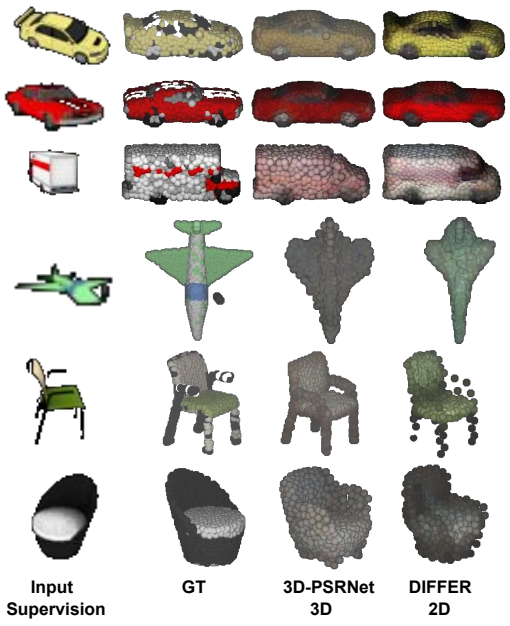

Figure 4: Qualitative results for color prediction on ShapeNet dataset [1]. We are able to accurately reconstruct the 3D shape of the model, while simultaneously capturing the color information in the input image. DIFFER has a higher visual correspondence in color to the input compared to 3D supervised 3D-PSRNet [11].

results in improved reconstruction ability. The depth supervision in DIFFER helps eliminate/reduce problems like presence of spurious points in concave regions and incorrect estimation of depth in thin objects like airplane.

4.7. Influence of Depth Well Radius

We observe the DIFFER projections for different radii r of the depth well in Eq. 3. Lower values of r produce sparse projection maps with 'holes', while higher values result in projections with larger areas (Fig. 5). We set r to an optimal value so as to fill up 'holes' while retaining finer parts.

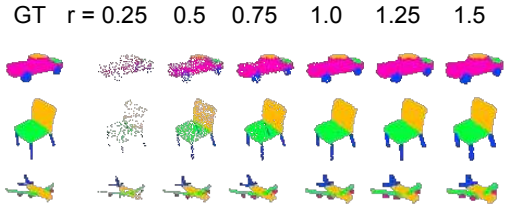

Figure 5: Projected part segmentations for different radii. We observe that lower values of r produce sparse projection maps with 'holes', while those higher values occupy larger areas. We set r to an optimal value so as to fill up holes while retaining finer parts.

5. Conclusion

In this work, we tackle the problem of jointly learning 3D structure and associated features from a single input image using weak multi-view 2D supervision. To this end, we develop a depth-aware differentiable feature renderer (DIFFER), which is used to train our model using only 2D data such as depth, color and part annotations as supervision. Through extensive quantitative and qualitative evaluation on ShapeNet and Pix3D datasets, we show that our approach performs comparably or sometimes even better than existing 3D supervised methods.

References

[1] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 4, 5, 6

[2] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3D-r2n2: A unified approach for single and multi-view 3D object reconstruction. In *European Conference on Computer Vision*, pages 628–644. Springer, 2016. 2

[3] H. Fan, H. Su, and L. Guibas. A point set generation network for 3D object reconstruction from a single image. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 38, 2017. 2, 4, 5

[4] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*, pages 484–499. Springer, 2016. 2

[5] E. Insafutdinov and A. Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *NeurIPS*, 2018. 2

[6] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri. 3d shape segmentation with projective convolutional networks. 2

[7] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri. 3D shape segmentation with projective convolutional networks. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017. 4, 5, 6

[8] H. Kato, Y. Ushiku, and T. Harada. Neural 3d mesh renderer. In *CVPR*, 2018. 2

[9] C.-H. Lin, C. Kong, and S. Lucey. Learning efficient point cloud generation for dense 3D object reconstruction. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 2

[10] P. Mandikal, K. L. Navaneet, M. Agarwal, and R. V. Babu. 3D-LMNet: Latent embedding matching for accurate and diverse 3d point cloud reconstruction from a single image. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018. 2

[11] P. Mandikal, K. L. Navaneet, and R. V. Babu. 3D-PSRNet: Part segmented 3d point cloud reconstruction from a single image. In *3D Reconstruction Meets Semantics Workshop (ECCVW)*, 2018. 2, 4, 5, 6

[12] S. Muralikrishnan, V. G. Kim, and S. Chaudhuri. Tags2parts: Discovering semantic regions from shape tags. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2926–2935, 2018. 2

[13] K. L. Navaneet, P. Mandikal, M. Agarwal, and R. V. Babu. CAPNet: Continuous approximation projection for 3d point cloud reconstruction using 2d supervision. In *AAAI*, 2019. 2, 3, 4, 5

[14] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017. 2, 4, 5

[15] C. R. Qi, L. Yi, H. Su, and L. 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. 2

[16] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 190–198. IEEE, 2017. 2

[17] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2530–2539, 2018. 2

[18] X. Sun, J. Wu, X. Zhang, Z. Zhang, C. Zhang, T. Xue, J. B. Tenenbaum, and W. T. Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *CVPR*, 2018. 1, 5

[19] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, 2017. 2, 5

[20] J. Wu, Y. Wang, T. Xue, X. Sun, B. Freeman, and J. Tenenbaum. Marrnet: 3D shape reconstruction via 2.5 d sketches. In *Advances In Neural Information Processing Systems*, pages 540–550, 2017. 2

[21] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3D shapenets: A deep representation for volumetric shapes. In *CVPR*, pages 1912–1920, 2015. 2

[22] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee. Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In *Advances in Neural Information Processing Systems*, 2016. 2

[23] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas. A scalable active framework for region annotation in 3d shape collections. *SIGGRAPH Asia*, 2016. 4

[24] R. Zhu, H. K. Galoogahi, C. Wang, and S. Lucey. Rethinking reprojection: Closing the loop for pose-aware shape reconstruction from a single image. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 57–65. IEEE, 2017. 2