

# 3D-PSRNet: Part Segmented 3D Point Cloud Reconstruction From a Single Image

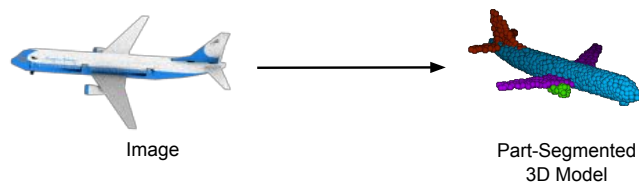
Priyanka Mandikal<sup>\*[0000-0002-6257-7417]</sup>, Navaneet K L<sup>\*[0000-0002-2420-3684]</sup>,  
and R. Venkatesh Babu<sup>[0000-0002-1926-1804]</sup>

Indian Institute of Science, Bangalore, India  
priyanka.mandikal@gmail.com, {navaneetl,venky}@iisc.ac.in

**Abstract.** We propose a mechanism to reconstruct part annotated 3D point clouds of objects given just a single input image. We demonstrate that jointly training for both reconstruction and segmentation leads to improved performance in both the tasks, when compared to training for each task individually. The key idea is to propagate information from each task so as to aid the other during the training procedure. Towards this end, we introduce a *location-aware segmentation loss* in the training regime. We empirically show the effectiveness of the proposed loss in generating more faithful part reconstructions while also improving segmentation accuracy. We thoroughly evaluate the proposed approach on different object categories from the ShapeNet dataset to obtain improved results in reconstruction as well as segmentation. Codes are available at <https://github.com/val-iisc/3d-psrnet>.

**Keywords:** Point cloud, 3D reconstruction, 3D part segmentation

## 1 Introduction



**Fig. 1.** Semantic point cloud reconstruction.

Human object perception is based on semantic reasoning [8]. When viewing the objects around us, we can not only mentally estimate their 3D shape from

---

\* equal contribution

limited information, but we can also reason about object semantics. For instance, upon viewing the image of an airplane in Figure 1, we might deduce that it contains four distinct parts - body, wings, tail, and turbine. Recognition of these parts further enhances our understanding of individual part geometries as well as the overall 3D structure of the airplane. This ability to perceive objects driven by semantics is important for our interaction with the world around us and the manipulation of objects within it.

In machine vision, the ability to infer the 3D structures from single-view images has far-reaching applications in the field of robotics and perception. Semantic understanding of the perceived 3D object is particularly advantageous in tasks such as robot grasping, object manipulation, etc.

Deep neural networks have been successfully employed for tackling the problem of 3D reconstruction. Most of the existing literature propose techniques for predicting the voxelized representation format. However, this representation has a number of drawbacks. First, it suffers from sparsity of information. All the information that is needed to perceive the 3D structure is provided by the surface voxels, while the voxels within the volume increase the representation space with minimal addition of information. Second, the neural network architectures required for processing and predicting 3D voxel maps make use of 3D CNNs, which are computationally heavy and lead to considerable overhead during training and inference. For these reasons, there have been concerted efforts to explore representations that involve reduced computational complexity compared to voxel formats. Very recently, there have been works focusing on designing neural network architectures and loss formulations to process and predict 3D point clouds [13,14,3,16,9]. Since point clouds consist of points being sampled uniformly on the object’s surface, they are able to encode maximal information about the object’s 3D characteristics. The information-rich encoding and compute-friendly architectures makes it an ideal candidate for 3D shape generation and reconstruction tasks. Hence, we consider the point cloud as our representation format.

In this work, we seek to answer three important questions in the tasks of semantic object reconstruction and segmentation: **(1)** What is an effective way of inferring an accurate semantically annotated 3D point cloud representation of an object when provided with its two-dimensional image counterpart? **(2)** How do we incorporate object geometry into the segmentation framework so as to improve segmentation accuracy? **(3)** How do we incorporate semantic understanding into the reconstruction framework so as to improve the reconstruction of individual parts? We achieve the former by training a neural network to jointly optimize for the reconstruction as well as segmentation losses. We empirically show that such joint training achieves superior performance on both reconstruction and segmentation tasks when compared to two different neural networks that are trained on each task independently. To enable the flow of information between the two tasks, we propose a novel loss formulation to integrate the knowledge from both the predicted semantics and the reconstructed geometry.

In summary, our contributions in this work are as follows:

- We propose 3D-PSRNet, a part segmented 3D reconstruction network, which is jointly optimized for the tasks of reconstruction and segmentation.
- To enable the flow of information from one task to another, we introduce a novel loss function called *location-aware segmentation loss*. We empirically show that the proposed loss function aids in the generation of more faithful part reconstructions, while also resulting in more accurate segmentations.
- We evaluate 3D-PSRNet on a synthetic dataset to achieve state-of-the-art performance in the task of semantic 3D object reconstruction from a single image.

## 2 Related Work

### 3D Reconstruction

In recent times, deep learning based approaches have achieved significant progress in the field of 3D reconstruction. The earlier works focused on voxel-based representations [4,19,2]. Girdhar *et al.* [4] map the 3D model and the corresponding 2D representations to a common embedding space to obtain a representation which is both predictable from 2D images and is capable of generating 3D objects. Wu *et al.* [19] utilize variational auto-encoders with an additional adversarial criterion to obtain improved reconstructions. Choy *et al.* [2] employ a 3D recurrent network to obtain reconstructions from multiple input images. While the above works directly utilize the ground truth 3D models in the training stage, [20,17,18,22] try to reconstruct the 3D object using 2D observations from multiple view-points.

Several recent works have made use of point clouds in place of voxels to represent 3D objects [3,5,11]. Fan *et al.* [3] showed that point cloud prediction is not only computationally efficient but also outperforms voxel-based reconstruction approaches. Groueix *et al.* [5] represented a 3D shape as a collection of parametric surface elements and constructed a mesh from the predicted point cloud. Mandikal *et al.* [11] trained an image encoder in the latent space of a point cloud auto-encoder, while also enforcing a constraint to obtain diverse reconstructions. However, all of the above works focus solely on the point cloud reconstruction task.

### 3D Semantic Segmentation

Semantic segmentation using neural networks has been extensively studied in the 2D domain [10,6]. The corresponding task in 3D has been recently explored by works such as [15,13,14,7,12]. Song *et al.* [15] take in a depth map of a scene as input and predict a voxelized occupancy grid containing semantic labels on a per-voxel basis. They optimize for the multi-class segmentation loss and argue that scene completion aids semantic label prediction and vice versa. Our representation format is a 3D point cloud while [15] outputs voxels. This gives rise to a number of differences in the training procedure. Voxel based methods predict an occupancy grid and hence optimize for the cross-entropy loss for both reconstruction as well as segmentation. On the other hand, point cloud based works optimize distance-based metrics for reconstruction and cross-entropy for

segmentation. We introduce a location-aware segmentation loss tailored for point cloud representations.

[13,14] introduce networks that take in point cloud data so as to perform classification and segmentation. They introduce network architectures and loss formulations that are able to handle the inherent un-orderedness of the point cloud data. While [3] predicts only the 3D point cloud geometry from 2D images, and [13,14] segment input point clouds, our approach stresses the importance of jointly optimizing for reconstruction and segmentation while transitioning from 2D to 3D.

### 3 Approach

In this section, we introduce our model, 3D-PSRNet, which generates a part-segmented 3D point cloud from a 2D RGB image. As a baseline for comparison, we train two separate networks for the task of reconstruction and segmentation (Figure 2(a)). Given an RGB image  $I$  as input, the reconstruction network ( $baseline_{rec}$ ) outputs a 3D point cloud  $\hat{X}_p \in \mathbb{R}^{N_p \times 3}$ , where  $N_p$  is the number of points in the point cloud. Given a 3D point cloud  $X_p \in \mathbb{R}^{N_p \times 3}$  as input, the segmentation network ( $baseline_{seg}$ ) predicts the class labels  $\hat{X}_c \in \mathbb{R}^{N_p \times N_c}$ , where  $N_c$  is the number of classes present in the object category. During inference, image  $I$  is passed through  $baseline_{rec}$  to obtain  $\hat{X}_p$ , which is then passed through  $baseline_{seg}$  to obtain  $\hat{X}_c$ .

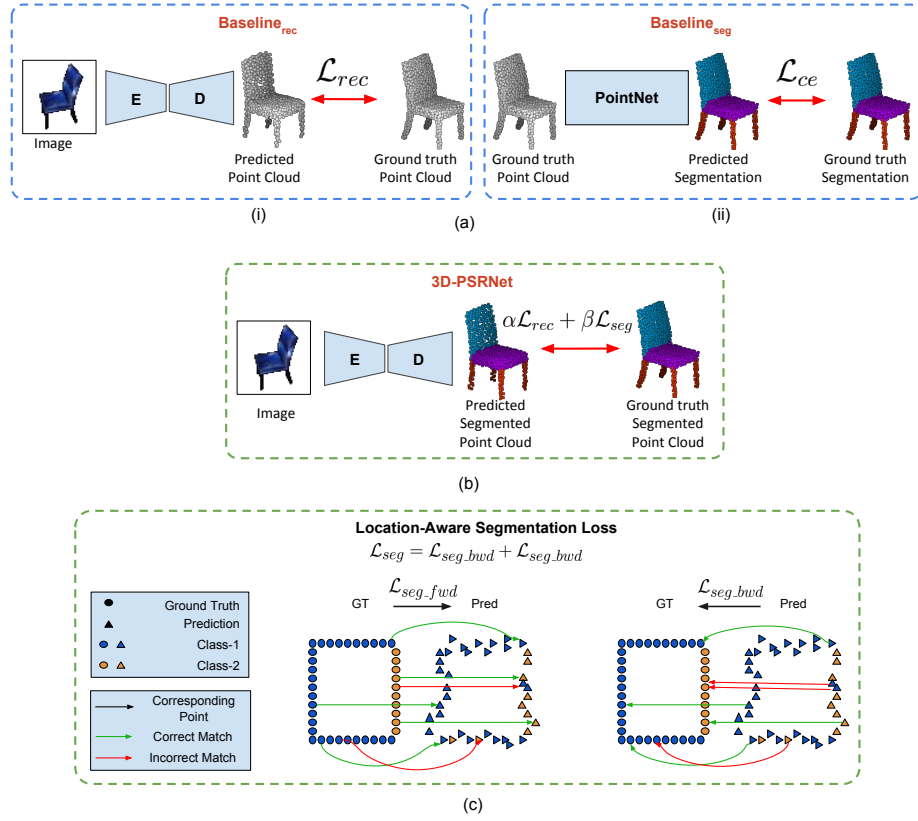
Our training pipeline consists of jointly predicting  $(\hat{X}_p, \hat{X}_c)$  (Figure 2(b)). The reconstruction network is modified such that an additional  $N_c$  predictions, representing the class probabilities of each point, are made at the final layer. The network is simultaneously trained with reconstruction and segmentation losses, as explained below.

#### 3.1 Loss Formulation

**Reconstruction Loss** We require a loss formulation that is invariant to the order of points in the point cloud. To satisfy this criterion, the Chamfer distance between the ground truth point cloud  $X_p$  and predicted point cloud  $\hat{X}_p$  is chosen as the reconstruction loss. The loss function is defined as:

$$\mathcal{L}_{rec} = d_{Chamfer}(X_p, \hat{X}_p) = \sum_{i \in X_p} \min_{j \in \hat{X}_p} \|i - j\|_2^2 + \sum_{i \in \hat{X}_p} \min_{j \in X_p} \|i - j\|_2^2 \quad (1)$$

**Segmentation Loss** We use point-wise softmax cross-entropy loss (denoted by  $\mathcal{L}_{ce}$ ) between the ground truth class labels  $X_c$  and the predicted class labels  $\hat{X}_c$ . For the training of  $baseline_{seg}$ , since there is direct point-to-point correspondence between  $X_p$  and  $\hat{X}_c$ , we directly apply the segmentation loss as the cross-entropy



**Fig. 2.** Semantic point cloud reconstruction approaches. (a) Baseline: (i) A reconstruction network takes in an image input and predicts a 3D point cloud reconstruction of it. (ii) A segmentation network takes in a 3D point cloud as input and predicts semantic labels for every input point. (b) Our approach takes in an image as input and predicts a part segmented 3D point cloud by jointly optimizing for both reconstruction and segmentation, while also additionally propagating information from the semantic labels to improve reconstruction. (c) Point correspondences for location-aware segmentation loss. Incorrect reconstructions and segmentations are both penalized. The overall segmentation loss is the summation of the forward and backward segmentation losses.

loss between  $X_c$  and  $\hat{X}_c$ :

$$\mathcal{L}_{ce}(X_c, \hat{X}_c) = \sum_{\substack{x \in X_c \\ \hat{x} \in \hat{X}_c}} [x \log(\hat{x}) + (1 - x) \log(1 - \hat{x})] \quad (2)$$

However, during joint training, there exists no such point-to-point correspondence between the ground truth and predicted class labels. We therefore intro-

duce the *location-aware segmentation loss* to propagate semantic information between matching point pairs (Figure 2(c)). The loss consists of two terms:

- (1) **Forward segmentation loss** ( $\mathcal{L}_{seg\_fwd}$ ): For every point  $i \in X_p$ , we find the closest point  $i' \in \hat{X}_p$ , and apply  $L_{ce}$  on their corresponding class labels.

$$\mathcal{L}_{seg\_fwd} = \frac{1}{N_p} \sum_{i \in X_p} \mathcal{L}_{ce}(X_{c_i}, \hat{X}_{c_{i'}}) \quad (3)$$

- (2) **Backward segmentation loss** ( $\mathcal{L}_{seg\_bwd}$ ): For every point  $i \in \hat{X}_p$ , we find the closest point  $i' \in X_p$ , and apply  $L_{ce}$  on their corresponding class labels.

$$\mathcal{L}_{seg\_bwd} = \frac{1}{N_p} \sum_{i \in \hat{X}_p} \mathcal{L}_{ce}(X_{c_{i'}}, \hat{X}_{c_i}) \quad (4)$$

The overall segmentation loss is then the summation of the forward and backward segmentation losses:

$$\mathcal{L}_{seg} = \mathcal{L}_{seg\_fwd} + \mathcal{L}_{seg\_bwd} \quad (5)$$

The total loss during joint training is then given by,

$$\mathcal{L}_{tot} = \alpha \mathcal{L}_{rec} + \beta \mathcal{L}_{seg} \quad (6)$$

### 3.2 Implementation Details

For training the baseline segmentation network  $baseline_{seg}$ , we follow the architecture of the segmentation network of PointNet [13], which consists of ten 1D convolutional layers of filter sizes [64, 64, 64, 128, 1024, 512, 256, 128, 128,  $N_c$ ], where  $N_c$  is the number of class labels. A global maxpool function is applied after the fifth layer and the resulting feature is concatenated with each individual point feature, as is done in the original paper. Learning rate is set to  $5e^{-4}$  and batch normalization is applied at all the layers of the network. The networks for the baseline reconstruction network and the joint 3D-PSRNet are similar in architecture. They consist of four 2D convolutional layers with number of filters as [32, 64, 128, 256], followed by four fully connected layers with output dimensions of size [128, 128, 128,  $N_p \times 3$ ] (reconstruction) and [128, 128, 128,  $N_p \times (3 + N_c)$ ] (joint), where  $N_p$  is the number of points in the point cloud. We set  $N_p$  to be 1024 in all our experiments. Learning rate for  $baseline_{rec}$  and 3D-PSRNet are set to  $5e^{-5}$  and  $5e^{-4}$  respectively. We use a minibatch size of 32 in all the experiments. We train the individual reconstruction and segmentation networks for 1000 epochs, while the joint network (3D-PSRNet) is trained for 500 epochs. We choose the best model according to the corresponding minimum loss. In Eq. 6, the values of  $\alpha$  and  $\beta$  are set to  $1e^4$  and 1 respectively for joint training.

## 4 Experiments

### 4.1 Dataset

We train all our networks on synthetic models from the ShapeNet dataset [1] whose part annotated ground truth point clouds are provided by [21]. Our dataset comprises of 7346 models from three exemplar categories - chair, car and airplane. We render each model from ten different viewing angles with azimuth values in the range of  $[0^\circ, 360^\circ]$  and elevation values in the range of  $[-20^\circ, 40^\circ]$  so as to obtain a dataset of size 73,460. We use the train/validation/test split provided by [21] and train a single model on all the categories in all our experiments.

### 4.2 Evaluation Methodology

- (1) **Reconstruction:** We report both the Chamfer Distance (Eqn. 1) as well as the Earth Mover’s Distance (or EMD) computed on 1024 points in all our evaluations. EMD between two point sets  $\hat{X}_p$  and  $X_p$  is given by:

$$d_{EMD}(X_p, \hat{X}_p) = \min_{\phi: X_p \rightarrow \hat{X}_p} \sum_{x \in X_p} \|x - \phi(x)\|_2 \quad (7)$$

where  $\phi: X_p \rightarrow \hat{X}_p$  is a bijection. For computing the metrics, we renormalize both the ground truth and predicted point clouds within a bounding box of length 1 unit.

- (2) **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, compute IoU between groundtruth and prediction. If the union of groundtruth and prediction points is empty, then count part IoU as 1. Then we average IoUs for all part types in category C to get mIoU for that shape. To calculate mIoU for the category, we take average of mIoUs for all shapes in that category. Since there is no correspondence between the ground truth and predicted points, we use a mechanism similar to the one described in Section 3.1 for computing the forward and backward mIoUs, before averaging them out to get the final mIoU as follows:

$$mIoU(X_c, \hat{X}_c) = \frac{1}{2N_c} \sum_i \frac{N_{ii}}{\sum_j N_{ij} + \sum_j N_{ji} - N_{ii}} + \frac{1}{2N_c} \sum_i \frac{\hat{N}_{ii}}{\sum_j \hat{N}_{ij} + \sum_j \hat{N}_{ji} - \hat{N}_{ii}} \quad (8)$$

where  $N_{ij}$  is the number of points in category  $i$  in  $X_c$  predicted as category  $j$  in  $\hat{X}_c$  for forward point correspondences between  $X_c$  and  $\hat{X}_c$ . Similarly  $\hat{N}_{ij}$  is for backward point correspondences.  $N_c$  is the total number of categories.

Category	Metric	PSGN-FC [3] + PointNet [13]	3D-PSRNet
Chair	Chamfer	6.82	<b>6.57</b>
	EMD	11.37	<b>10.10</b>
	mIoU	78.09	<b>81.92</b>
Car	Chamfer	5.48	<b>5.14</b>
	EMD	5.88	<b>5.53</b>
	mIoU	59.0	<b>61.57</b>
Airplane	Chamfer	<b>4.06</b>	<b>4.06</b>
	EMD	7.06	<b>6.24</b>
	mIoU	62.86	<b>68.64</b>
<b>Mean</b>	Chamfer	5.45	<b>5.26</b>
	EMD	8.10	<b>7.29</b>
	mIoU	66.65	<b>70.71</b>

**Table 1.** Reconstruction and Segmentation metrics on ShapeNet [1]. 3D-PSRNet significantly outperforms the baseline in both the reconstruction and segmentation metrics on all categories. Chamfer and EMD metrics are scaled by 100.

### 4.3 Results

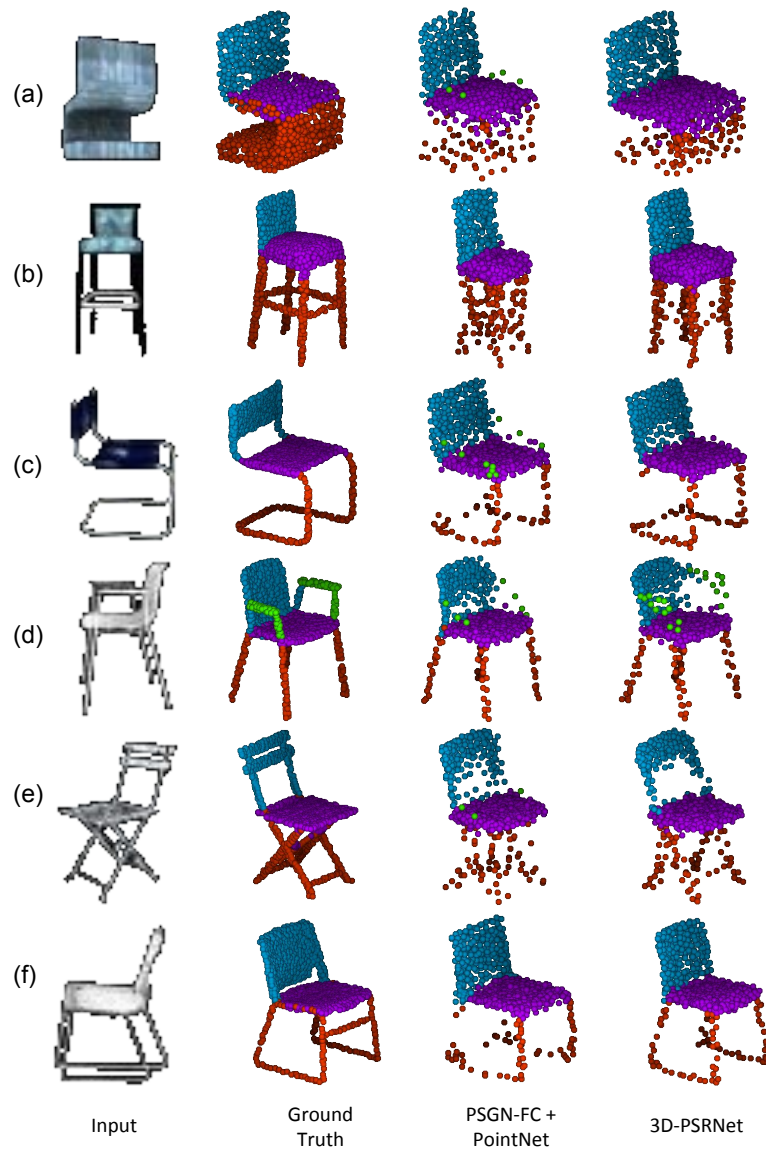
Table 1 presents the quantitative results on ShapeNet for the baseline and joint training approaches. 3D-PSRNet achieves considerable improvement in both the reconstruction (Chamfer, EMD) and segmentation (mIoU) metrics. It outperforms the baseline approach in every metric on all categories. On an average, we obtain **4.1%** improvement in mIoU.

The qualitative results are presented in Figures 3 and 4. 3D-PSRNet obtains more faithful reconstructions compared to the baseline to achieve better correspondence with the input image. It also predicts more uniformly distributed point clouds. We observe that joint training results in reduced hallucination of parts (for e.g. predicting handles for chairs without handles) and spurious segmentations. We also show a few failure cases of our approach in Figure 5. The network misses out on some finer structures present in the object (e.g. dual turbines in the case of airplanes). The reconstructions are poorer for uncommon input samples. However, these drawbacks also exist in the baseline approach.

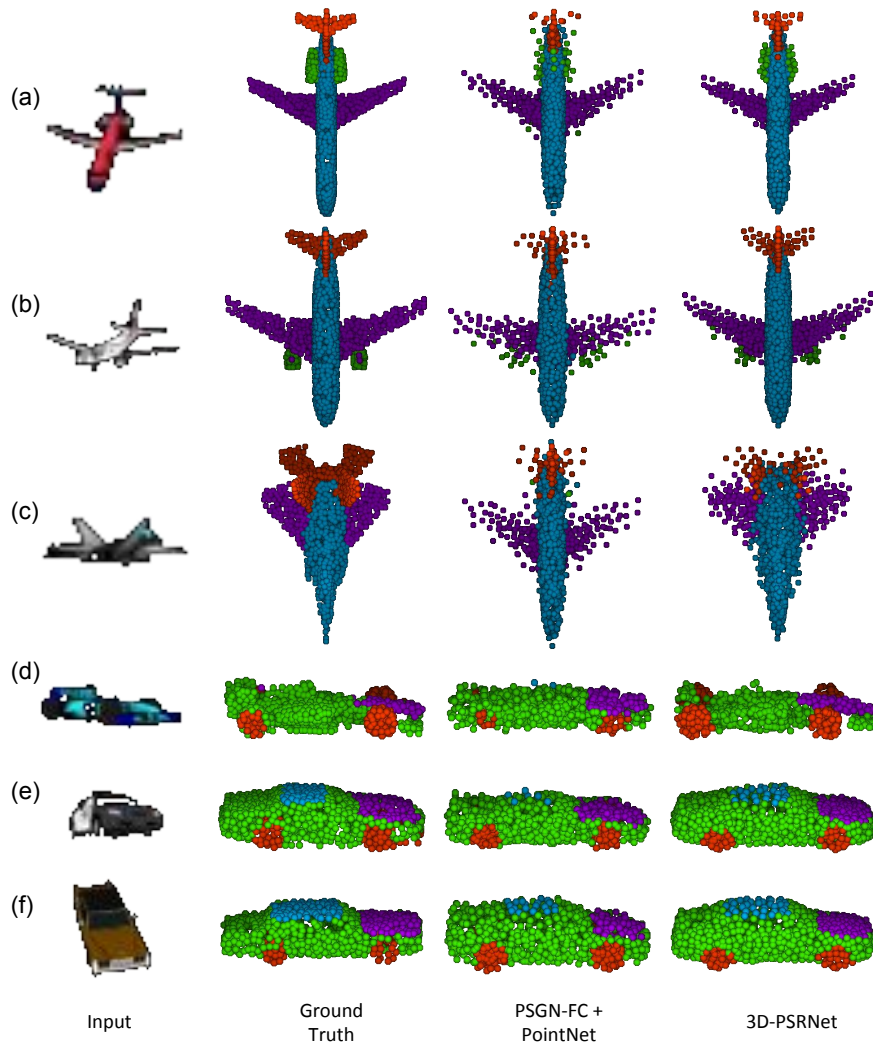
### 4.4 Relative Importance of Reconstruction and Segmentation losses

We present an ablative study on the relative weightage of the reconstruction and segmentation losses in Eq. 6. We fix the value of  $\beta$  to one, while  $\alpha$  is varied from  $10^2$  to  $10^5$ . Figure 6 presents the plot of Chamfer, EMD and mIoU metrics for varying values of  $\alpha$ . We observe that for very low value of  $\alpha$ , both the reconstruction and segmentation metrics are worse off, while there is minimal effect on the average metrics for  $\alpha$  greater than  $10^3$ . Based on Figure 6, we set the value of  $\alpha$  to  $10^4$  in all our experiments.

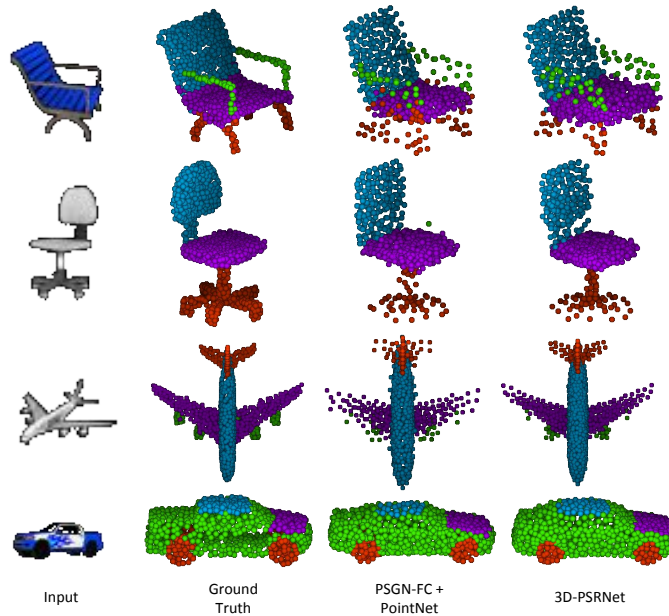




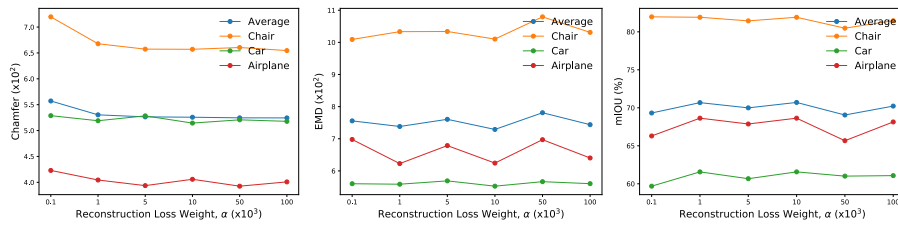
**Fig. 3.** Qualitative results on the chair category from ShapeNet [1]. Compared to the baseline (PSGN [3] + PointNet [13]), we are better able to capture the details present in the input image. Individual parts such as legs (b,e,f) and handles (d) are reconstructed with greater accuracy. Additionally, while outlier points are present in the baseline (a,c), our method produces more uniformly distributed reconstructions.



**Fig. 4.** Qualitative results on airplanes and cars from ShapeNet [1]. Compared to the baseline (PSGN [3] + PointNet [13]), we are better able to reconstruct individual parts in each category resulting in better overall shape. Our method produces sharper reconstruction of tails and wings in airplanes (a,b). We also obtain more uniformly distributed points (as is visible in the wing region of airplanes). In cars, our reconstructions better correspond to the input image compared to the baseline.



**Fig. 5.** Failure cases of our method. We notice that our method fails to get finer details in some instances, such as leg details in chairs, dual turbines present in airplanes, and certain car types.



**Fig. 6.** Ablative study on weight for reconstruction loss,  $\alpha$ . Chamfer, EMD and mIOU metrics are calculated for different values of  $\alpha$ . Based on the plots, we choose the value of  $\alpha$  to be  $10^4$ .

## 5 Conclusion

In this paper, we highlighted the importance of jointly learning the tasks of 3D reconstruction and object part segmentation. We introduced a loss formulation in the training regime to enable propagating information between the two tasks so as to generate more faithful part reconstructions while also improving segmentation accuracy. We thoroughly evaluated against existing reconstruction and segmentation baselines, to demonstrate the superiority of the proposed ap-

proach. Quantitative and qualitative evaluation on the ShapeNet dataset demonstrate the effectiveness in generating more accurate point clouds with detailed part information in comparison to the current state-of-the-art reconstruction and segmentation networks.

## References

1. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3D model repository. arXiv preprint arXiv:1512.03012 (2015) [7](#), [8](#), [9](#), [10](#)
2. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3D-r2n2: A unified approach for single and multi-view 3D object reconstruction. In: European Conference on Computer Vision. pp. 628–644. Springer (2016) [3](#)
3. Fan, H., Su, H., Guibas, L.: A point set generation network for 3D object reconstruction from a single image. In: Conference on Computer Vision and Pattern Recognition (CVPR). vol. 38 (2017) [2](#), [3](#), [4](#), [8](#), [9](#), [10](#)
4. Girdhar, R., Fouhey, D.F., Rodriguez, M., Gupta, A.: Learning a predictable and generative vector representation for objects. In: European Conference on Computer Vision. pp. 484–499. Springer (2016) [3](#)
5. Groueix, T., Fisher, M., Kim, V.G., Russell, B., Aubry, M.: AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2018) [3](#)
6. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Computer Vision (ICCV), 2017 IEEE International Conference on. pp. 2980–2988. IEEE (2017) [3](#)
7. Kalogerakis, E., Averkiou, M., Maji, S., Chaudhuri, S.: 3d shape segmentation with projective convolutional networks. In: CVPR (2017) [3](#)
8. Koopman, S.E., Mahon, B.Z., Cantlon, J.F.: Evolutionary constraints on human object perception. *Cognitive science* **41**(8), 2126–2148 (2017) [1](#)
9. Li, Y., Bu, R., Sun, M., Chen, B.: PointCNN. arXiv preprint arXiv:1801.07791 (2018) [2](#)
10. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015) [3](#)
11. Mandikal, P., K L, N., Agarwal, M., Babu, R.V.: 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) [3](#)
12. Muralikrishnan, S., Kim, V.G., Chaudhuri, S.: Tags2parts: Discovering semantic regions from shape tags. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2926–2935 (2018) [3](#)
13. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3D classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE* **1**(2), 4 (2017) [2](#), [3](#), [4](#), [6](#), [8](#), [9](#), [10](#)
14. Qi, C.R., Yi, L., Su, H., 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 (2017) [2](#), [3](#), [4](#)
15. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic scene completion from a single depth image. In: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. pp. 190–198. IEEE (2017) [3](#)

16. Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, V., Yang, M.H., Kautz, J.: Splatnet: Sparse lattice networks for point cloud processing. arXiv preprint arXiv:1802.08275 (2018) [2](#)
17. Tulsiani, S., Zhou, T., Efros, A.A., Malik, J.: Multi-view supervision for single-view reconstruction via differentiable ray consistency. In: CVPR. vol. 1, p. 3 (2017) [3](#)
18. Wu, J., Wang, Y., Xue, T., Sun, X., Freeman, B., Tenenbaum, J.: Marrnet: 3D shape reconstruction via 2.5 d sketches. In: Advances In Neural Information Processing Systems. pp. 540–550 (2017) [3](#)
19. Wu, J., Zhang, C., Xue, T., Freeman, B., Tenenbaum, J.: Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In: Advances in Neural Information Processing Systems. pp. 82–90 (2016) [3](#)
20. Yan, X., Yang, J., Yumer, E., Guo, Y., 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) [3](#)
21. Yi, L., Kim, V.G., Ceylan, D., Shen, I.C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., Guibas, L.: A scalable active framework for region annotation in 3d shape collections. SIGGRAPH Asia (2016) [7](#)
22. Zhu, R., Galoogahi, H.K., Wang, C., Lucey, S.: Rethinking reprojection: Closing the loop for pose-aware shape reconstruction from a single image. In: Computer Vision (ICCV), 2017 IEEE International Conference on. pp. 57–65. IEEE (2017) [3](#)