Instance Neural Radiance Field Supplementary Material

Yichen Liu[1*] Benran Hu[2*] Junkai Huang[2*] Yu-Wing Tai[3] Chi-Keung Tang[1]
[1]The Hong Kong University of Science and Technology
[2]Carnegie Mellon University [3]Dartmouth College

1. NeRF 3D Instance Segmentation Dataset

Leveraging 3D-FRONT [1] and the data generating approach of [4], we produce a new benchmark for instance-level 3D scene understanding curated for NeRF. 3D-FRONT is a large-scale synthetic indoor scene dataset, from which NeRF-RPN renders RGB images and layout configuration and tailors it as a benchmark for object detection task in NeRF. As shown in Table 1, apart from multi-view images with camera poses and ground truth 3D bounding boxes, 2D ground truth instance segmentation and 3D ground truth instance masks on grids with class labels are included in our new dataset, which can be used for 3D segmentation in NeRF and other research areas.

Dataset	NeRF-RPN	Ours
# scenes	152	1015
RGB images	✓	✓
Camera poses	✓	✓
3D bounding boxes	✓	✓
2D inst seg GT	-	✓
3D voxelized inst seg GT	-	✓

Table 1: A comparison between the 3D-FRONT NeRF dataset in NeRF-RPN and ours.

2. NeRF-RCNN Architecture

In this section, we describe the architecture of NeRF-RCNN in detail. NeRF-RCNN is a proposal-based 3D mask prediction model that imitates the architecture of Mask-RCNN [3]. The input of NeRF-RCNN are the 3D radiance and density grid sampled from a pre-trained NeRF, and the Region of Interests (RoI) provided by NeRF-RPN [4]. For each RoI, we set the ground truth box with the largest intersection over union (IoU) as its regression target.

The first part of NeRF-RCNN is a backbone identical to [4] for feature extraction. The second part takes the feature of each RoI as input and predicts the 3D bounding box, classification probability and discrete 3D mask. To obtain the feature of a single proposal on a feature map, we extend RoIAlign [3] with one more dimension, making all RoI features consistent. Aligned features are fed into two heads, namely *box head* and *mask head*. *Box head* first flattens the inputs for fully connected layer encoding and then separates into box branch and classification branch. The box branch further regresses a RoI to a more accurate bounding box for each class, while the classification branch predicts the classification scores. We follow similar network architecture in [3] by changing the 2D convolution and strided convolution layers to their corresponding 3D version. The loss function of *box head* consists of two parts:

$$\mathcal{L}_{cls} = \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} \mathcal{L}_{BCE}(\boldsymbol{p}_i, \boldsymbol{p}_i^*), \tag{1}$$

$$\mathcal{L}_{reg} = \frac{1}{|\mathcal{N}_p|} \sum_{i \in \mathcal{N}_p} \sum_{k=1}^{L} p_{i,k}^* \mathcal{L}_{smooth}(\boldsymbol{t}_{i,k}, \boldsymbol{t}_i^*), \tag{2}$$

where $\boldsymbol{p}_i$ is the predicted classification score vector after sigmoid, $p_{i,k}$ is the $k-$th dimension of $\boldsymbol{p}_i$, $\boldsymbol{t}_{i,k}$ is the box offsets of class k , $\boldsymbol{p}_i^*, \boldsymbol{t}_i^*$ are ground-truth targets, $\mathcal{N}$ is the set of sampled RoIs, $\mathcal{N}_p$ is the set of positive samples, and L is the number of classes including background. $\mathcal{L}_{BCE}$ and $\mathcal{L}_{smooth}$ denote the binary cross entropy(BCE) loss and the smooth L1 loss in [2] respectively. Note that for an RoI associated with ground truth class c, only the $c-$th box regression BCE loss contributes to the total loss. $\boldsymbol{t}_{i,k} = (t_{x,k}, t_{y,k}, t_{z,k}, t_{w,k}, t_{l,k}, t_{h,k})$ is the box head output. The relationship between $\boldsymbol{t}_{i,k}$ and bounding box parameters x, y, z, w, h, l is defined similarly to [4]:

$$\begin{aligned} t_{x,k} &= (x_k - x_a)/w_a, \quad & t_{y,k} &= (y_k - y_a)/l_a, \\ t_{z,k} &= (z_k - z_a)/h_a, \quad & t_{z,k} &= \log(w_k/w_a), \\ t_{l,k} &= \log(l_k/l_a), & t_{h,k} &= \log(h_k/h_a), \end{aligned} \tag{3}$$

where x_k, y_k, z_k are the center coordinate, w_k, l_k, h_k are the lengths of sides, and $x_a, y_a, z_a, w_a, l_a, h_a$ are the corresponding parameter of the RoI.

*Equal contribution.

†This research is supported in part by the Research Grant Council of the Hong Kong SAR under grant no. 16201420.

The mask head is a convolutional neural network which predicts L binary masks with size $m \times m \times m$ for each RoI. $m = 5$ is used for the box head, and $m = 10$ for the mask head. We also apply the sigmoid function as the activation. The loss for the mask head $\mathcal{L}_M$ is defined as

$$\mathcal{L}_M = \frac{\lambda}{|\mathcal{N}_p|} \sum_{i \in \mathcal{N}_p} \sum_{k=1}^{L} p^*_{i,k} \mathcal{L}_p(\boldsymbol{m}_{i,k}, \boldsymbol{m}^*_i), \tag{4}$$

where m^*_i is the ground truth mask and $m_{i,k}$ is the predicted mask of class k. Similar to box regression, only the mask BCE loss corresponding to the ground truth label is included in $\mathcal{L}_M$.

The total loss of Instance-NeRF $\mathcal{L}$ is

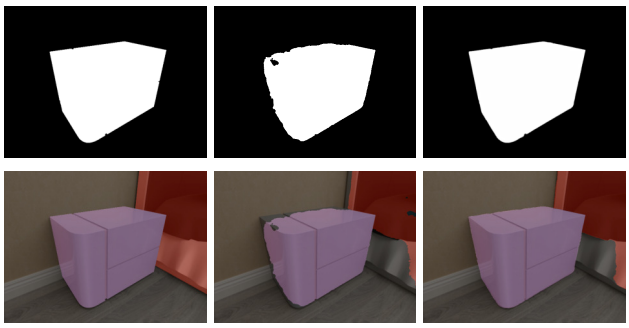
$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_1 \mathcal{L}_{reg} + \lambda_2 \mathcal{L}_{mask}, \tag{5}$$

where λ_1, λ_2 are hyper-parameters.

3. Qualitative Results of Ablation

We present additional visualization and qualitative comparisons to demonstrate the effectiveness of our proposed mask refinement stage.

Although adding instance label regularization can help smooth the instance field, the segmentation quality of the preliminary results can still be unsatisfactory, especially on the silhouette of the objects. Besides, regularization can sometimes smooth out detailed structures in the segmentation, like thin chair legs or lamp stands. As illustrated in Figure 1, performing 2D mask refinement using CascadePSP on the Instance-NeRF results and using it to guide further training can significantly improve the segmentation quality on the object boundaries.



GT w/o refinement w/ refinement

Figure 1: **Ablation on 2D mask refinement.** The first row shows the separate mask for the nightstand, which is used as the input to CascadePSP. The separate masks after refinement are then composed into a single segmentation map to further optimize the instance field, the results of which are presented in the bottom row.

4. Additional Qualitative Comparison

We demonstrate extra qualitative comparisons between our method and other related methods as mentioned in the main paper. The results are given in Figure 2. Please watch the video at `https://www.youtube.com/watch?v=wW9Bme73coI` for more qualitative comparison results.

References

[1] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *IJCV*, pages 1–25, 2021.

[2] Ross Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015.

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[4] Benran Hu, Junkai Huang, Yichen Liu, Yu-Wing Tai, and Chi-Keung Tang. Nerf-rpn: A general framework for object detection in nerfs. In *CVPR*, 2023.

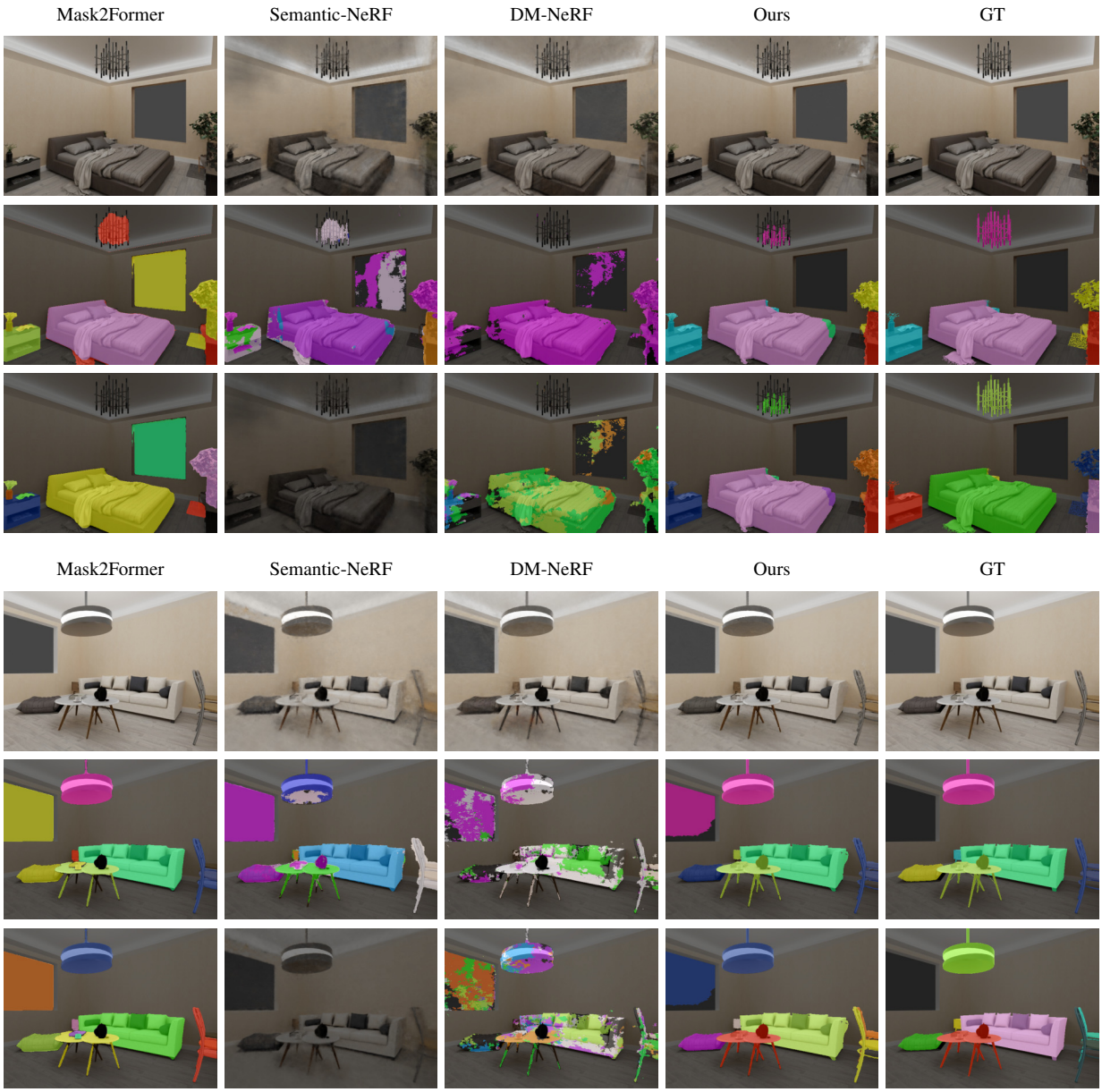


Figure 2: **Additional Comparison.** This figure illustrates the comparison between ours and other methods. For each group of comparison, rows from top to bottom are i. ground truth RGB images or the rendered RGB images from the NeRF models, ii. semantic segmentation, and iii. instance segmentation. The instance segmentation results from Semantic-NeRF are left empty as it does not produce instance-level information.

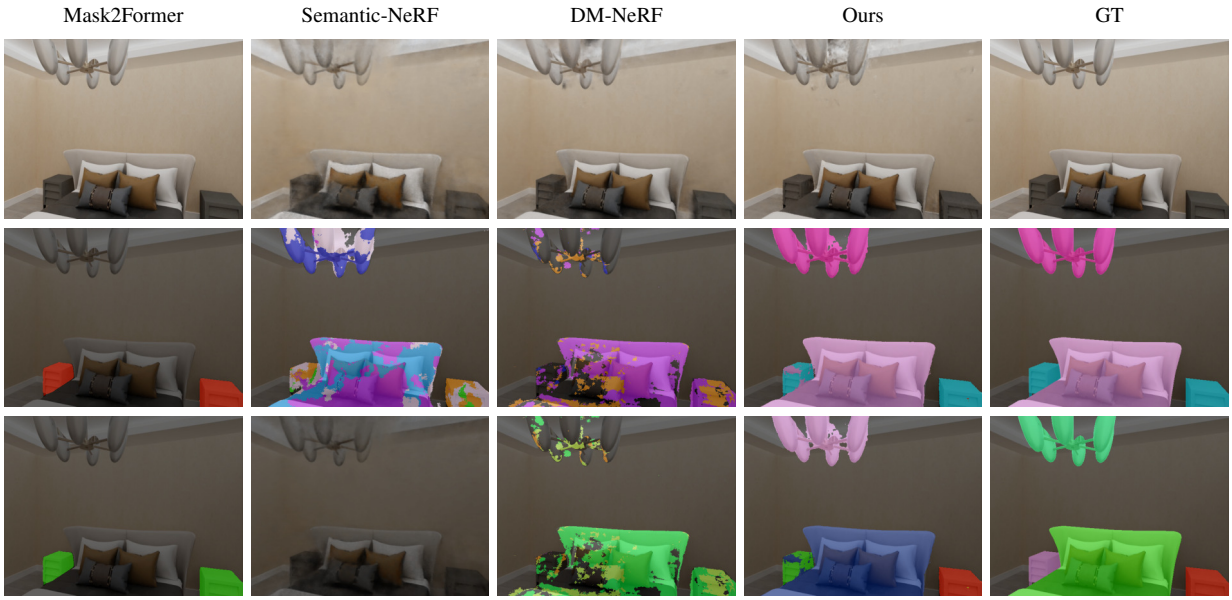


Figure 2: **Additional Comparison (cont.)** This figure illustrates the comparison between ours and other methods. For each group of comparison, rows from top to bottom are i. ground truth RGB images or the rendered RGB images from the NeRF models, ii. semantic segmentation, and iii. instance segmentation. The instance segmentation results from Semantic-NeRF are left empty as it does not produce instance-level information.