A. Model Architecture

The detailed architecture of our framework is shown in Fig. 1. We follow the MLPs design in NeRF [6], and a panoptic branch is added in the middle of the MLPs for generating panoptic feature \( f_p \) and predicting semantic logits \( s \). \( PE(\cdot) \) is the positional encoding function and \(+\) denotes the concatenate operation.

Figure 1. PCFF network architecture. The MLPs accept the 3D position \( x \) and direction \( d \) as input, and output the view-dependent color \( c \), the view-invariant density \( \sigma \), semantic logits \( s \) and panoptic feature \( f_p \).

B. Dataset Details

ScanNet. In our experiment, we choose 3 scenes including '0038_00', '0113_00', and '0192_00'. The resolution of images and panoptic annotations are resized to 640 \( \times \) 480. The train/test images are evenly sampled in each chosen scene, and our data split will be released with the code.

Replica. In our experiment, we choose 6 one-room scenes including 'room_0', 'room_1', 'room_2', 'office_2', 'office_3', and 'office_4'. The resolution of images and panoptic annotations are resized to 640 \( \times \) 480. We adopt the train/testing data split as SemanticNeRF [10] proposed.

ToyDesk. ToyDesk dataset contains two scenes ‘desk_1’ and ‘desk_2’ with 96 and 151 posed images and corresponding instance annotations, which resolutions are 640 \( \times \) 353 and 640 \( \times \) 480 respectively. We adopt the train/testing data split as ObjectNeRF [8] proposed. Since there are no semantic ground truth annotations in this dataset, we directly divide the scene into two semantic classes including foreground and background, where the foreground segmentation is the union of all instances.

C. Implementation Details

C.1. Compared Methods

SemanticNeRF [10] is an extension of NeRF that jointly encode geometry and semantics for semantic labeling. They use a batch size of 1024 rays. Although SemanticNeRF is not specially designed for object-compositional representation, instance-level scene decomposition can be achieved when using ground truth instance annotations to supervise this method.

ObjectNeRF [8] uses a two-branch framework to build object-compositional representation, where one branch is used for individual object modeling while the other is for scene representation. They use a batch size of 2048 rays. We note that their method utilizes the ground truth 3D point clouds of the target scene for additional depth supervision which is not used in other compared methods, thus we remove the depth loss in their model training for a fair comparison.

ObjectSDF [7] is a VolSDF [9]-based framework and achieves remarkable object extraction and reconstruction results by building an explicit connection between the instance predictions and object SDFs. They use a batch size of 1024 rays. We note that their full training needs 10000 epochs which costs almost 7 days, thus we properly shorten the required training epochs to 1500 for a fair comparison in time with other methods.
C.2. 2D Panoptic Segmentation Networks

We adopt three 2D panoptic segmentation networks including PanopticFPN [4], MaskFormer [3], and Mask2Former [2] for predicting network-inferred labels. All networks are employed by MMDetection [1] and pre-trained on COCO [5] dataset. These networks provide various pre-trained versions with different backbones (e.g. ResNet50). Due to our aim to generate accurate labels on real-world scenes, we select the best version of each network. Concretely, PanopticFPN uses ResNet-101, MaskFormer uses Swin-L and Mask2Former uses Swin-L.

C.3. Segmentation Accuracy Comparison

We conduct a segmentation accuracy comparison with SemanticNeRF [10] and ObjectSDF [7] in the Sec. 4.3 of the paper to demonstrate that proposed PCFF can address the 3D index inconsistency in network-inferred labels while others cannot. We give the implementation details here. Due to our method does not explicitly predict the instance labels for each 3D point, we alternatively use the feature similarity maps to generate approximate instance masks for segmentation evaluation. Concretely, we select a centered object by the query pixel in each ScanNet scene. The feature similarity map of the target view is generated by calculating the projected panoptic feature similarity between the query pixel and each 2D pixel in the target view. Therefore, the instance mask is generated by a threshold of 0.95, i.e., for pixel $a$, its instance value is set to 1 if the corresponding feature similarity is bigger than 0.95, and is set to 0 otherwise. We show the selected objects, feature similarity maps, approximate instance masks, and the instance masks of SemanticNeRF and ObjectSDF in sequence in Fig. 4. We notice that the visualizations are examples and the quantitative results are calculated on all test views.

D. More Experimental Results

D.1. Correlation Between Attributes

We claim that semantic $s$ is an easier attribute to learn than appearance $c$ in the Sec. 3.2 of the paper, thus we conduct a simple experiment to verify. As shown in Fig. 2, the prediction of $s$ is relatively correct even if the training is just started (20k iterations), while the rendering quality is low.

D.2. Using Different Network-Inferred Labels

We conduct the quantitative comparison to show the rendering performance on ScanNet of our methods when using network-inferred labels predicted by different 2D panoptic segmentation networks including PanopticFPN [4], MaskFormer [3], and Mask2Former [2]. PQ is the panoptic quality metric to measure the accuracy of predicted labels. We observe that the rendering quality is relative to the PQ, which verifies that our method can be benefited from the development of panoptic segmentation networks.

D.3. Parameter Analysis

We study the balance hyper-parameters in the total loss by setting them to different values and observing the rendering performance on Replica ‘office_4’ scene in Fig. 3. We use the PSNR($\uparrow$) and LPIPS($\downarrow$) to measure the rendering performance.

<table>
<thead>
<tr>
<th></th>
<th>ScanNet</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PS Methods</td>
<td>PQ ↑</td>
<td>PSNR ↑</td>
</tr>
<tr>
<td>Ground Truth</td>
<td>-</td>
<td>26.45</td>
</tr>
<tr>
<td>Panoptic-FPN [4]</td>
<td>44.1</td>
<td>26.26</td>
</tr>
<tr>
<td>MaskFormer [3]</td>
<td>53.2</td>
<td>26.32</td>
</tr>
<tr>
<td>Mask2Former [2]</td>
<td>57.6</td>
<td>26.34</td>
</tr>
</tbody>
</table>

Table 1. Comparison of rendering performance with labels predicted by different panoptic segmentation networks. PQ is the panoptic quality metric to measure the accuracy of labels.

Analysis of the $\lambda_{sem}$. The hyper-parameter $\lambda_{sem}$ is used to balance the weight of semantic loss $L_{sem}$. Experiments show that the best performance is achieved when $\lambda_{sem} = 1 \times 10^{-3}$. Continuing to decrease $\lambda_{sem}$, the rendering performance is degraded because the effectiveness of semantic-related strategies especially the semantic-guided regional refinement is weakened simultaneously.

Analysis of the $\lambda_{ins}$. The hyper-parameter $\lambda_{ins}$ is used to balance the weight of instance quadruplet loss $L_{ins}$, and the best performance is achieved when $\lambda_{ins} = 5 \times 10^{-4}$. Due to the instance quadruplet loss is additionally employed on the feature space, the rendering performance is degraded if a larger $\lambda_{ins}$ is used. On the contrary, a smaller $\lambda_{ins}$ will suppress the decomposition effectiveness.
D.4. Scene Rendering Comparison

We show the qualitative comparison results to show our rendering capacity in Fig 5. The results show ObjectNeRF [8] is prone to render noisy results if the depth supervision is removed, and the rendering results of ObjectSDF [7] are too smooth and lose texture details, especially on high-fidelity scenes such as Replica due to their method is developed based on SDF. Thanks to our proposed semantic-related strategies, our method achieves remarkable rendering results on multiple scene datasets.

D.5. Scene Editing Result on LLFF

We further show the editing result on the ‘room’ scene in the LLFF dataset. We notice that LLFF does not provide the ground truth instance annotations. The network-inferred labels are predicted by Mask2Former [2]. Our method successfully edits the target chair without influencing adjacent chairs, which demonstrates that our method can produce multi-view consistent scene editing results at instance level with network-inferred labels.

References


Figure 4. We show the selected objects, feature similarity maps, approximate instance masks and the instance masks of SemanticNeRF and ObjectSDF in sequence for demonstration and comparison. The red dots in input images are query pixels.

Figure 5. Qualitative comparison of rendering capacity on multiple scene datasets.
Figure 6. Query-based edits of the target chair on ‘room’ scene in LLFF dataset. The red dot in View #1 is the query pixel.