





# SLOTLIFTER: Slot-guided Feature Lifting for Learning Object-centric Radiance Fields

Yu Liu<sup>\*†1,2</sup>, Baoxiong Jia<sup>\*1</sup>, Yixin Chen<sup>1</sup>, and Siyuan Huang<sup>1</sup>

<sup>1</sup> State Key Laboratory of General Artificial Intelligence, BIGAI

<sup>2</sup> Department of Automation, Tsinghua University

\*Equal contribution <sup>†</sup>Work done as an intern at BIGAI

<https://slotlifter.github.io>

**Abstract.** The ability to distill object-centric abstractions from intricate visual scenes underpins human-level generalization. Despite the significant progress in object-centric learning methods, learning object-centric representations in the 3D physical world remains a crucial challenge. In this work, we propose SLOTLIFTER, a novel object-centric radiance model addressing scene reconstruction and decomposition jointly via slot-guided feature lifting. Such a design unites object-centric learning representations and image-based rendering methods, offering state-of-the-art performance in scene decomposition and novel-view synthesis on four challenging synthetic and four complex real-world datasets, outperforming existing 3D object-centric learning methods by a large margin. Through extensive ablative studies, we showcase the efficacy of designs in SLOTLIFTER, revealing key insights for potential future directions.

**Keywords:** Object-centric Radiance Fields · Slot-guided Feature Lifting

## 1 Introduction

The sense of objectness has been crucial to human cognition and generalization capabilities [33, 47]. Despite recent advances in visual perception [5, 15, 27, 40], achieving this generalization capability remains an unsolved challenge for existing models [26]. The pivotal role of object-centric understanding in human cognition necessitates models that can extract symbol-like object abstractions from complex visual signals, forming object-centric representations without supervision.

Recent years have witnessed substantial progress in object-centric learning [23, 30, 36, 44]. These methods aim to disentangle visual scenes into object-like entities for object-oriented reasoning and manipulation. Despite the remarkable progress made, existing approaches predominantly focus on 2D images. Since 2D images provide only partial views of the 3D physical world, object representations learned in the 2D domain are easily bound to 2D object attributes like colors [30], neglecting crucial information about object shape, geometry, and spatial relationships. Given the importance of these 3D attributes in representing the physical world, it is essential for models to form object abstractions in 3D environments to enhance understanding and interaction with the real world [16, 42].

To fulfill this goal, various attempts have been made to combine object-centric methods such as Slot-Attention [36] with 3D representations. Among them, multi-view image representations of 3D scenes [46, 48, 59] show competitive results on synthetic datasets given their effectiveness in preserving detailed object information. Nonetheless, translating the success of these methods from synthetic data to real-world scenarios has been proven to be non-trivial [43]. Specifically, aggregating information from multi-view real images and drawing correspondences between them naturally requires more intricate model designs. Meanwhile, decoding from object-centric representations to 3D (*e.g.*, novel views) places higher demands on the learned representations (*i.e.*, slots) as it now needs to infer about the 3D scene from a series of calibrated partial view projections. Recently, OSRT [42] scales up the dimensions of slots and reconstructs scenes with a Transformer-based encoder-decoder architecture, demonstrating powerful decomposition and reconstruction ability in complex 3D scenes. However, its success is built at the cost of inadmissible data and computation demands (64 TPUv2 chips for 7 days on 1M scenes). This urges the need for methods to effectively align information from calibrated multi-view images and reconstruct 3D scenes from the compressed object-centric representations.

In this work, we present SLOTLIFTER, a novel approach to learning object-centric representations in 3D scenes, inspired by recent advances in image-based rendering methods [6, 12, 22, 50, 51, 56, 58, 62]. In contrast to previous object-centric methods that focus solely on decoding information from slots, our method leverages lifted 2D input-view feature(s) to initialize 3D point features, which interact with the learned slot representations via a cross-attention-based transformer for predicting volume rendering parameters. This design enhances the granularity of details for novel-view synthesis while providing more explicit guidance for slot learning. Additionally, with no auxiliary losses needed, SLOTLIFTER relies only on the reconstruction loss and naturally requires less sampling overheads during training compared with existing 3D object-centric learning models like uORF and OSRT. This results in significantly fewer computational resources needed ( $\sim 5x$  faster) to achieve desirable outcomes. Through comprehensive experiments on four challenging synthetic and four complex real-world datasets, we observe consistent and significant performance improvement of SLOTLIFTER over existing 3D object-centric models on both scene decomposition ( $\sim 10+$  ARI) and novel-view synthesis ( $\sim 2+$  PSNR). We further show the effectiveness of each module through extensive ablative analyses and discussions, offering new insights into developing object-centric learning techniques for complex 3D scenes. In summary, our main contributions are as follows:

1. We propose SLOTLIFTER, a novel model for unsupervised object-centric learning in 3D scenes that effectively aggregates multi-view features for object-centric decoding via an innovative slot-guided feature lifting design.
2. We comprehensively evaluate SLOTLIFTER across four challenging synthetic and four real-world benchmarks. Our results consistently show that SLOTLIFTER significantly outperforms existing methods in both scene decomposition and novel-view synthesis, achieving state-of-the-art performance.

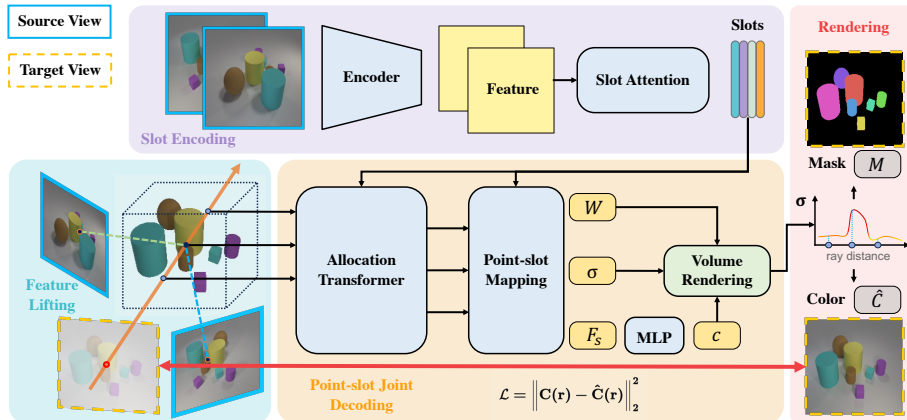
3. We conduct extensive ablative analyses demonstrating SLOTLIFTER’s potential in object-centric learning and image-based rendering, especially given its superior performance on established complex real-world datasets (*e.g.*, ScanNet and DTU) against state-of-the-art image-based rendering methods. We anticipate that our findings will stimulate further advancements in overcoming current limitations of 3D object-centric models.

## 2 Related Work

**Object-centric Learning** Prior studies in object-centric learning [3, 4, 13, 19, 20, 23–25, 35, 36, 64] have demonstrated proficiency in disentangling visual scenes into object-centric representations primarily on synthetic datasets, but they often struggle with handling complex real-world scenes. Notably, Slot-Attention [36] has fostered many powerful variants [5, 10, 17, 28, 30, 34, 43, 44, 52, 54] across various tasks and domains. However, these methods typically focus solely on learning object-centric representations from static images, thereby overlooking motion and 3D geometry information crucial for decomposing real-world complex scenes in an object-centric manner. Recognizing the potential benefits of motion information, [18, 32, 45] utilize video data to carve out object representations, demonstrating the effectiveness of the additional information provided beyond static images in the context of object-centric learning. Nonetheless, the use of 3D geometry information for object-centric learning has been largely left untouched. In this work, we pinpoint these crucial aspects by integrating advancements in image-based rendering with Slot-Attention, aiming to improve the acquisition of 3D object-centric representations within complex real-world environments.

**Novel-view Synthesis with NeRFs** Recent advances in Neural Radiance Field (NeRF) methods [2, 39, 49] have shown notable success in novel-view synthesis and 3D scene reconstruction. However, a significant drawback of these methods is the scene-specific long training time needed for optimizing each scene. The demand for better time efficiency has led to the emergence of generalizable NeRF methods [6, 8, 12, 22, 50, 51, 56, 58, 62]. These methods aim to synthesize novel views based on given images of scenes without per-scene optimization. For instance, PixelNeRF [58] and IBRNet [51] adopt volume rendering techniques, using features from nearby views to reconstruct novel views. MVSNeRF [6] constructs cost-volumes from nearby views for novel-view rendering. PointNeRF [56] leverages latent point clouds as anchors for radiance fields to improve both efficiency and performance. GNT [50] uses a transformer to integrate features from different views and demonstrates the powerful capability for generalizable novel-view synthesis. In contrast to these methods, SLOTLIFTER leverages an object-centric multi-view feature aggregation module and point-slot mapping module to more effectively encode 3D complex scenes for generalizable novel-view synthesis.

**3D Object-centric methods** Previous methods [42, 46, 48, 59] have attempted to extend Slot-Attention to 3D scenes for scene decomposition and novel-view synthesis. uORF [59], ObSuRF [48], sVORF [41], and uOCF [37] combine Slot-Attention



**Fig. 1: SLOTLIFTER overview.** SLOTLIFTER extracts slots from input view(s) during slot encoding. It then lifts 2D feature maps of input view(s) to initialize 3D point features, which serve as queries in the allocation transformer for point-slot joint decoding. This process yields the point-slot mapping  $W_p$ , density  $\sigma$ , and the slot-aggregated point feature  $F_s$  via an attention layer. Finally, SLOTLIFTER uses these results for rendering novel-view images and segmentation masks via volume rendering.

(or its variants) with NeRF [39] and use rendering losses as objectives for unsupervised slot learning. Additionally, OSRT [42] and COLF [46] further introduce Slot-Attention into the light field model to improve both model performance and inference speed. Nevertheless, uORF, COLF, and uOCF necessitate extra auxiliary losses, such as adversarial loss and LPIPS loss with a prolonged training period, which prevents downsampling rays and needs more computation. ObSuRF and uOCF require training with depth as a guidance signal. OSRT suffers from the heavy computation and training overhead required for properly reconstructing views from input pose and image embeddings. In contrast, SLOTLIFTER lifts the 2D multi-view feature to 3D and uses these point features to query multi-view information from the learned slots effectively. From our experiments, SLOTLIFTER not only outperforms previous 3D object-centric methods for unsupervised scene decomposition and novel-view synthesis but also obtains higher training efficiency.

### 3 SLOTLIFTER

In this section, we introduce our model, SLOTLIFTER, that combines object-centric learning modules with image-based rendering techniques. Our goal is to effectively learn scene reconstruction and decomposition by reconstructing input-view image(s). We present an overview of our SLOTLIFTER model in Fig. 1.

#### 3.1 Background

**Object-centric learning via Slot-Attention** Given  $N$  input feature vectors  $\mathbf{X} \in \mathbb{R}^{N \times D_f}$ , Slot-Attention [36] maps them to a set of  $K$  output vectors (*i.e.*,

slots)  $\mathbf{S} \in \mathbb{R}^{K \times D_s}$  via an iterative attention mechanism. The  $K$  slots compete to explain the input features  $\mathbf{X}$  by computing the attention matrix  $\mathbf{A}$  between  $\mathbf{S}$  and  $\mathbf{X}$ . The attention matrix is then used to aggregate feature vectors  $\mathbf{X}$  using a weighted mean. These aggregated features are embedded into slots  $\mathbf{S}$  by iteratively updating as follows:

$$\mathbf{A} = \text{softmax} \left( \frac{k(\mathbf{X}) \cdot q(\tilde{\mathbf{S}})^T}{\sqrt{D}} \right) \quad (1)$$

$$\mathbf{S} = \mathcal{U}_\theta(\tilde{\mathbf{S}}, \mathbf{W}^T v(\mathbf{X})), \text{ where } \mathbf{W}_{i,j} = \frac{\mathbf{A}_{i,j}}{\sum_{m=1}^N \mathbf{A}_{m,j}}.$$

$q(\cdot)$ ,  $k(\cdot)$ ,  $v(\cdot)$  are linear projections,  $\tilde{\mathbf{S}}$  denotes random initialized slots and  $\mathcal{U}_\theta(\cdot)$  represents the iterative update function often implemented with GRU [9], LayerNorm [1] and a residual MLP. As pointed out by Jia *et al.* [30], this iterative update process could be susceptible to instability when propagating gradients back into the iterative process. They therefore proposed a bi-level method, dubbed BO-QSA, to improve the optimization within Slot-Attention with learnable slot initialization instead of random sampled ones.

**Neural Radiance Fields** Given rays  $\{\mathbf{r}\}$  of a camera view, NeRF samples points along each ray and represent 3D scenes with a feature field  $\mathbf{F}_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$  mapping the 3D location  $\mathbf{x}$  and the view direction  $\mathbf{d}$  to color  $\mathbf{c}$  and volume density  $\sigma$ , and then renders the color of each ray via volume rendering [38]:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^N T_i [1 - \exp(-\sigma_i \delta_i)] \mathbf{c}_i, \quad (2)$$

where  $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j)$  and  $\delta_i$  is the distance between adjacent volumes along a ray. While NeRF achieves impressive novel-view synthesis quality, it adds stringent demands on model training given the number of points needed for approximating  $\hat{\mathbf{C}}(\mathbf{r})$  in Eq. (2). It also exhibits no generalization capabilities as each scene is optimized individually without shared prior knowledge.

### 3.2 Slot-guided Feature Lifting

**Scene Encoding** To render a novel target view  $\mathbf{I}_t$ , we leverage Slot-Attention to encode scene representations from  $L$  source view(s)  $\{\mathbf{I}_l\}_{l=1}^L$  ( $L = 1$  for single-view input) and lift 2D features to 3D for approximating the latent feature field  $\mathbf{F}_\Theta$ . We start by extracting 2D feature maps  $\{\mathbf{F}_l^{2D} \in \mathbb{R}^{H \times W \times D_f}\}_{l=1}^L$  from each source view. Next, we follow Eq. (1) to obtain object-centric scene features  $\mathbf{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_K\} \in \mathbb{R}^{K \times D_s}$  from these input 2D features via Slot-Attention. Inspired by image-based rendering methods, we consider constructing an additional 3D scene feature field by lifting 2D input-view features for capturing the fine-grained details in the input. Specifically, for target view  $\mathbf{I}_t$ , we sample points  $\mathbf{P} \in \mathbb{R}^{N \times 3}$  along

each ray  $\mathbf{r}$  and project each 3D point  $\mathbf{p} = (x, y, z)$  onto the image coordinates  $\pi(\mathbf{p}) = (x', y')$  to obtain its set of corresponding 2D features  $\mathbf{F}_{\text{lift}}(\mathbf{p})$  by:

$$\mathbf{F}_{\text{lift}}(\mathbf{p}) = [\mathbf{F}_1^{2D}[\pi(\mathbf{p})], \dots, \mathbf{F}_L^{2D}[\pi(\mathbf{p})]].$$

Without adding further ambiguity to the notations, we use  $\mathbf{F}_{\text{lift}} \in \mathbb{R}^{N \times L \times D_r}$  to represent the feature field obtained for all points in  $\mathbf{P}$ . After obtaining the lifted point features  $\mathbf{F}_{\text{lift}}$ , we pool the multi-view features to obtain 3D point features:

$$\mathbf{F}_p = \text{MLP}([\text{Mean}(\mathbf{F}_{\text{lift}}), \text{Var}(\mathbf{F}_{\text{lift}})]) + \mathbf{E}_p, \quad (3)$$

where  $\mathbf{E}_p \in \mathbb{R}^{N \times D_p}$  are positional embeddings for preserving the spatial information of 3D points. Notably, for single-view input, we ignore the variance term and let  $\mathbf{F}_p = \text{MLP}(\mathbf{F}_{\text{lift}}) + \mathbf{E}_p$ . This feature serves a similar role as  $\mathbf{F}_\Theta$  discussed in Eq. (2), providing fine-grained 3D features with spatial location considered.

**Point-slot Mapping** After scene encoding, given the slots  $\mathbf{S}$  and the point features  $\mathbf{F}_p$ , we design a point-slot joint decoding process to leverage both point and slot features for rendering. First, we calculate the point-slot mapping  $W_p$ , identifying the points that a slot  $\mathbf{s}_i \in \mathbf{S}$  contributes to. Specifically, we use a cross-attention-based allocation transformer, leveraging point features  $\mathbf{F}_p$  as queries and slot representations  $\mathbf{S}$  as keys and values to allocate slots to 3D points. As some points map to vacant areas in the 3D space, we add an additional learnable empty slot  $\mathbf{s}_\emptyset$  for these vacant points to query from. This process could be summarized as:

$$\mathbf{S}' = \{\mathbf{s}_\emptyset, \mathbf{s}_1, \dots, \mathbf{s}_k\}, \quad \tilde{\mathbf{F}}_p = \text{CrossAttn}(Q = \mathbf{F}_p, KV = \mathbf{S}').$$

After this process, the 3D point features  $\tilde{\mathbf{F}}_p$  contain information queried from object-centric slot representations. Finally, we obtain the point-slot mapping and the slot-aggregated point feature  $\mathbf{F}_s$  via an attention layer following:

$$\mathbf{F}_s = \mathbf{W}_p \mathbf{S}', \quad \text{where } \mathbf{W}_p = \text{softmax}\left(\frac{q(\tilde{\mathbf{F}}_p) \cdot k(\mathbf{S}')^T}{\sqrt{D}}\right).$$

We use  $q(\cdot)$ ,  $k(\cdot)$  to denote linear projections,  $\mathbf{W}_p \in \mathbb{R}^{N \times (K+1)}$  for the mapping weights from slots to points,  $D$  for the latent feature dimension. In essence, this process aims to obtain decodable 3D representations from learned slots. We can find the corresponding slot mapping (*i.e.*, contribution) weight from  $\mathbf{W}_p^i$  for each 3D point  $\mathbf{p}_i$ , thereby predicting its slot assignment for scene decomposition.

**Slot-based Density** For notation purposes, we use  $\mathbf{A}_p = q(\tilde{\mathbf{F}}_p) \cdot k(\mathbf{S}')^T$  throughout the subsequent texts for simplicity. To provide more direct guidance to slots, we use the attention weights  $\mathbf{A}_p$  from the mapping module to estimate the density value following [59]:

$$\sigma_i = \text{sum}(\mathbf{W}_p^{i,1:K+1} \odot \text{ReLU}(\mathbf{A}_p^{i,1:K+1})), \quad (4)$$

where  $i$  denotes  $i$ -th point,  $\odot$  denotes Hadamard production, and  $\mathbf{A}_p^{i,1:K+1}$  denotes the attention weights of the last  $K$  slots, ignoring the first empty slot in  $\mathbf{S}'$ . We add a ReLU layer over  $\mathbf{A}_p$  to suppress the contribution of slots less related to a specific point  $\mathbf{F}_p^i$  in density prediction. Finally, we add  $\mathbf{F}_s$  with the positional embedding  $\mathbf{E}_p$  and pass it into an MLP for predicting colors  $\mathbf{c}$ . Similarly, given the 3D point-slot mapping weight  $\mathbf{W}_p^i \in R^K$  of each point, SLOTLIFTER is able to render 2D segmentation masks  $\mathbf{M}$  using the same rendering scheme:

$$\begin{aligned} \mathbf{c} &= \text{MLP}(\mathbf{F}_s + \mathbf{E}_p), \quad \mathbf{C}(\mathbf{r}) = \sum_{i=1}^N T_i [1 - \exp(-\sigma_i \delta_i)] \mathbf{c}_i, \\ \mathbf{M}(\mathbf{r}) &= \sum_{i=1}^N T_i [1 - \exp(-\sigma_i \delta_i)] \mathbf{W}_p^i, \end{aligned} \quad (5)$$

where  $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j)$  and  $\delta_i$  is the distance between adjacent volumes along a ray following Eq. (2).

### 3.3 Training

**Objective** For training, we utilize the mean squared error (MSE) between the rendered rays  $\mathbf{C}(\mathbf{r})$  and the ground truth colors  $\hat{\mathbf{C}}(\mathbf{r})$  as our learning objective:

$$\mathcal{L}_{\text{recon}} = \|\mathbf{C}(\mathbf{r}) - \hat{\mathbf{C}}(\mathbf{r})\|^2.$$

**Random Masking** Although incorporating feature lifting into 3D object-centric learning improves the utilization of 3D information, it also poses a significant problem. Since both lifted point features  $\mathbf{F}_p$  and slot features  $\mathbf{S}$  originate from 2D multi-view images, the model can converge to degenerate scenarios, relying solely on lifted features for rendering and ignoring the information in slots. We avoid this degenerate case by randomly masking the lifted features in the sampled points, using only positional embeddings  $\mathbf{E}_p$  for these points to enforce alignment between slots and 3D point grids. In implementation, we use a cosine annealing schedule on the masking ratio from 0.99 to 0 for 30K steps.

## 4 Experiment

We present experimental results of SLOTLIFTER on **4 synthetic** and **4 complex real-world** datasets, evaluating its capability in novel view synthesis and unsupervised scene decomposition. The experimental settings are as follows:

**Datasets** For synthetic scenes, we evaluate SLOTLIFTER on 3 commonly used datasets CLEVR-567, Room-Chair, and Room-Diverse proposed by uORF [59]. We further select a more complex variant of Room-Diverse, Room-Texture [37], that provides synthetic rooms with real objects from ABO [11] for evaluating 3D object-centric learning. For complex real-world scenes, we use Kitchen-Shiny [37], Kitchen-Matte [37], ScanNet [14], and DTU MVS [29] to evaluate models' capability on novel-view synthesis and scene decomposition.

**Table 1: Quantitative comparison for segmentation in synthetic scenes.** SLOTLIFTER achieves the best performance on most metrics. Especially, when the dataset complexity increases (*e.g.*, from Room-Chair to Room-Diverse), SLOTLIFTER makes remarkable improvements (10+ ARI). We report all models with (mean  $\pm$  standard deviation) across 3 experiment trials except for sVORF where we report the best performance ( $\dagger$ ) adapted from the paper.

Method	CLEVR-567			Room-Chair			Room-Diverse		
	3D metric		2D metric	3D metric		2D metric	3D metric		2D metric
	NV-ARI $\uparrow$	ARI $\uparrow$	FG-ARI $\uparrow$	NV-ARI $\uparrow$	ARI $\uparrow$	FG-ARI $\uparrow$	NV-ARI $\uparrow$	ARI $\uparrow$	FG-ARI $\uparrow$
Slot-Attention [36]	-	3.5 $\pm$ 0.7	<b>93.2<math>\pm</math>1.5</b>	-	38.4 $\pm$ 18.4	40.2 $\pm$ 4.5	-	17.4 $\pm$ 11.3	43.8 $\pm$ 11.7
uORF [59]	83.8 $\pm$ 0.3	86.3 $\pm$ 0.1	87.4 $\pm$ 0.8	74.3 $\pm$ 1.9	78.8 $\pm$ 2.6	88.8 $\pm$ 2.7	56.9 $\pm$ 0.2	65.6 $\pm$ 1.0	67.8 $\pm$ 1.7
BO-uORF [30]	78.4 $\pm$ 0.7	87.4 $\pm$ 0.5	89.2 $\pm$ 0.3	80.9 $\pm$ 0.2	82.2 $\pm$ 1.0	91.6 $\pm$ 2.3	62.5 $\pm$ 0.5	72.6 $\pm$ 0.2	76.8 $\pm$ 0.2
COLF [46]	55.8 $\pm$ 0.1	69.0 $\pm$ 0.4	92.4 $\pm$ 1.7	80.7 $\pm$ 0.1	85.6 $\pm$ 0.04	89.8 $\pm$ 0.1	52.5 $\pm$ 0.3	66.5 $\pm$ 0.4	64.7 $\pm$ 0.7
SLOTLIFTER	<b>87.0<math>\pm</math>2.5</b>	<b>93.7<math>\pm</math>1.1</b>	91.3 $\pm$ 1.6	<b>89.7<math>\pm</math>0.5</b>	<b>92.6<math>\pm</math>0.3</b>	<b>91.9 <math>\pm</math>0.3</b>	<b>77.5<math>\pm</math>0.7</b>	<b>90.0<math>\pm</math>0.8</b>	<b>84.3<math>\pm</math>2.7</b>
sVORF $\dagger$ [41]	81.5	82.7	92.0	87.0	87.8	<b>92.4</b>	75.6	78.4	86.6
SLOTLIFTER $\dagger$	<b>89.0</b>	<b>94.6</b>	<b>93.1</b>	<b>90.3</b>	<b>92.9</b>	92.1	<b>78.1</b>	<b>90.6</b>	<b>86.7</b>

**Metrics** We evaluate the quality of novel-view synthesis with three common metrics: LPIPS [61], SSIM [53], and PSNR. In particular, we use LPIPS<sub>alex</sub> for synthetic scenes and LPIPS<sub>vgg</sub> for real-world scenes to be consistent with previous methods. Following [46, 59], we evaluate the quality of scene decomposition with four metrics: Adjusted Rand Index (ARI), FG-ARI (*i.e.*, ARI computed only on foreground objects), NV-ARI (*i.e.*, ARI on novel views), and NV-FG-ARI.

#### 4.1 Object-centric Learning in Synthetic Scenes

**Setup** To perform a fair comparison between SLOTLIFTER and existing methods, we follow the setup of uORF [59] and use only **one source view** as input to render the other novel views. As we only use a single source view, we modify the multi-view feature aggregation to  $F_p = \text{MLP}(F_{\text{lift}}) + E_p$  as discussed in Sec. 3.2. We train our model using the Lion [7] optimizer with a learning rate of  $5 \times 10^{-5}$  for 250k iterations. We use a batch size of 4 and sample 1024 rays for each scene.

**Baselines** We compare SLOTLIFTER with previous state-of-the-art 3D object-centric methods including uORF [59], COLF [46], and sVORF [41]. We also report the results of the improved uORF (BO-uORF) introduced by Jia *et al.* [30] as a competitive baseline in evaluating the results on these datasets.

**Results and Analysis** We evaluate the performance of SLOTLIFTER for unsupervised scene decomposition and present our quantitative results in Tab. 1 and Tab. 2. SLOTLIFTER outperforms existing 3D object-centric learning methods, achieving the best performance across all datasets. We also visualize qualitative results for segmentation in Fig. 2 and Fig. 3. As

**Table 2: Quantitative comparison for scene decomposition and novel view synthesis on Room-Texture.**

Method	Scene segmentation			Novel view synthesis		
	NV-ARI $\uparrow$	ARI $\uparrow$	FG-ARI $\uparrow$	LPIPS $\downarrow$	SSIM $\uparrow$	PSNR $\uparrow$
uORF [59]	57.8	67.0	9.3	0.254	0.711	24.23
BO-uORF [30]	60.4	69.7	35.4	0.215	0.739	25.26
COLF [46]	1.1	23.5	53.2	0.504	0.670	22.98
uOCF-N [37]	72.2	79.1	58.4	0.138	0.796	28.81
uOCF-P [37]	70.4	78.5	56.3	0.136	0.798	28.85
SLOTLIFTER	<b>79.3</b>	<b>86.0</b>	<b>70.7</b>	<b>0.131</b>	<b>0.858</b>	<b>30.68</b>



**Table 3: Quantitative comparison for novel-view synthesis in synthetic scenes.** SLOTLIFTER outperforms existing methods on the majority of metrics in three datasets, rendering novel views of much higher quality, especially for complex datasets.

Method	CLEVR-567			Room-Chair			Room-Diverse		
	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑
NeRF-AE [59]	0.1288	0.8658	27.16	0.1166	0.8265	28.13	0.2458	0.6688	24.80
uORF [59]	0.0859	0.8971	29.28	0.0821	0.8722	29.60	0.1729	0.7094	25.96
BO-uORF [30]	0.0618	0.9260	30.85	0.0733	0.8938	30.61	0.1515	0.7363	26.96
COLF [46]	0.0608	0.9346	31.81	0.0485	0.8934	30.93	0.1274	0.7308	26.02
sVORF [41]	0.0211	<b>0.9701</b>	<b>37.20</b>	0.0824	0.8992	33.04	0.1637	0.7825	29.41
SLOTLIFTER	<b>0.0184</b>	0.9680	36.09	<b>0.0410</b>	<b>0.9358</b>	<b>34.63</b>	<b>0.1159</b>	<b>0.8479</b>	<b>29.97</b>

shown in Tab. 1 and Tab. 2, SLOTLIFTER significantly outperforms current state-of-the-art methods by a large margin on all datasets. We also observe from Fig. 2 and Fig. 3 that SLOTLIFTER better handles occlusion between objects, offering more complete segmentation. Notably, compared with task-specific auxiliary designs in current baselines (*e.g.*, adversarial loss used in uORF), SLOTLIFTER models each slot equivalently and relies solely on the reconstruction loss  $\mathcal{L}_{\text{recon}}$  for achieving the good performance. We attribute this effectiveness to our scene encoding design and provide more analyses in Sec. 4.3.

We also evaluate the capability of our SLOTLIFTER for novel-view synthesis and present our quantitative results compared with existing methods in Tab. 2, Tab. 3, and visualize qualitative results in Fig. 2, Fig. 3. As shown in Tab. 2 and Tab. 3, our model outperforms existing methods on almost all metrics across the four datasets, rendering novel views of much higher quality, especially for complex datasets. As visualized in Fig. 2 and Fig. 3, SLOTLIFTER captures more detailed texture, shape, and pose of objects compared with baseline models.

Additionally, compared to uORF [59] that needs to train for 6 days on Room-Diverse with a single Nvidia RTX 3090 GPU, SLOTLIFTER is more efficient, requiring only 30 hours (5x speed up) training time. This is afforded by: (i) the feature lifting design provides detailed information for rendering and leads to a faster model convergence rate; (ii) the slot-based density prediction and rendering in SLOTLIFTER requires only 1 radiance field while models like uORF, uOCF, and sVORF compute K fields for each slot; (iii) with no auxiliary losses on the fully rendered image, SLOTLIFTER only needs 1024 (or even 256) sampled rays for training with the reconstruction loss, thus largely reducing the computation overhead. Please refer to Tab. A.3 in the *supplementary* for more comparisons.

## 4.2 Object-centric Learning in Real-world Scenes

**Setup** To show the effectiveness of SLOTLIFTER on real-world complex scenes, we evaluate SLOTLIFTER on Kitchen-Shiny and Kitchen-Matte following uOCF [37]. We use the same train/test split for these two datasets with **single-view input** following settings in uOCF. Unlike uOCF which requires training with 2 stages to learn object priors, we train SLOTLIFTER with reconstruction loss in 1 stage.



**Fig. 2: Qualitative comparison on synthetic scenes.** Compared to BO-uORF, SLOTLIFTER renders novel-view images and segmentation masks in much higher quality, especially in detailed object attributes like color and shape (best viewed with zoom-in for the **highlighted details**).

We also consider ScanNet [14] and DTU [29], which are well-established datasets for evaluating generalizable novel-view synthesis [22, 55, 62], as more challenging real-world benchmarks to test models’ capability on processing complex real-world scenes. For ScanNet, we follow the standard training and evaluation scheme in existing works [55, 62], sample 100 scenes for training, and evaluate our method on the 8 unseen testing scenes introduced. On DTU, we follow the setup of PixelNeRF [58] and NeRFusion [62], train all models on the 88 training scenes, and test on the 15 test scenes. For both ScanNet and DTU, we follow the standard setting in generalizable novel-view synthesis and provide **4 source nearby views** selected according to previous work [6, 22, 50, 51, 56, 62] as inputs.

**Baselines** For evaluating object-centric learning, we compare our SLOTLIFTER with existing state-of-the-art 3D object-centric models, including uORF, BO-uORF, COLF, and uOCF on Kitchen-Shiny and Kitchen-Matte. On ScanNet, we mainly compare the SLOTLIFTER with the improved uORF model for object-centric learning as uOCF requires a two-stage training scheme with auxiliary losses thus not directly comparable. We additionally add OSRT [42] as a powerful baseline as it has demonstrated its effectiveness in decomposing complex scenes.

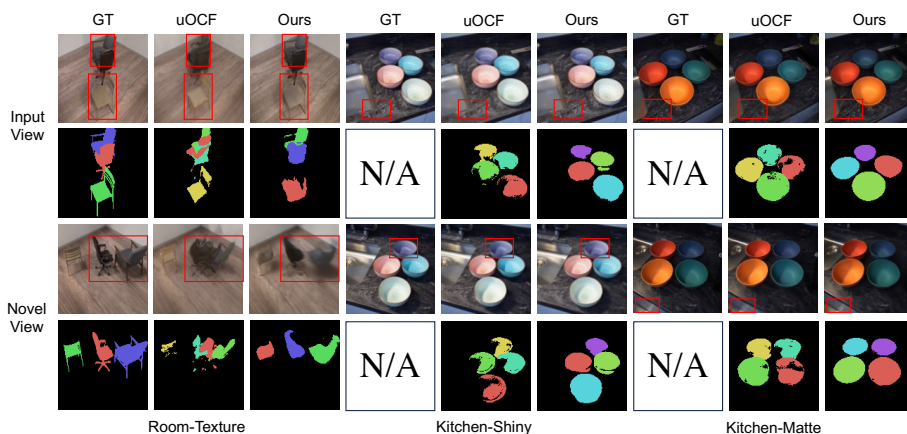
For generalizable novel-view synthesis, compare SLOTLIFTER and state-of-the-art generalizable NeRFs like NeRFusion [62] on ScanNet and DTU MVS. Additionally, we re-train the recent state-of-the-art method GNT [50] for generalizable

**Table 4: Quantitative comparison for novel view synthesis on Kitchen-Shiny and Kitchen-Matte.** SLOTLIFTER presents significant improvements ( $\sim 4$  PSNR) and the best results on all perceptual scores.

Method	Kitchen-Shiny			Kitchen-Matte		
	LPIPS $\downarrow$	SSIM $\uparrow$	PSNR $\uparrow$	LPIPS $\downarrow$	SSIM $\uparrow$	PSNR $\uparrow$
uORF [59]	0.336	0.602	19.23	0.092	0.808	26.07
BO-uORF [30]	0.318	0.639	19.78	0.067	0.832	27.36
COLF [46]	0.397	0.561	18.30	0.236	0.643	20.68
uOCF-N [37]	0.055	0.842	27.87	0.055	0.841	28.25
uOCF-P [37]	0.049	0.862	28.58	0.043	0.867	29.40
<b>SLOTLIFTER</b>	<b>0.035</b>	<b>0.928</b>	<b>32.02</b>	<b>0.030</b>	<b>0.939</b>	<b>32.92</b>

**Table 5: Quantitative comparison on ScanNet.**  $\dagger$  We use the official implementations provided to re-train and evaluate the models on ScanNet.

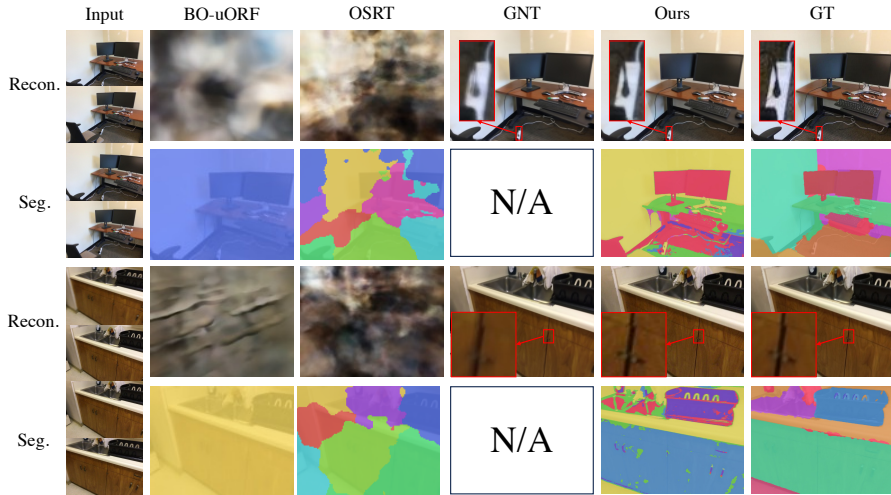
Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	NV-FG-ARI $\uparrow$
IBRNet [51]	21.19	0.786	0.358	-
NeRFusion [62]	22.99	0.838	0.335	-
PointNeRF [56]	20.47	0.642	0.544	-
SurfelNeRF [22]	23.82	0.845	0.327	-
GNT $\dagger$ [50]	27.76	0.8791	0.2197	-
BO-uORF $\dagger$ [59]	12.72	0.3393	0.6975	0.0
OSRT $\dagger$ [42]	13.34	0.2746	0.6337	29.7
<b>SLOTLIFTER</b>	<b>28.36</b>	<b>0.9200</b>	<b>0.1891</b>	<b>31.1</b>



**Fig. 3: Qualitative comparison on Room-Texture, Kitchen-Shiny, and Kitchen-Matte.** Compared to the SOTA method uOCF, SLOTLIFTER renders novel-view images and segmentation masks in higher quality, offering more complete segmentation and more detailed textures (best viewed with zoom-in for the **highlighted details**).

novel-view synthesis on ScanNet as a strong baseline to validate the effectiveness of our method (see more implementation details in Appendix A.2).

**Results and Analysis** We present quantitative evaluations in Tab. 4 and Tab. 5, and visualize qualitative results in Fig. 3 and Fig. 4. Similar to results in synthetic datasets, we observe a consistent improvement in object-centric learning on real-world datasets. In Kitchen-Shiny and Kitchen-Matte, as there is no ground truth segmentation annotation available, we qualitatively compare SLOTLIFTER with existing methods in Fig. 3. We demonstrate that SLOTLIFTER renders segmentation masks with higher quality, offering more complete object segmentations. The quantitative evaluation results on ScanNet in Tab. 5 also demonstrate that SLOTLIFTER outperforms existing 3D object-centric methods with more accurate segmentation masks predicted as shown in Fig. 4. Notably, Fig. 4 also shows that despite the relatively marginal performance gap (compared with improvements in synthetic datasets) between OSRT and SLOTLIFTER in NV-ARI-FG, OSRT



**Fig. 4: Qualitative results on ScanNet.** Our SLOTLIFTER achieves the best performance for novel-view rendering, even surpassing the recent state-of-the-art model GNT, while BO-uORF and OSRT struggle to render novel-view images on ScanNet.

**Table 6: Quantitative comparison on DTU.**

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
PixelNeRF [58]	19.31	0.789	0.382
IBRNet [51]	26.04	0.917	0.190
MVSNerF [6]	26.63	<b>0.931</b>	0.168
NeRFusion [62]	26.19	0.922	0.177
SLOTLIFTER	<b>26.75</b>	0.896	<b>0.157</b>

**Table 7: Sensitivity of random masking ratio scheduling.**

Decay Steps	PSNR $\uparrow$	NV-ARI $\uparrow$	ARI $\uparrow$	FG-ARI $\uparrow$
0	29.89	74.4	85.8	43.6
10000	<b>30.01</b>	74.9	86.9	42.1
30000	29.80	<b>77.5</b>	<b>90.3</b>	84.8
60000	29.53	77.3	90.1	<b>85.7</b>
100000	28.68	76.4	89.4	83.6

generates uniformly distributed masks without properly separating the objects. This originates from an unfair privilege of OSRT when calculating ARI as this metric mainly considers coverage as an important factor. We provide further analyses and discussions on improving SLOTLIFTER for complex real-world scenes in Appendix B.1.

As shown in Tabs. 4-6, we observe a consistent advantage of SLOTLIFTER on most datasets for novel-view synthesis. This includes outperforming state-of-the-art methods dedicatedly designed for generalizable novel-view synthesis like SurfelNeRF and GNT. Meanwhile, Tab. 5 and Fig. 4 show that methods like BO-uORF and OSRT struggle to render novel-view images in complex settings, achieving only a PSNR of less than 14 with no meaningful rendered results. Notably, OSRT achieves a PSNR of 27 on training scenes but fails to generalize to unseen scenes (see more discussions in Appendix B.2). These results further validate the effectiveness of SLOTLIFTER compared with previous 3D object-centric learning methods.

**Table 8: Ablations analysis of module designs in SLOTLIFTER.**

Method	Room-Diverse						ScanNet			
	LPIPS↓	SSIM↑	PSNR↑	NV-ARI↑	ARI↑	FG-ARI↑	LPIPS↓	SSIM↑	PSNR↑	NV-FG-ARI↑
w/o Feature Lift.	0.2537	0.7716	28.20	71.4	75.8	65.3	0.5622	0.5129	11.60	0.0
w/o Random Mask	<b>0.1169</b>	<b>0.8470</b>	<b>29.89</b>	74.4	85.8	43.6	<b>0.1861</b>	<b>0.9208</b>	27.86	17.63
w/o Slot Density	0.1180	0.8456	29.82	76.3	87.6	77.3	0.1937	0.9134	27.42	6.6
FullModel	0.1180	0.8454	29.80	<b>77.5</b>	<b>90.3</b>	<b>84.8</b>	0.1891	0.9200	<b>28.36</b>	<b>31.1</b>

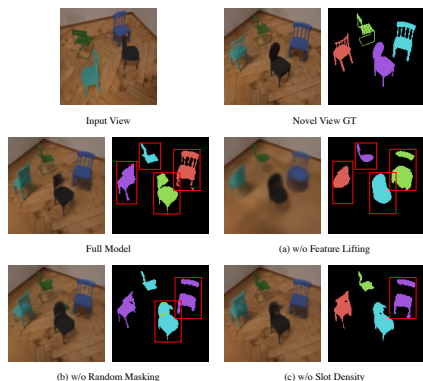
### 4.3 Ablative Study

To investigate the effectiveness of our designs in SLOTLIFTER, including scene encoding, random masking, slot-based density, and the number of slots and source views, we conduct ablative studies on both synthetic (Room-Diverse) and real-world (ScanNet) scenes. We also investigate the effect of the number of sampled rays and leave the results in Tab. A.4 in the supplementary.

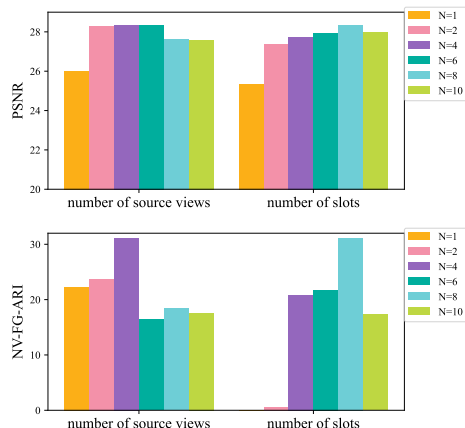
**Scene Encoding** We consider removing the feature lifting operation and initializing point features solely with positional embeddings, *i.e.*,  $\mathbf{F}_p = \mathbf{E}_p$ . As shown in Tab. 8 and Fig. 5, the performance of both novel-view synthesis and scene decomposition on Room-Diverse drops significantly without lifted multi-view features, especially for LPIPS and FG-ARI. In fact, it is hard to establish the mapping between slots and 3D points via only positional information. This problem is more severe in complex real-world scenes (*e.g.*, Scannet), where SLOTLIFTER struggles in rendering novel views without feature lifting, achieving only a PSNR of 11.6. This issue is also shared by uORF and OSRT as presented in Sec. 4.2 and demonstrates the significance of the feature lifting design.

**Random Masking** As shown in Tab. 8 and Fig. 5, abandoning the random masking scheme described in Sec. 3.3 slightly improves the rendering performance (LPIPS, SSIM, PSNR) but significantly decreases the scene decomposition capability of SLOTLIFTER, especially for FG-ARI. We also find that the model sometimes converges to the degenerate scenario as discussed in Sec. 3.3 without the random masking scheme, leading to a collapse in scene decomposition (*i.e.*, uniform segmentation predictions) with ARI scores lower than 40. This affirms our supposition that, without random masking, the model is likely to degenerate and rely solely on lifted features for rendering, thereby ignoring the information in slots. We also explore how the masking ratio decay scheduling influences performance. As shown in Tab. 7, increasing decay steps slightly harms rendering performance and significantly improves segmentation performance after a certain amount of steps ( $\sim 10K$  steps). After the number of decay steps exceeds 30K, continuing to increase the number of steps will only bring marginal improvement.

**Slot-based Density** As shown in Tab. 8, compared with using an additional MLP layer for predicting the density value, using slot-based density slightly improves the quality of novel-view synthesis on ScanNet and significantly improves the performance of scene decomposition on both datasets, especially for ScanNet. We attribute this effectiveness to the fact that the slot-based density is more involved in point-slot interactions. This leads to more information propagation



**Fig. 5: Visualization of model ablation analysis.** (a) Without feature lifting, SLOTLIFTER renders blurred images and imprecise segmentation masks. (b) Without random masking, SLOTLIFTER cannot segment objects correctly. (c) Using slot-based density helps SLOTLIFTER learn more accurate segmentation.



**Fig. 6: Ablative studies over the number of source views and slots.** We set the number of slots to 8 for different numbers of source views and set the number of source views to 4 for different numbers of slots.

to slots, thus improving the learned object-centric representations for accurately segmenting foreground objects.

**Sensitivity to Number of Slots and Source Views** As discussed in Sec. 3.2, SLOTLIFTER can accept a various number of source views as input. We investigate how the number of slots and source views influences the performance of SLOTLIFTER on ScanNet. As shown in Fig. 6, SLOTLIFTER is sensitive to the number of slots, which is consistent with previous research on Slot-Attention. In addition, the number of source views also has a significant impact on model performance, as it influences both the extracted slots and the lifted 3D point features which are essential components for slot-guided feature lifting in SLOTLIFTER.

## 5 Conclusion

We present SLOTLIFTER, an object-centric radiance field model for unsupervised 3D object-centric representation learning. Our SLOTLIFTER employs slot-guided feature lifting to improve the interaction between lifted input view features and learned slots during decoding. SLOTLIFTER achieves state-of-the-art performance with large improvements on four challenging synthetic and four complex real-world datasets for scene decomposition and novel-view synthesis and uses much less training time, demonstrating its effectiveness and efficiency. Furthermore, SLOTLIFTER demonstrates superior performance for novel-view synthesis on real-world datasets, underscoring its potential to narrow the gap to real-world scenes.

## Acknowledgement

We gratefully thank all colleagues from BIGAI for fruitful discussions. We would also like to thank the anonymous reviewers for their constructive feedback. This work reported herein was supported by Beijing Natural Science Foundation (QY23126).

## References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
3. Bear, D., Fan, C., Mrowca, D., Li, Y., Alter, S., Nayebi, A., Schwartz, J., Fei-Fei, L.F., Wu, J., Tenenbaum, J., et al.: Learning physical graph representations from visual scenes. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS) (2020)
4. Burgess, C.P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., Lerchner, A.: Monet: Unsupervised scene decomposition and representation. arXiv preprint arXiv:1901.11390 (2019)
5. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of International Conference on Computer Vision (ICCV) (2021)
6. Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., Su, H.: Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In: Proceedings of International Conference on Computer Vision (ICCV) (2021)
7. Chen, X., Liang, C., Huang, D., Real, E., Wang, K., Liu, Y., Pham, H., Dong, X., Luong, T., Hsieh, C.J., et al.: Symbolic discovery of optimization algorithms. arXiv preprint arXiv:2302.06675 (2023)
8. Chen, Y., Ni, J., Jiang, N., Zhang, Y., Zhu, Y., Huang, S.: Single-view 3d scene reconstruction with high-fidelity shape and texture. In: Proceedings of International Conference on 3D Vision (3DV) (2024)
9. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
10. Choudhury, S., Laina, I., Rupprecht, C., Vedaldi, A.: Unsupervised part discovery from contrastive reconstruction. Proceedings of Advances in Neural Information Processing Systems (NeurIPS) (2021)
11. Collins, J., Goel, S., Deng, K., Luthra, A., Xu, L., Gundogdu, E., Zhang, X., Vicente, T.F.Y., Dideriksen, T., Arora, H., et al.: Abo: Dataset and benchmarks for real-world 3d object understanding. In: Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
12. Cong, W., Liang, H., Wang, P., Fan, Z., Chen, T., Varma, M., Wang, Y., Wang, Z.: Enhancing nerf akin to enhancing llms: Generalizable nerf transformer with mixture-of-view-experts. In: Proceedings of International Conference on Computer Vision (ICCV) (2023)
13. Crawford, E., Pineau, J.: Spatially invariant unsupervised object detection with convolutional neural networks. In: Proceedings of AAAI Conference on Artificial Intelligence (AAAI) (2019)

14. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
15. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
16. Driess, D., Xia, F., Sajjadi, M.S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al.: Palm-e: An embodied multimodal language model. arXiv preprint arXiv:2303.03378 (2023)
17. Du, Y., Li, S., Sharma, Y., Tenenbaum, J., Mordatch, I.: Unsupervised learning of compositional energy concepts. Proceedings of Advances in Neural Information Processing Systems (NeurIPS) (2021)
18. Elsayed, G.F., Mahendran, A., van Steenkiste, S., Greff, K., Mozer, M.C., Kipf, T.: Savi++: Towards end-to-end object-centric learning from real-world videos. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS) (2022)
19. Engelcke, M., Kosiorek, A.R., Jones, O.P., Posner, I.: Genesis: Generative scene inference and sampling with object-centric latent representations. In: Proceedings of International Conference on Learning Representations (ICLR) (2020)
20. Eslami, S., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., Hinton, G.E., et al.: Attend, infer, repeat: Fast scene understanding with generative models. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS) (2016)
21. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
22. Gao, Y., Cao, Y.P., Shan, Y.: Surfelfnerf: Neural surfel radiance fields for online photorealistic reconstruction of indoor scenes. In: Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
23. Greff, K., Kaufman, R.L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., Lerchner, A.: Multi-object representation learning with iterative variational inference. In: Proceedings of International Conference on Machine Learning (ICML) (2019)
24. Greff, K., Rasmus, A., Berglund, M., Hao, T., Valpola, H., Schmidhuber, J.: Tagger: Deep unsupervised perceptual grouping. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS) (2016)
25. Greff, K., Van Steenkiste, S., Schmidhuber, J.: Neural expectation maximization. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS) (2017)
26. Greff, K., Van Steenkiste, S., Schmidhuber, J.: On the binding problem in artificial neural networks. arXiv preprint arXiv:2012.05208 (2020)
27. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
28. Hénaff, O.J., Koppula, S., Shelhamer, E., Zoran, D., Jaegle, A., Zisserman, A., Carreira, J., Arandjelović, R.: Object discovery and representation networks. In: Proceedings of European Conference on Computer Vision (ECCV) (2022)
29. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR) (2014)



30. Jia, B., Liu, Y., Huang, S.: Improving object-centric learning with query optimization. In: Proceedings of International Conference on Learning Representations (ICLR) (2023)
31. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)* **42**(4) (2023)
32. Kipf, T., Elsayed, G.F., Mahendran, A., Stone, A., Sabour, S., Heigold, G., Jonchkowski, R., Dosovitskiy, A., Greff, K.: Conditional object-centric learning from video. In: Proceedings of International Conference on Learning Representations (ICLR) (2022)
33. Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J.: Building machines that learn and think like people. *Behavioral and brain sciences* **40**, e253 (2017)
34. Lamb, A., He, D., Goyal, A., Ke, G., Liao, C.F., Ravanelli, M., Bengio, Y.: Transformers with competitive ensembles of independent mechanisms. *arXiv preprint arXiv:2103.00336* (2021)
35. Lin, Z., Wu, Y.F., Peri, S.V., Sun, W., Singh, G., Deng, F., Jiang, J., Ahn, S.: Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. In: Proceedings of International Conference on Learning Representations (ICLR) (2020)
36. Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., Kipf, T.: Object-centric learning with slot attention. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS) (2020)
37. Luo, R., Yu, H.X., Wu, J.: Unsupervised discovery of object-centric neural fields. *arXiv preprint arXiv:2402.07376* (2024)
38. Max, N.: Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics* **1**(2), 99–108 (1995)
39. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* **65**(1), 99–106 (2021)
40. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193* (2023)
41. Qi, D., Yang, T., Zhang, X.: Slot-guided volumetric object radiance fields. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS) (2023)
42. Sajjadi, M.S., Duckworth, D., Mahendran, A., van Steenkiste, S., Pavetić, F., Lučić, M., Guibas, L.J., Greff, K., Kipf, T.: Object scene representation transformer. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS) (2022)
43. Seitzer, M., Horn, M., Zadaianchuk, A., Zietlow, D., Xiao, T., Simon-Gabriel, C., He, T., Zhang, Z., Schölkopf, B., Brox, T., et al.: Bridging the gap to real-world object-centric learning. In: Proceedings of International Conference on Learning Representations (ICLR) (2023)
44. Singh, G., Deng, F., Ahn, S.: Illiterate dall-e learns to compose. In: Proceedings of International Conference on Learning Representations (ICLR) (2021)
45. Singh, G., Wu, Y.F., Ahn, S.: Simple unsupervised object-centric learning for complex and naturalistic videos. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS) (2022)

46. Smith, C., Yu, H.X., Zakharov, S., Durand, F., Tenenbaum, J.B., Wu, J., Sitzmann, V.: Unsupervised discovery and composition of object light fields. *Transactions on Machine Learning Research (TMLR)* (2023)
47. Spelke, E.S., Kinzler, K.D.: Core knowledge. *Developmental science* **10**(1), 89–96 (2007)
48. Stelzner, K., Kersting, K., Kosiorek, A.R.: Decomposing 3d scenes into objects via unsupervised volume segmentation. *arXiv preprint arXiv:2104.01148* (2021)
49. Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P.P., Barron, J.T., Kretzschmar, H.: Block-nerf: Scalable large scene neural view synthesis. In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)* (2022)
50. Varma, M., Wang, P., Chen, X., Chen, T., Venugopalan, S., Wang, Z.: Is attention all that nerf needs? In: *Proceedings of International Conference on Learning Representations (ICLR)* (2022)
51. Wang, Q., Wang, Z., Genova, K., Srinivasan, P.P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.: Ibrnet: Learning multi-view image-based rendering. In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)* (2021)
52. Wang, Y., Shen, X., Hu, S.X., Yuan, Y., Crowley, J.L., Vaufreydaz, D.: Self-supervised transformers for unsupervised object discovery using normalized cut. In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)* (2022)
53. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* **13**(4), 600–612 (2004)
54. Wang, Z., Shou, M.Z., Zhang, M.: Object-centric learning with cyclic walks between parts and whole. *arXiv preprint arXiv:2302.08023* (2023)
55. Wei, Y., Liu, S., Rao, Y., Zhao, W., Lu, J., Zhou, J.: Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In: *Proceedings of International Conference on Computer Vision (ICCV)* (2021)
56. Xu, Q., Xu, Z., Philip, J., Bi, S., Shu, Z., Sunkavalli, K., Neumann, U.: Point-nerf: Point-based neural radiance fields. In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)* (2022)
57. Yang, J., Pavone, M., Wang, Y.: Freenerf: Improving few-shot neural rendering with free frequency regularization. In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)* (2023)
58. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)* (2021)
59. Yu, H.X., Guibas, L.J., Wu, J.: Unsupervised discovery of object radiance fields. In: *Proceedings of International Conference on Learning Representations (ICLR)* (2022)
60. Zadaianchuk, A., Seitzer, M., Martius, G.: Object-centric learning for real-world videos by predicting temporal feature similarities. In: *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)* (2023)
61. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)* (2018)
62. Zhang, X., Bi, S., Sunkavalli, K., Su, H., Xu, Z.: Nerfusion: Fusing radiance fields for large-scale scene reconstruction. In: *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)* (2022)

63. Zheng, S., Zhou, B., Shao, R., Liu, B., Zhang, S., Nie, L., Liu, Y.: Gps-gaussian: Generalizable pixel-wise 3d gaussian splatting for real-time human novel view synthesis. arXiv preprint arXiv:2312.02155 (2023)
64. Zoran, D., Kabra, R., Lerchner, A., Rezende, D.J.: Parts: Unsupervised segmentation with slots, attention and independence maximization. In: Proceedings of International Conference on Computer Vision (ICCV) (2021)

# SLOTLIFTER: Slot-guided Feature Lifting for Learning Object-centric Radiance Fields

## Supplementary Material

### A Implementation Details

#### A.1 SLOTLIFTER

##### Architecture Design

**Scene Encoding** We employ a U-net-like encoder  $E_\phi$  with ResNet34 [27] to extract 2D image features, similar to IBRNet [51]. This architecture truncates after `layer3` as the encoder and adds two up-sampling layers with convolutions and skip-connections as the decoder. Instead of extracting two sets of feature maps for coarse and fine networks as IBRNet, we extract a shared feature map. In addition, we concatenate multi-view images with their corresponding ray directions and camera positions to provide more spatial information, enabling slots to learn 3D information from 2D multi-view features via Slot-Attention. Given extracted feature maps, we obtain slots via Slot-Attention and 3D point features via feature lifting described in Sec. 3.2. We add the point positional embedding  $E_p$  in Eq. (3), which considers point location  $\mathbf{p}$  and ray direction  $\mathbf{d}$  simultaneously by:

$$E_p = \text{MLP}(\text{Concat}([\text{PosEmb}(\mathbf{p}), \text{PosEmb}(\mathbf{d})])),$$

where PosEmb is a Fourier transformation with a frequency of 10 while MLP is used to fuse point location and ray direction information and project positional embedding to the same dimension as point features.

**Point-slot Joint Decoding** Our point-slot joint decoding contains an allocation transformer and an attention-based point-slot mapping module. The allocation transformer consists of four transformer layers, and each layer includes a cross-attention layer, a 1D convolution layer, and a self-attention layer. We use 1D convolution and self-attention to model the relationship among points along a ray. The design is based on the insight that spatially adjacent points are more likely to be associated with the same slot. Additionally, We use the weighted sum of attention weights to estimate the density value in Eq. (4). As this design may restrict the scale of density by the attention weights between slots and point features, the density obtained from Eq. (4) is multiplied by a learnable parameter  $s_\sigma$  to rescale it.

**Hyperparameters and Training Details** We train our SLOTLIFTER by sampling 1024 rays for each scene with a learning rate of  $5 \times 10^{-5}$ , a linear learning rate warm-up of 10000 steps, and an exponentially decaying schedule.

**Table A.1:** Training configuration for our SLOTLIFTER. The values in parentheses are adopted for the ScanNet and DTU datasets.

Training	Scene Batch Size	4 (2)
	Ray Batch Size	1024
	LR	5e-5
	LR Warm-up Steps	10000
	LR Decay Steps	50000
	Max Steps	250K
	Num. Source Views	1 (4)
	Grad. Clip	0.5
Scene Encoding	Feature Dimension	64 (32)
	Slot Dimension	256
	Iterations	3
	$\sigma$ Annealing Steps	30000
Point-slot Decoding	Num. Layers	4
	Attention Heads	4
	Feature Dimension	64

**Table A.2:** Image resolution and the number of slots on different datasets.

Dataset	CLEVR567	Room-Chair	Room-Diverse	Room-Texture
Resolution	128×128	128×128	128×128	128×128
Number of slots	8	5	5	5
Dataset	Kitchen-Shiny	Kitchen-Matte	DTU MVS	ScanNet
Resolution	128×128	128×128	400×300	640×480
Number of slots	5	5	8	8

All the images are resized to 128×128 for synthetic data and 640×480 for real-world data. Image resolution and the number of slots  $K$  used on different datasets are shown in Tab. A.2. To encourage SLOTLIFTER to segment the background properly, we use the locality constraint proposed by uORF [59]. Specifically, we set a background bound and enforce every point outside the bound being mapped to the empty slot or the first slot. The locality constraint is imposed for the first 50K iterations, preventing SLOTLIFTER from segmenting the background as 2 separate objects. Note that our SLOTLIFTER does not require a background-aware Slot-Attention like uORF since our slots are initialized by learnable queries, enabling Slot-Attention to learn to individually segment the background and foreground objects. On ScanNet and DTU MVS, we adopt the coarse-to-fine sampling scheme on ScanNet following previous methods, sampling 64 points along each ray for the coarse sampling and another 64 points for the fine sampling. We found that the coarse-to-fine sampling scheme aids SLOTLIFTER in rendering novel

**Table A.3: Efficiency and performance comparisons on Room-Diverse.** We evaluate all the methods on an NVIDIA RTX 3090 GPU.

Model	PSNR $\uparrow$	LPIPS $\downarrow$	NV-ARI $\uparrow$	ARI $\uparrow$	GPU Memory $\downarrow$	Training Time $\downarrow$
uORF	25.96	0.1729	56.9	65.6	24 GB	6 days
BO-uORF	26.96	0.1515	62.5	72.6	24 GB	6 days
ours(N=256)	29.83	0.1345	76.1	88.7	3.5 GB	10 hours
ours(N=512)	<b>29.84</b>	0.1277	76.2	88.0	6 GB	19 hours
ours(N=1024)	29.80	<b>0.1180</b>	<b>77.5</b>	<b>90.3</b>	12 GB	30 hours

**Table A.4: Ablations on the number of rays with different image sizes.** Increasing the number of rays slightly improves rendering and segmentation quality, while reducing image size slightly decreases both rendering and segmentation quality.

Number of rays	ScanNet (640 $\times$ 480)		Room-Diverse (128 $\times$ 128)				Room-Diverse (64 $\times$ 64)			
	PSNR $\uparrow$	NV-FG-ARI $\uparrow$	PSNR $\uparrow$	LPIPS $\downarrow$	NV-ARI $\uparrow$	ARI $\uparrow$	PSNR $\uparrow$	LPIPS $\downarrow$	NV-ARI $\uparrow$	ARI $\uparrow$
256	27.27	19.8	29.83	0.1345	76.1	88.7	29.55	0.0792	69.2	82.4
512	27.92	<b>32.3</b>	<b>29.84</b>	0.1277	76.2	88.0	<b>29.57</b>	0.0731	69.9	<b>83.2</b>
1024	<b>28.36</b>	31.1	29.80	<b>0.1180</b>	<b>77.5</b>	<b>90.3</b>	<b>29.57</b>	<b>0.0687</b>	<b>70.2</b>	82.4

views with higher quality. The training configuration is summarized in Tab. A.1. Additionally, we found the background occlusion regularization loss from [37, 57] is helpful on the Kitchen-Matte and Kitchen-Shiny datasets for preventing the background slot segmenting foreground objects but it has little effect on the rendering quality. We only use this loss on the Kitchen-Matte and Kitchen-Shiny datasets because we didn’t find it helpful on other datasets.

## A.2 Baselines

**uORF and BO-uORF** The experimental results of uORF [59] on CLEVR-567, Room-Chair, and Room-Diverse are taken from their paper. We trained the BO-uORF model on CLEVR-567, Room-Chair, Room-Diverse, and ScanNet using the official implementation of uORF and BO-QSA. As (BO-)uORF only accepts single source view input, we selected the closest view to the target view as the source view for it. Unfortunately, due to design limitations, such as model architecture, adversarial loss, perceptual loss, etc., we could not train the BO-uORF model at the resolution of 640 $\times$ 480. Therefore, we had to use a resolution of 128 $\times$ 128 following their original settings. We use 8 slots for uORF as same as our method.

**OSRT** We trained OSRT [42] on CLEVR-567, Room-Chair, Room-Diverse, and ScanNet using the implementation recommended by the authors on the project website of OSRT. We observed that OSRT’s performance of scene decomposition is highly sensitive to the batch size used during training, which is also mentioned

in the implementation. Due to the computational limitations, we trained the OSRT for 250K iterations using a batch size of 64 and sampling 2048 rays for each scene with 2 Nvidia A100 GPUs. To train OSRT on the ScanNet dataset, we resized all images to  $128 \times 128$ .

**GNT** We trained GNT [50] on ScanNet using their official implementation. We trained GNT for 250K iterations with their config `gnt_full.txt` in their repository, which uses a learning rate of  $5 \times 10^{-4}$ , samples 2048 rays for each scene, and selects 10 nearby source views to render the target view.

## B Additional Discussions

### B.1 Potential Improvements

Although SLOTLIFTER exhibits superior performance in novel-view synthesis and scene decomposition compared to state-of-the-art 3D object-centric learning methods, its scene decomposition performance still falls short under real-world settings. This is particularly noteworthy considering the recent success of 2D object-centric models on real-world images (see in Tab. A.5). We attribute this undesired effect to the unconstrained point-slot mapping process. As elaborated in Sec. 3.2, the slots are mapped to the 3D points which are later projected to the target view image. With only reconstruction loss, the information in the target image can be backpropagated to both slots and lifted point features. This adds no direct guidance or constraints on slot learning and can easily make the learned slots attend to features that best render the scene instead of decomposing it.

To account for this issue we considered guiding slots to decompose scenes with semantic priors in pre-trained models. Inspired by recent object-centric learning methods DINO-SAUR [43] and VideoSAUR [60] that replace image reconstruction with feature reconstruction, we propose to improve the scene decomposition ca-

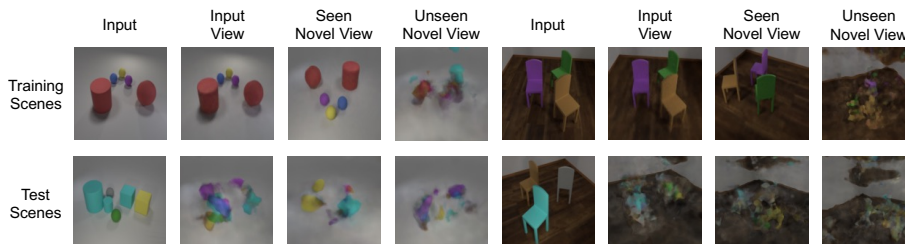
pability of SLOTLIFTER by adding a feature reconstruction loss. Specifically, we first extract DINOv2 [40] features  $\hat{\mathbf{H}}$  for the target view as ground truth. Similar to the color prediction in Eq. (5) we add an MLP to predict a feature grid  $\mathbf{h}$  and render 2D features  $\mathbf{H}$ . Next, we add the feature reconstruction loss over the predicted target-view feature, *i.e.*  $\mathcal{L}_{\text{feat}} = 1 - D(\mathbf{H}, \hat{\mathbf{H}})$ , where  $D$  denotes the cosine similarity. As shown in Tab. A.5, this feature reconstruction loss improves the segmentation performance on ScanNet, but it harms the rendering quality of novel view images, decreasing the PSNR to 25.38. This result reveals the key conflict between the high-level semantic guidance and the low-level appearance guidance which is commonly shared in object-centric models. Adding 3D geometry or temporal constraints (*e.g.*, shape and temporal consistency) that reveal

**Table A.5: Quantitative segmentation results on ScanNet.** FG-ARI is evaluated on the input view(s). ‘‘MV’’ indicates 3D multi-view inputs.

Model	Modality	FG-ARI $\uparrow$	NV-FG-ARI $\uparrow$	PSNR $\uparrow$
Slot-Attention [36]	2D	31.1	-	-
DINO-SAUR [43]	2D	<b>47.6</b>	-	-
OSRT [42]	MV	29.8	29.7	13.34
SLOTLIFTER	MV	32.0	31.1	<b>28.36</b>
SLOTLIFTER w/ $\mathcal{L}_{\text{feat}}$	MV	36.1	<b>35.7</b>	25.38

**Table A.6: Quantitative results of OSRT in synthetic scenes.** We present the best performance of our reimplemented OSRT. The performance of OSRT is hindered by its requirements for large amounts of data and computational demands.

Model	CLEVR-567			Room-Chair			Room-Diverse		
	NV-ARI $\uparrow$	FG-ARI $\uparrow$	PSNR $\uparrow$	NV-ARI $\uparrow$	FG-ARI $\uparrow$	PSNR $\uparrow$	NV-ARI $\uparrow$	FG-ARI $\uparrow$	PSNR $\uparrow$
OSRT <sup>†</sup> [42]	3.1	10.3	20.73	5.4	24.0	20.99	7.4	39.3	24.58
uORF [59]	83.8 $\pm$ 0.3	87.4 $\pm$ 0.8	29.28	74.3 $\pm$ 1.9	88.8 $\pm$ 2.7	29.60	56.9 $\pm$ 0.2	67.8 $\pm$ 1.7	25.96
<b>SLOTLIFTER</b>	<b>87.0<math>\pm</math>2.8</b>	<b>91.3<math>\pm</math>1.8</b>	<b>36.09</b>	<b>89.7<math>\pm</math>0.5</b>	<b>91.9<math>\pm</math>0.3</b>	<b>34.63</b>	<b>77.5<math>\pm</math>0.7</b>	<b>84.3<math>\pm</math>2.9</b>	<b>29.97</b>



**Fig. A.1: Qualitative results of OSRT.** OSRT tends to overfit training scenes and training views, making it difficult to generalize to unseen scenes and unseen views.

objectness can potentially solve this problem and we leave it as an important future work.

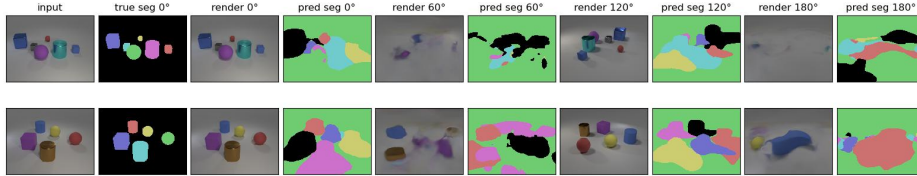
On the other hand, the superior performance on ScanNet and DTU implies better scene encoding in SLOTLIFTER, supporting potential conjectures that these latent slots work similarly to latent feature grids with point features interpolated over them for better novel-view synthesis. This echoes the success of feature-grid-based methods (*e.g.*, Plenoxels [21]) for improving the performance of NeRF.

## B.2 Further Discussions about Previous Methods

**(BO-)uORF** As shown in Tab. 5 and Fig. 4, BO-uORF failed to render novel views and decompose scenes in complex real-world scenes, achieving only a PSNR of 12.72 and a NV-FG-ARI of 0.0. Moreover, to demonstrate that the failure of BO-uORF is not due to lower resolution, we trained our SLOTLIFTER with a resolution of  $128 \times 128$  and achieved a PSNR of 29.31.

**OSRT** We present the quantitative results of OSRT in Tab. A.6 and visualize qualitative results in Fig. A.1. The performance of OSRT is hindered by its requirements for large amounts of data and computational demands, especially on CLEVR-567 which only has 1000 training scenes. We observed that the OSRT tended to overfit training scenes ( Fig. A.1 ), making it difficult to generalize to unseen scenes. We attempted a larger batch size of 256 and trained OSRT for more iterations (750K), but the overfitting issue persisted. We also visualize





**Fig. A.2: Qualitative results of OSRT from the implementation recommended by the authors of OSRT.** OSRT performs well on seen views (*e.g.*,  $0^\circ$ ,  $120^\circ$ ), but has difficulty in unseen views (*e.g.*,  $60^\circ$ ,  $180^\circ$ ).

the results provided by this implementation in Fig. A.2, which demonstrates a similar experimental phenomenon that OSRT performs well on seen views (*e.g.*,  $0^\circ$ ,  $120^\circ$ ), but has difficulty in unseen views (*e.g.*,  $60^\circ$ ,  $180^\circ$ ). Quantitatively, on the CLEVR-567 dataset, OSRT achieved 47+ PSNR on training scenes, but only 20.73 on test scenes. Similarly, on the ScanNet dataset, OSRT achieved 27+ PSNR on training scenes, but only 13.34 on test scenes. These results demonstrate that OSRT may memorize all the training scenes with its powerful transformer encoder-decoder, requiring a lot of data to overcome the overfitting problem. The number of training scenes used in our paper might not be sufficient to train the OSRT(1000 for CLEVR-567 and Room-Chair, 5000 for Room-Diverse, and 100 for ScanNet), leading to the failure case. We attribute this ineffectiveness to its lack of inductive bias for 3D scenes, which is a main distinction between OSRT and our SLOTLIFTER.

## C Limitations and Future Work

**Inference efficiency** While our SLOTLIFTER has significantly improved training efficiency compared to other 3D object-centric models, its inference efficiency is not satisfactory compared with light field methods (*e.g.*, COLF and OSRT). The primary reason for this issue is that NeRF representations require the sampling of a large number of points with expensive computations, most of which are wasted on irrelevant vacant points. Although light field methods, such as OSRT, are very efficient for inference, they lack the use of 3D information and require a lot of data and computation commands to overcome the overfitting problem. Some recent works, such as those based on point clouds [56], surfels [22], and Gaussian Splatting [31, 63], have demonstrated high efficiency for inference and good utilization of 3D information, which could be integrated into future work to improve the inference efficiency.

**Details of Complex Object** As depicted in Fig. A.5 and Fig. A.6, the SLOTLIFTER encounters challenges in accurately rendering and segmenting chair legs from different angles, particularly when dealing with real chairs in Room-Texture. A primary issue contributing to this difficulty lies in ray sampling. NeRF-based techniques typically employ ray sampling during the training process to reduce computational load. For example, in our case, we sample 1024

rays from an image with  $128 \times 128 = 16384$  pixels. Consequently, the majority of rays focus on the background and larger objects, leaving finer details like chair legs with limited attention. While increasing the number of sampled rays could address this issue, it would also escalate the computational demands. The integration of Gaussian Splatting [31] has the potential to assist in balancing computational requirements with rendering quality. Moreover, we have observed that this problem exists in other approaches as well. Nevertheless, it appears to be mitigated in uOCF [37] due to its training with the object prior, which could potentially aid our SLOTLIFTER in addressing this particular challenge.

## D Additional Visualizations

We provide more qualitative results of our SLOTLIFTER in the following pages.



Fig. A.3: Novel view synthesis and unsupervised segmentation on CLEVR-567.



Fig. A.4: Novel view synthesis and unsupervised segmentation on Room-Chair.



Fig. A.5: Novel view synthesis and unsupervised segmentation on Room-Diverse.



Fig. A.6: Novel view synthesis and unsupervised segmentation on Room-Texture.

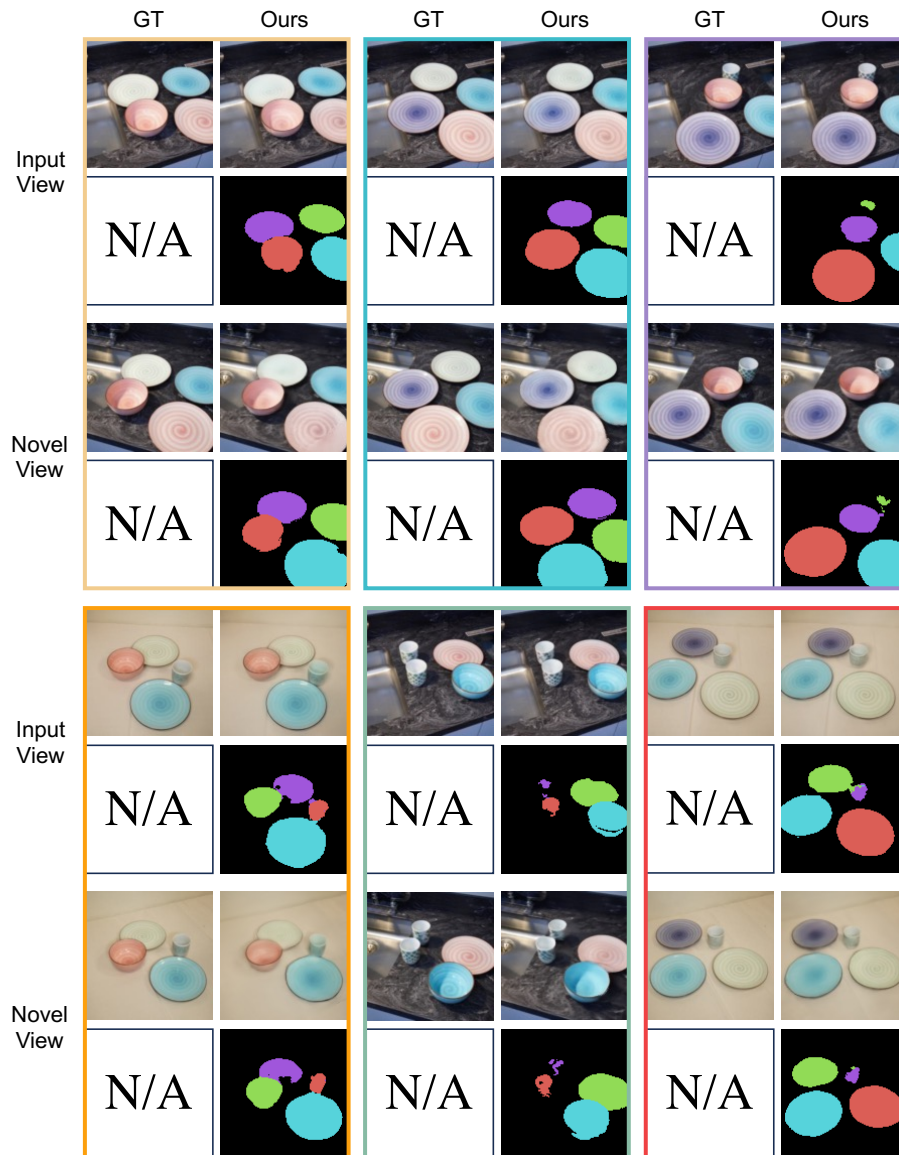


Fig. A.7: Novel view synthesis and unsupervised segmentation on Kitchen-Shiny.

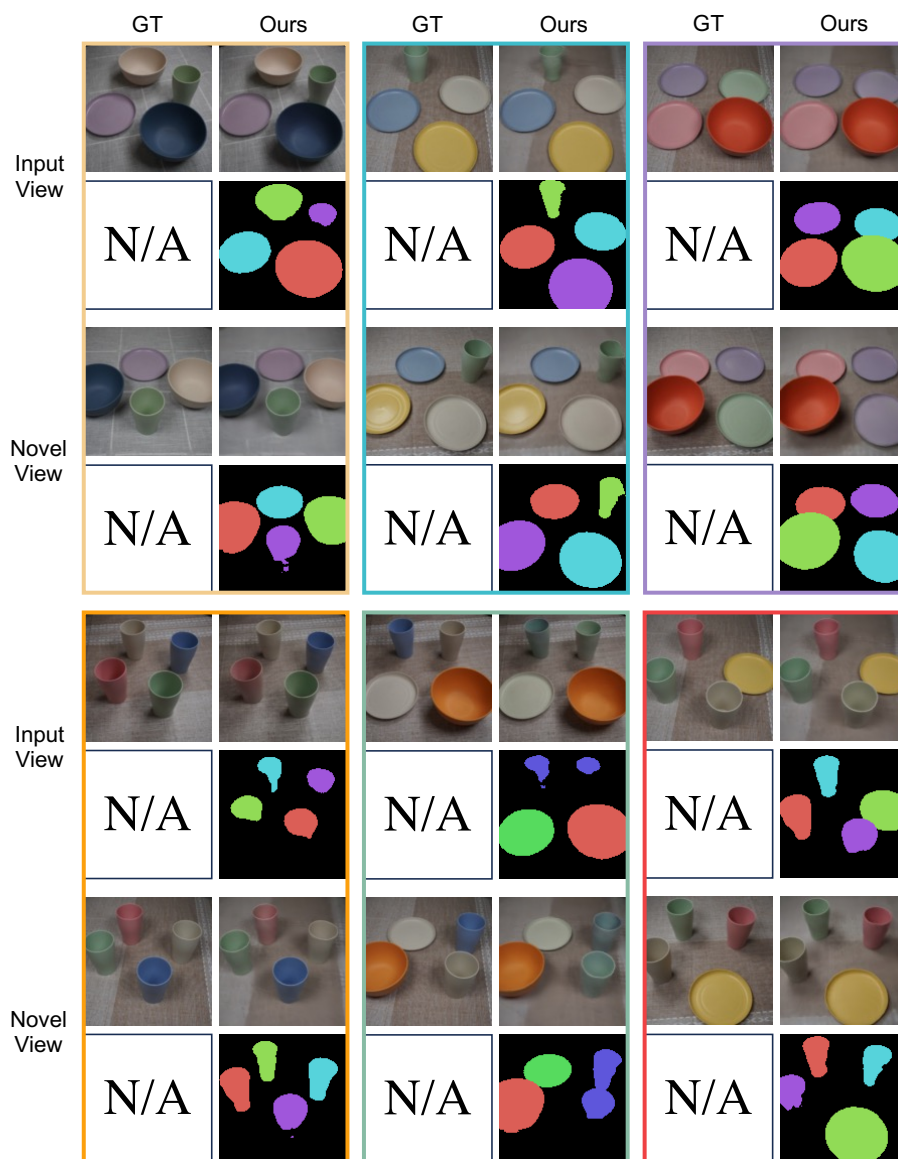


Fig. A.8: Novel view synthesis and unsupervised segmentation on Kitchen-Matte.



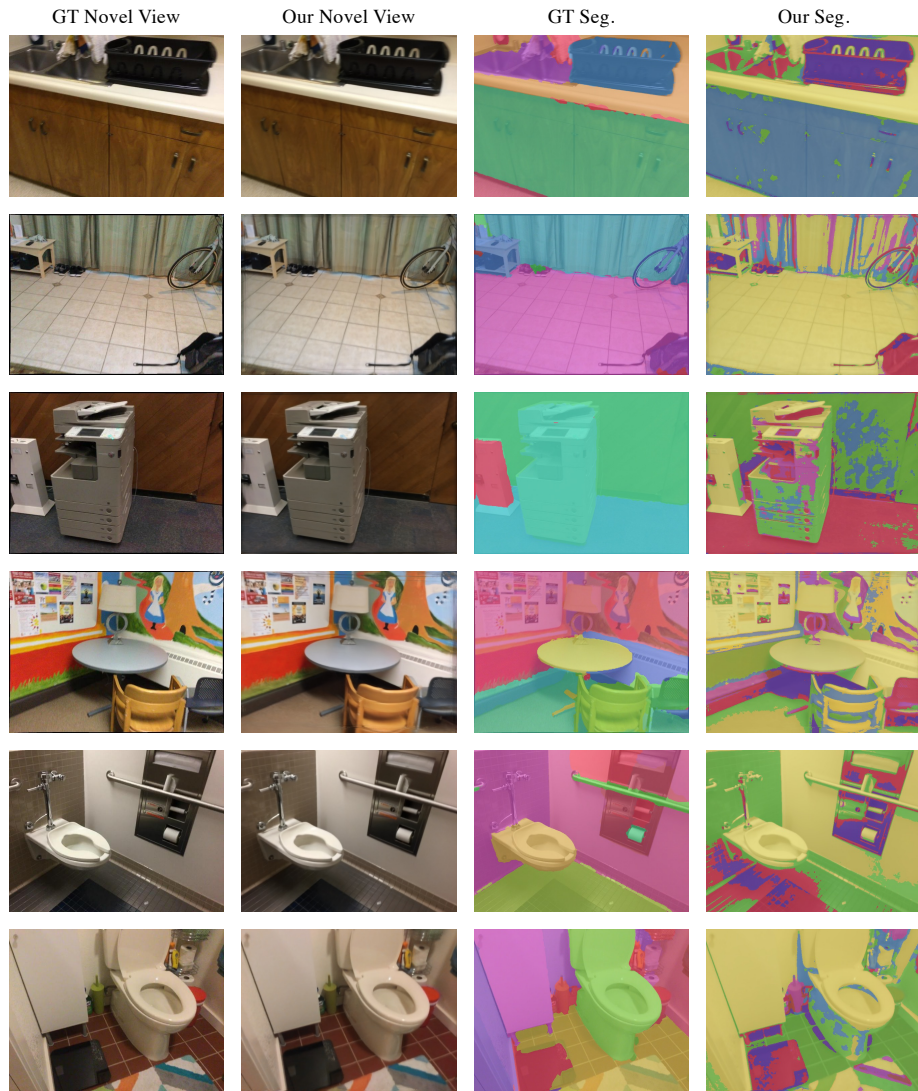


Fig. A.9: Novel view synthesis and unsupervised segmentation on ScanNet.

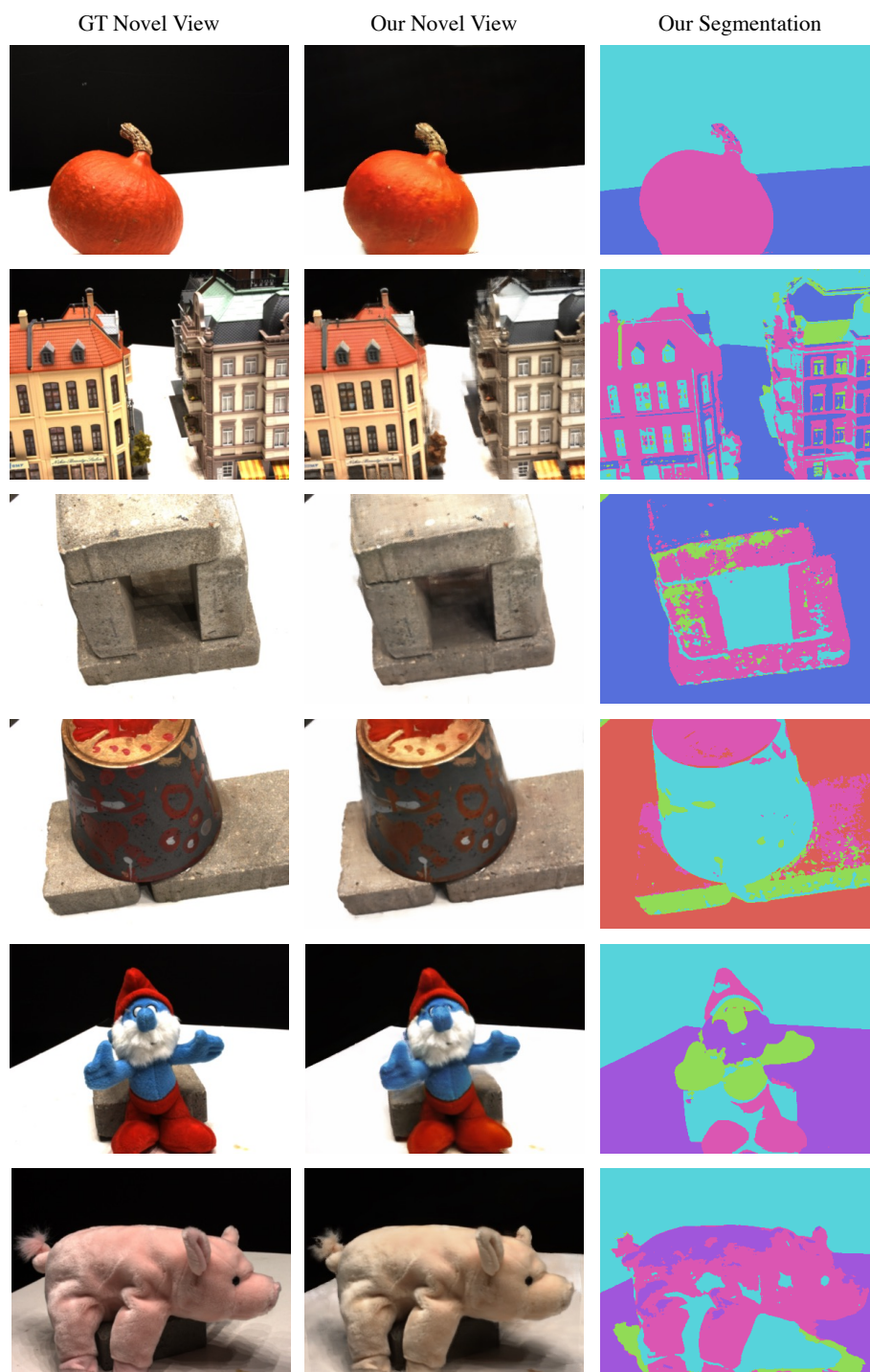


Fig. A.10: Novel view synthesis and unsupervised segmentation on DTU MVS.