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

Yu Liu*†[1,2], Baoxiong Jia*[1], Yixin Chen[1], and Siyuan Huang[1]

[1] State Key Laboratory of General Artificial Intelligence, BIGAI
[2] Department of Automation, Tsinghua University
*Equal contribution †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 [31, 45]. Despite recent advances in visual perception [5, 15, 26, 38], achieving this generalization capability remains an unsolved challenge for existing models [25]. 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 [22, 29, 34, 42]. 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 [29], 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, 40].

To fulfill this goal, various attempts have been made to combine object-centric methods such as Slot-Attention [34] with 3D representations. Among them, multi-view image representations of 3D scenes [44, 46, 56] 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 [41]. 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 [40] 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 [6,12,21,48,49,54,55,58]. 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, 22–24, 33, 34, 59] 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 [34] has fostered many powerful variants [5, 10, 17, 27, 29, 32, 41, 42, 50, 52] 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, 30, 43] 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 [2,37,47] 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, 12,21,48,49,54,55,58]. These methods aim to synthesize novel views based on given images of scenes without per-scene optimization. For instance, PixelNeRF [55] and IBRNet [49] 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 [54] leverages latent point clouds as anchors for radiance fields to improve both efficiency and performance. GNT [48] 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 [40,44,46,56] have attempted to extend Slot-Attention to 3D scenes for scene decomposition and novel-view synthesis. uORF [56], ObSuRF [46], sVORF [39], and uOCF [35] combine Slot-Attention

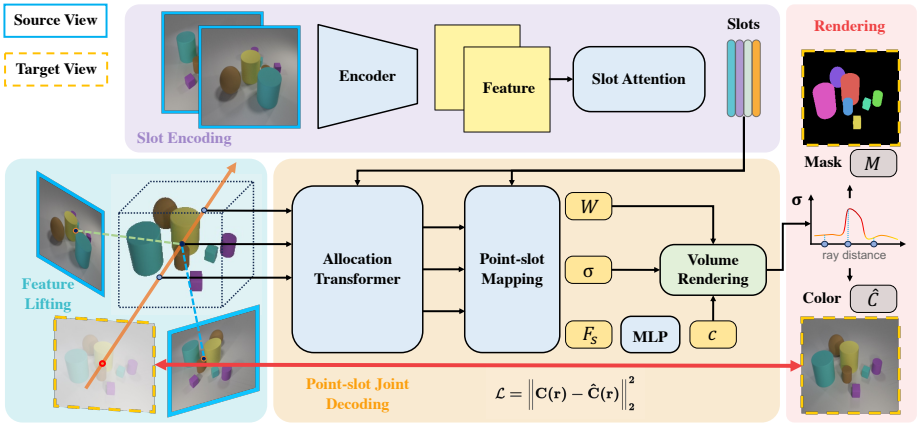

Fig. 1: SlotLifter overview. SlotLifter extracts slots from input view(s) during slot encoding. It then lifts 2D feature maps of input view(s) to initialize 3D point features, which serve as queries in the allocation transformer for point-slot joint decoding. This process yields the point-slot mapping $\boldsymbol{W}_p$, density σ, and the slot-aggregated point feature $\boldsymbol{F}_s$ 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 [37] and use rendering losses as objectives for unsupervised slot learning. Additionally, OSRT [40] and COLF [44] 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 $\boldsymbol{X} \in \mathbb{R}^{N \times D_f}$, Slot-Attention [34] maps them to a set of K output vectors (*i.e.*,

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

$$\begin{aligned} \boldsymbol{A} &= \text{softmax}\left(\frac{k(\boldsymbol{X}) \cdot q(\tilde{\boldsymbol{S}})^T}{\sqrt{D}}\right) \\ \boldsymbol{S} &= \mathcal{U}_\theta(\tilde{\boldsymbol{S}},\ \boldsymbol{W}^T v(\boldsymbol{X})),\ \text{where } \boldsymbol{W}_{i,j} = \frac{\boldsymbol{A}_{i,j}}{\sum_{m=1}^{N} \boldsymbol{A}_{m,j}}. \end{aligned} \tag{1}$$

$q(\cdot),\ k(\cdot),\ v(\cdot)$ are linear projections, $\tilde{\boldsymbol{S}}$ 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.* [29], 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 $\{\boldsymbol{r}\}$ of a camera view, NeRF samples points along each ray and represent 3D scenes with a feature field $\boldsymbol{F}_\Theta : (\boldsymbol{x}, \boldsymbol{d}) \rightarrow (\boldsymbol{c}, \sigma)$ mapping the 3D location $\boldsymbol{x}$ and the view direction $\boldsymbol{d}$ to color $\boldsymbol{c}$ and volume density σ, and then renders the color of each ray via volume rendering [36]:

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

where $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j)$ and δ_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 $\hat{\boldsymbol{C}}(\boldsymbol{r})$ 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 $\boldsymbol{I}_t$, we leverage Slot-Attention to encode scene representations from L source view(s) $\{\boldsymbol{I}_l\}_{l=1}^{L}$ ($L = 1$ for single-view input) and lift 2D features to 3D for approximating the latent feature field $\boldsymbol{F}_\Theta$. We start by extracting 2D feature maps $\{\boldsymbol{F}_l^{\text{2D}} \in \mathbb{R}^{H \times W \times D_\text{f}}\}_{l=1}^{L}$ from each source view. Next, we follow Eq. (1) to obtain object-centric scene features $\boldsymbol{S} = \{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_K\} \in \mathbb{R}^{K \times D_\text{s}}$ 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 $\boldsymbol{I}_t$, we sample points $\boldsymbol{P} \in \mathbb{R}^{N \times 3}$ along

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

$$\boldsymbol{F}_{\text{lift}}(\boldsymbol{p}) = \left[\boldsymbol{F}_1^{\text{2D}}[\pi(\boldsymbol{p})], \cdots, \boldsymbol{F}_L^{\text{2D}}[\pi(\boldsymbol{p})]\right].$$

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

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

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

Point-slot Mappping After scene encoding, given the slots $\boldsymbol{S}$ and the point features $\boldsymbol{F}_p$, 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 $\boldsymbol{s}_i \in \boldsymbol{S}$ contributes to. Specifically, we use a cross-attention-based allocation transformer, leveraging point features $\boldsymbol{F}_p$ as queries and slot representations $\boldsymbol{S}$ 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 $\boldsymbol{s}_\varnothing$ for these vacant points to query from. This process could be summarized as:

$$\boldsymbol{S}' = \{\boldsymbol{s}_\varnothing, \boldsymbol{s}_1, \cdots, \boldsymbol{s}_k\},\ \tilde{\boldsymbol{F}}_p = \text{CrossAttn}(Q = \boldsymbol{F}_p, KV = \boldsymbol{S}').$$

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

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

We use $q(\cdot),\ k(\cdot)$ to denote linear projections, $\boldsymbol{W}_p \in \mathbb{R}^{N \times (K+1)}$ 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 $\boldsymbol{W}_p^i$ for each 3D point $\boldsymbol{p}_i$, thereby predicting its slot assignment for scene decomposition.

Slot-based Density For notation purposes, we use $\boldsymbol{A}_p = q(\tilde{\boldsymbol{F}}_p) \cdot k(\boldsymbol{S}')^T$ throughout the subsequent texts for simplicity. To provide more direct guidance to slots, we use the attention weights $\boldsymbol{A}_p$ from the mapping module to estimate the density value following [56]:

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

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

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

where $T_i = \exp(-\sum_{j=1}^{i-1}\sigma_j\delta_j)$ and δ_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 $\boldsymbol{C}(r)$ and the ground truth colors $\hat{\boldsymbol{C}}(r)$ as our learning objective:

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

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 $\boldsymbol{F}_p$ and slot features $\boldsymbol{S}$ 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 $\boldsymbol{E}_p$ 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 [56]. We further select a more complex variant of Room-Diverse, Room-Texture [35], 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 [35], Kitchen-Matte [35], ScanNet [14], and DTU MVS [28] 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 ± standard deviation) across 3 experiment trials except for sVORF where we report the best performance (†) 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↑	ARI↑	FG-ARI↑	NV-ARI↑	ARI↑	FG-ARI↑	NV-ARI↑	ARI↑	FG-ARI↑
Slot-Attention [34]	-	3.5±0.7	**93.2±1.5**	-	38.4±18.4	40.2±4.5	-	17.4±11.3	43.8±11.7
uORF [56]	83.8±0.3	86.3±0.1	87.4±0.8	74.3±1.9	78.8±2.6	88.8±2.7	56.9±0.2	65.6±1.0	67.8±1.7
BO-uORF [29]	78.4±0.7	87.4±0.5	89.2±0.3	80.9±0.2	82.2±1.0	91.6±2.3	62.5±0.5	72.6±0.2	76.8±0.2
COLF [44]	55.8±0.1	69.0±0.4	92.4±1.7	80.7±0.1	85.6±0.04	89.8±0.1	52.5±0.3	66.5±0.4	64.7±0.7
SlotLifter	**87.0±2.5**	**93.7±1.1**	91.3±1.6	**89.7±0.5**	**92.6±0.3**	**91.9 ±0.3**	**77.5±0.7**	**90.0±0.8**	**84.3±2.7**
sVORF† [39]	81.5	82.7	92.0	87.0	87.8	**92.4**	75.6	78.4	86.6
SlotLifter†	**89.0**	**94.6**	**93.1**	**90.3**	**92.9**	92.1	**78.1**	**90.6**	**86.7**

Metrics We evaluate the quality of novel-view synthesis with three common metrics: LPIPS [57], SSIM [51], and PSNR. In particular, we use $\text{LPIPS}_{\text{alex}}$ for synthetic scenes and $\text{LPIPS}_{\text{vgg}}$ for real-world scenes to be consistent with previous methods. Following [44, 56], 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 [56] 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 $\boldsymbol{F}_p = \text{MLP}(\boldsymbol{F}_{\text{lift}}) + \boldsymbol{E}_p$ as discussed in Sec. 3.2. We train our model using the Lion [7] optimizer with a learning rate of 5×10^{-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 [56], COLF [44], and sVORF [39]. We also report the results of the improved uORF (BO-uORF) introduced by Jia *et al.* [29] 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	Sceee segmentation			Novel view synthesis		
	NV-ARI↑	ARI↑	FG-ARI↑	LPIPS↓	SSIM↑	PSNR↑
uORF [56]	57.8	67.0	9.3	0.254	0.711	24.23
BO-uORF [29]	60.4	69.7	35.4	0.215	0.739	25.26
COLF [44]	1.1	23.5	53.2	0.504	0.670	22.98
uOCF-N [35]	72.2	79.1	58.4	0.138	0.796	28.81
uOCF-P [35]	70.4	78.5	56.3	0.136	0.798	28.85
SlotLifter	**79.3**	**86.0**	**70.7**	**0.131**	**0.858**	**30.68**

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 [56]	0.1288	0.8658	27.16	0.1166	0.8265	28.13	0.2458	0.6688	24.80
uORF [56]	0.0859	0.8971	29.28	0.0821	0.8722	29.60	0.1729	0.7094	25.96
BO-uORF [29]	0.0618	0.9260	30.85	0.0733	0.8938	30.61	0.1515	0.7363	26.96
COLF [44]	0.0608	0.9346	31.81	0.0485	0.8934	30.93	0.1274	0.7308	26.02
sVORF [39]	0.0211	**0.9701**	**37.20**	0.0824	0.8992	33.04	0.1637	0.7825	29.41
SlotLifter	**0.0184**	0.9680	36.09	**0.0410**	**0.9358**	**34.63**	**0.1159**	**0.8479**	**29.97**

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 [56] 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 [35]. 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.

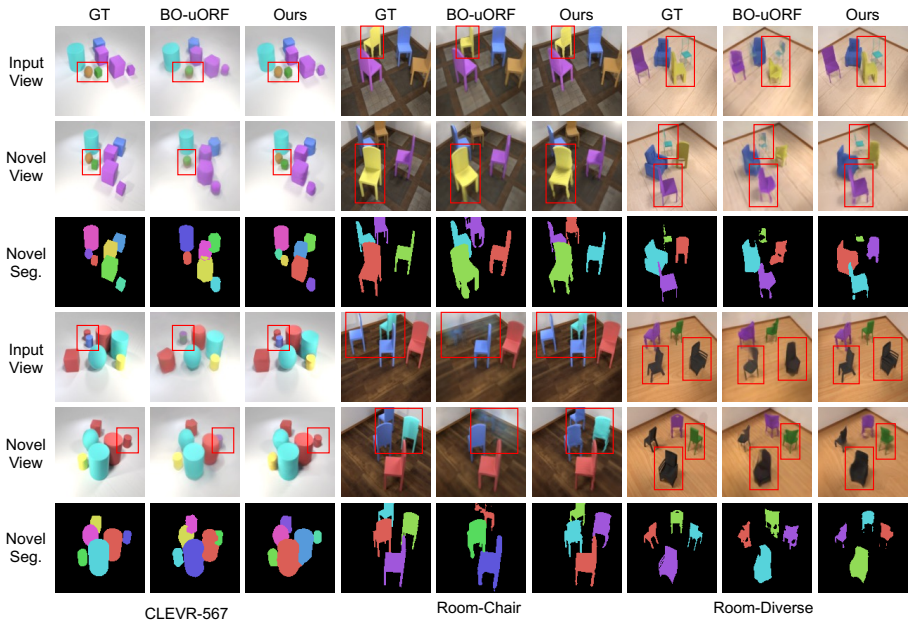

Fig. 2: Qualitative comparison on synthetic scenes. Compared to BO-uORF, SlotLifter renders novel-view images and segmentation masks in much higher quality, especially in detailed object attributes like color and shape (best viewed with zoom-in for the highlighted details).

We also consider ScanNet [14] and DTU [28], which are well-established datasets for evaluating generalizable novel-view synthesis [21,53,58], 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 [53, 58], sample 100 scenes for training, and evaluate our method on the 8 unseen testing scenes introduced. On DTU, we follow the setup of PixelNeRF [55] and NeRFusion [58], 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, 21, 48, 49, 54, 58] 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 [40] 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 [58] on ScanNet and DTU MVS. Additionally, we re-train the recent state-of-the-art method GNT [48] for generalizable

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

Method	Kitchen-Shiny			Kitchen-Matte		
	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑
uORF [56]	0.336	0.602	19.23	0.092	0.808	26.07
BO-uORF [29]	0.318	0.639	19.78	0.067	0.832	27.36
COLF [44]	0.397	0.561	18.30	0.236	0.643	20.68
uOCF-N [35]	0.055	0.842	27.87	0.055	0.841	28.25
uOCF-P [35]	0.049	0.862	28.58	0.043	0.867	29.40
SLOTLIFTER	**0.035**	**0.928**	**32.02**	**0.030**	**0.939**	**32.92**

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

Method	PSNR↑	SSIM↑	LPIPS↓	NV-FG-ARI ↑
IBRNet [49]	21.19	0.786	0.358	-
NeRFusion [58]	22.99	0.838	0.335	-
PointNeRF [54]	20.47	0.642	0.544	-
SurfelNeRF [21]	23.82	0.845	0.327	-
GNT† [48]	27.76	0.8791	0.2197	-
BO-uORF† [56]	12.72	0.3393	0.6975	0.0
OSRT† [40]	13.34	0.2746	0.6337	29.7
SLOTLIFTER	**28.36**	**0.9200**	**0.1891**	**31.1**

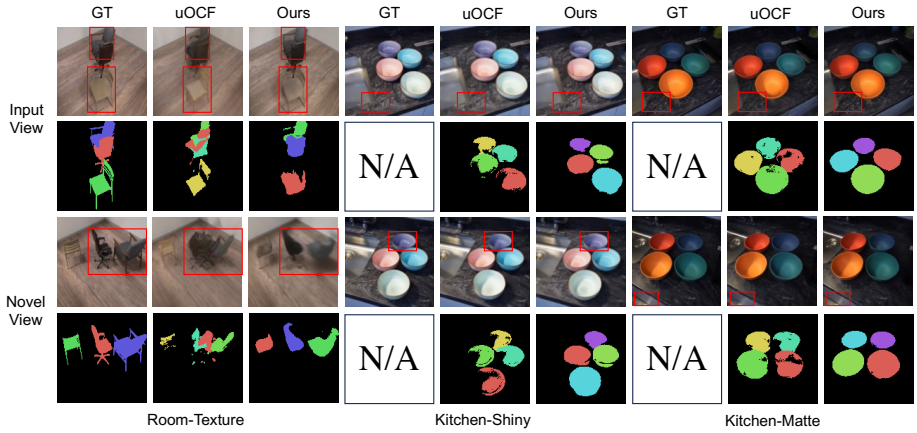

Fig. 3: Qualitative comparison on Room-Texture, Kitchen-Shiny, and Kitchen-Matte. Compared to the SOTA method uOCF, SLOTLIFTER renders novel-view images and segmentation masks in higher quality, offering more complete segmentation and more detailed textures (best viewed with zoom-in for the highlighted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

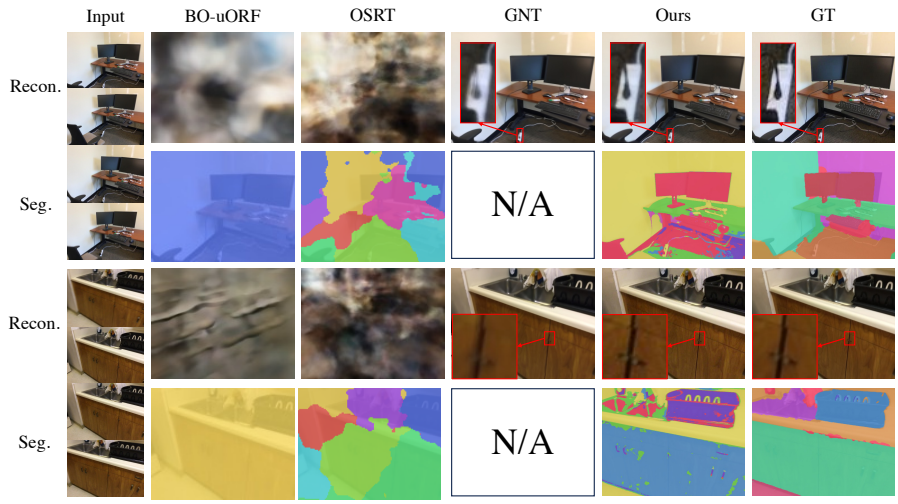

Fig. 4: Qualitative results on ScanNet. Our SlotLifter achieves the best performance for novel-view rendering, even surpassing the recent state-of-the-art model GNT, while BO-uORF and OSRT struggle to render novel-view images on ScanNet.

Table 6: Quantitative comparison on DTU.

Method	PSNR↑	SSIM↑	LPIPS↓
PixelNeRF [55]	19.31	0.789	0.382
IBRNet [49]	26.04	0.917	0.190
MVSNeRF [6]	26.63	**0.931**	0.168
NeRFusion [58]	26.19	0.922	0.177
SlotLifter	**26.75**	0.896	**0.157**

Table 7: Sensitivity of random masking ratio scheduling.

Decay Steps	PSNR↑	NV-ARI↑	ARI↑	FG-ARI ↑
0	29.89	74.4	85.8	43.6
10000	**30.01**	74.9	86.9	42.1
30000	29.80	**77.5**	**90.3**	84.8
60000	29.53	77.3	90.1	**85.7**
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.

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 Surfel-NeRF 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	**0.1169**	**0.8470**	**29.89**	74.4	85.8	43.6	**0.1861**	**0.9208**	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	**77.5**	**90.3**	**84.8**	0.1891	0.9200	**28.36**	**31.1**

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.*, $\boldsymbol{F}_p = \boldsymbol{E}_p$. 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

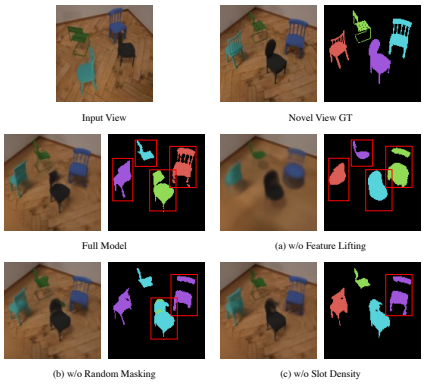

Fig. 5: Visualization of model ablation analysis. (a) Without feature lifting, SlotLifter renders blurred images and imprecise segmentation masks. (b) Without random masking, SlotLifter cannot segment objects correctly. (c) Using slot-based density helps SlotLifter learn more accurate segmentation.

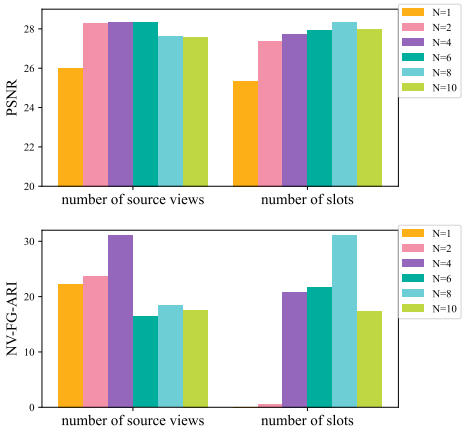

Fig. 6: Ablative studies over the number of source views and slots. We set the number of slots to 8 for different numbers of source views and set the number of source views to 4 for different numbers of slots.

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. Gao, Y., Cao, Y.P., Shan, Y.: Surfelnerf: Neural surfel radiance fields for online photorealistic reconstruction of indoor scenes. In: Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
22. 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)
23. 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)
24. Greff, K., Van Steenkiste, S., Schmidhuber, J.: Neural expectation maximization. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS) (2017)
25. Greff, K., Van Steenkiste, S., Schmidhuber, J.: On the binding problem in artificial neural networks. arXiv preprint arXiv:2012.05208 (2020)
26. 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)
27. 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)
28. 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)
29. Jia, B., Liu, Y., Huang, S.: Improving object-centric learning with query optimization. In: Proceedings of International Conference on Learning Representations (ICLR) (2023)

30. Kipf, T., Elsayed, G.F., Mahendran, A., Stone, A., Sabour, S., Heigold, G., Jonschkowski, R., Dosovitskiy, A., Greff, K.: Conditional object-centric learning from video. In: Proceedings of International Conference on Learning Representations (ICLR) (2022)
31. 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)
32. 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)
33. 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)
34. 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)
35. Luo, R., Yu, H.X., Wu, J.: Unsupervised discovery of object-centric neural fields. arXiv preprint arXiv:2402.07376 (2024)
36. Max, N.: Optical models for direct volume rendering. IEEE Transactions on Visualization and Computer Graphics **1**(2), 99–108 (1995)
37. 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)
38. 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)
39. QI, D., Yang, T., Zhang, X.: Slot-guided volumetric object radiance fields. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS) (2023)
40. 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)
41. 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)
42. Singh, G., Deng, F., Ahn, S.: Illiterate dall-e learns to compose. In: Proceedings of International Conference on Learning Representations (ICLR) (2021)
43. 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)
44. 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)
45. Spelke, E.S., Kinzler, K.D.: Core knowledge. Developmental science **10**(1), 89–96 (2007)
46. Stelzner, K., Kersting, K., Kosiorek, A.R.: Decomposing 3d scenes into objects via unsupervised volume segmentation. arXiv preprint arXiv:2104.01148 (2021)

47. 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)
48. 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)
49. 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)
50. 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)
51. 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)
52. Wang, Z., Shou, M.Z., Zhang, M.: Object-centric learning with cyclic walks between parts and whole. arXiv preprint arXiv:2302.08023 (2023)
53. 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)
54. 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)
55. 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)
56. Yu, H.X., Guibas, L.J., Wu, J.: Unsupervised discovery of object radiance fields. In: Proceedings of International Conference on Learning Representations (ICLR) (2022)
57. 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)
58. 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)
59. 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)