UNSUPERVISED DISCOVERY OF OBJECT-CENTRIC NEURAL FIELDS

Anonymous authors
Paper under double-blind review

ABSTRACT

We study inferring 3D object-centric scene representations from a single image. While recent methods have shown potential in unsupervised 3D object discovery from simple synthetic images, they fail to generalize to real-world scenes with visually rich and diverse objects. This limitation stems from their object representations, which entangle objects' intrinsic attributes like shape and appearance with extrinsic, viewer-centric properties such as their 3D location. To address this fundamental bottleneck, we propose unsupervised discovery of Object-Centric neural Fields (uOCF). uOCF focuses on learning the intrinsics of objects and models the extrinsics separately. Our approach significantly improves systematic generalization, thus enabling unsupervised learning of high-fidelity object-centric scene representations from sparse real-world images. To evaluate our approach, we collect three new datasets including two real kitchen environments. Extensive experiments show that uOCF enables unsupervised discovery of visually rich objects from a single real image, allowing applications such as 3D object segmentation and scene manipulation. Impressively, uOCF even demonstrates zero-shot generalizability to unseen, more difficult objects. We attach an *overview video* in our supplement.

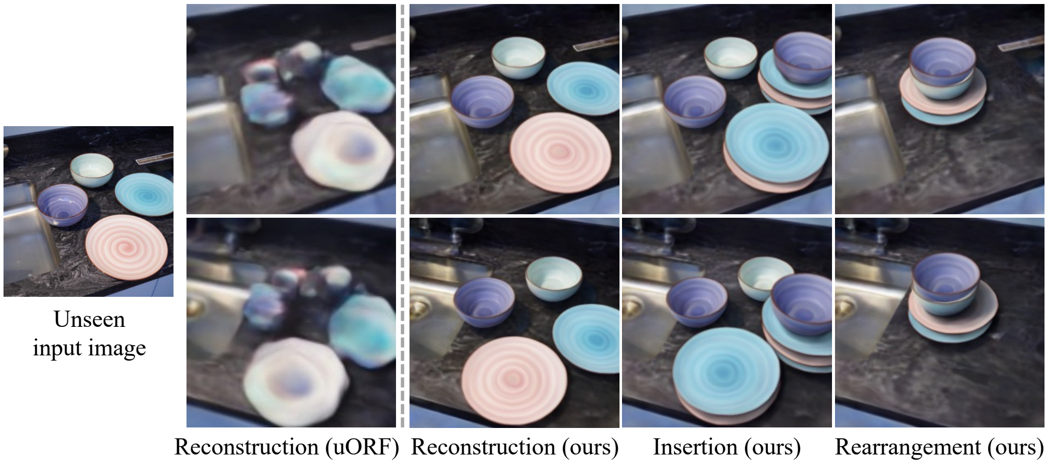
1 INTRODUCTION

Creating factorized, object-centric 3D scene representations is a fundamental ability in human vision and a long-standing topic of interest in computer vision and machine learning. Some recent works have explored unsupervised learning of 3D factorized scene representations from images only (Stelzner et al., 2021; Yu et al., 2022; Smith et al., 2023; Jia et al., 2023). These methods have delivered promising results in 3D object discovery and reconstruction from a simple synthetic image.

However, difficulties arise when attempting to generalize these approaches to complex real-world scenes. The bottleneck lies in their object representations: they represent each object in the viewer's frame, entangling intrinsic object attributes such as shape and appearance with extrinsic properties such as the object's location. This paradigm means a slight shift in the object's location or a subtle camera adjustment can significantly alter its latent representation. Intuitively, an object's intrinsic attributes should remain consistent irrespective of its location, yet this invariance is ignored in existing 3D object-centric learning models. As demonstrated by convolutional networks (Zhang, 2019), accounting for this invariance is crucial for generalization (Chattopadhyay et al., 2020; Deng et al., 2022). Thus, the entanglement of object intrinsics and extrinsics substantially reduces the sample efficiency and hinders systematic generalization.

In this work, we propose unsupervised discovery of Object-Centric neural Fields (uOCF). uOCF learns to infer 3D object-centric scene representations from a single image. Unlike existing methods, uOCF focuses on learning the intrinsics of objects and models the extrinsics separately. As shown in Figure 1, this design enables uOCF to generalize to real-world scenes using our object-centric prior learning and object-centric sampling techniques. We train uOCF on sparse multi-view images without annotations. During inference, uOCF generates object-centric neural radiance fields with their 3D locations explicitly estimated, as well as a background radiance field.

To evaluate our approach, we introduce new challenging datasets for 3D object discovery, including two real kitchen datasets and a synthetic room dataset. The two real datasets feature varied kitchen backgrounds and objects from multiple categories. The synthetic room dataset features chairs with diverse, realistic shapes and textures. Across all these datasets, uOCF offers high-fidelity discovery



Unseen input image

Reconstruction (uORF) Reconstruction (ours) Insertion (ours) Rearrangement (ours)

Figure 1: We propose unsupervised discovery of Object-Centric neural Fields (uOCF), which infers factorized 3D scene representations from an unseen real image, thus enabling scene reconstruction and manipulation from novel views. We compare uOCF to the state-of-the-art method uORF (Yu et al., 2022).

of object-centric neural fields, allowing applications such as unsupervised 3D object segmentation and scene manipulation from a real image. Remarkably, uOCF's generalizability even facilitates *zero-shot* 3D object discovery on images with unseen objects.

In summary, our contributions are threefold: First, we highlight the previously overlooked role of object-centric modeling in unsupervised 3D object discovery and introduce unsupervised discovery of Object-Centric neural Fields (uOCF), a novel approach that effectively disentangles object intrinsic properties from their 3D locations. Second, to achieve such disentanglement and perform high-fidelity reconstruction, we propose object-centric prior learning and object-centric sampling techniques. Finally, we collect three challenging datasets, namely Room-Texture, Kitchen-Easy, and Kitchen-Hard. Our uOCF demonstrates its exceptional performance compared with existing methods on these datasets, even unlocking zero-shot single-image object discovery on unseen objects.

2 Related Works

Unsupervised Object Discovery. Prior to the rise of deep learning, traditional methods for object discovery (often referred to as co-segmentation) primarily aimed at locating visually similar objects across a collection of images (Sivic et al., 2005; Russell et al., 2006), where objects are defined as visual words or clusters of patches (Grauman & Darrell, 2006; Joulin et al., 2010). This clustering concept was later incorporated into deep learning techniques for improved grouping results (Li et al., 2019; Vo et al., 2020). The incorporation of deep probabilistic inference propelled the field towards factorized scene representation learning (Eslami et al., 2016). These methods decompose a visual scene into several components, where objects are often modeled as latent codes that can be decoded into image patches (Kosiorek et al., 2018; Crawford & Pineau, 2019; Jiang et al., 2020; Lin et al., 2020), scene mixtures (Greff et al., 2016; 2017; 2019; Burgess et al., 2019; Engelcke et al., 2019; Locatello et al., 2020; Biza et al., 2023; Didolkar et al., 2023), or layers (Monnier et al., 2021). Despite their efficacy in decomposing visual scenes, they do not model the objects' 3D nature.

To model the 3D nature of scenes and objects, several methods have tried to learn 3D-aware representations from multi-view images of a single-scene (Liang et al., 2022) or of large datasets for generalization (Eslami et al., 2018; Chen et al., 2020; Sajjadi et al., 2022). The latest research emphasizes the single-image inference of object-centric factorized scene representations (Stelzner et al., 2021; Yu et al., 2022; Smith et al., 2023). Notably, Yu et al. (2022) propose unsupervised discovery of object radiance fields (uORF) from a single image. Later works improve the efficiency (Smith et al., 2023) and segmentation (Jia et al., 2023). However, their object representations suffer from entangled extrinsic properties and limited generalizability to complex real scenes. Contrasting these, our approach allows high-quality inference on real images by introducing object-centric modeling.

Scene Decomposition. Decomposing visual scenes on an object-by-object basis and estimating their semantic/geometric attributes has been explored in several recent works (Wu et al., 2017; Yao et al., 2018; Kundu et al., 2018; Ost et al., 2021). Some approaches, like AutoRF (Müller et al., 2022),

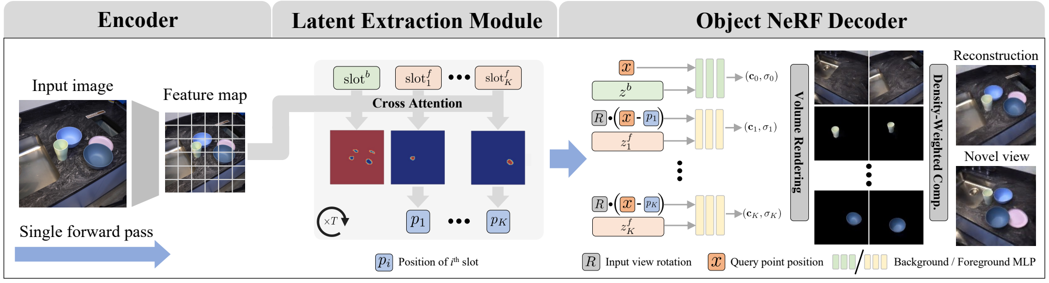


Figure 2: With a single forward pass, uOCF processes a single image input to infer a set of object-centric radiance fields along with their 3D locations, as well as a background radiance field. uOCF is trained on sparse multi-view images of a collection of scenes and uses a single image as input during inference.

successfully reconstruct specific objects (*e.g.*, cars) from annotated images. Others decompose visual scenes into the background and individual objects represented by neural fields (Yang et al., 2021; Wu et al., 2022). Though our study is relevant to these works, we emphasize unsupervised learning, negating the requirement for object annotations.

Neural Fields. Neural fields have revolutionized 3D scene modeling. Early works have shown promising geometry representations (Sitzmann et al., 2019; Park et al., 2019). The seminal work by Mildenhall et al. (2020) on neural radiance fields has opened up a burst of research on neural fields. We refer readers to recent survey papers (Tewari et al., 2020; Xie et al., 2022) for a comprehensive overview. Particularly, compositional generative neural fields such as GIRAFFE (Niemeyer & Geiger, 2021) and others (Nguyen-Phuoc et al., 2020; Wang et al., 2023b) also allow learning object representations from image collections. Yet, they target unconditional generation and cannot tackle inference (Yu et al., 2022).

3 Approach

Given a single input image, our goal is to infer object-centric neural radiance fields (*i.e.*, each discovered object is represented in the local object coordinate frame rather than the world frame or the viewer frame) and the positions of the objects in the 3D scene. The object-centric design not only promotes generalizability due to representation invariance but also facilitates the incorporation of two novel techniques, namely object-centric prior learning and object-centric sampling. These techniques further enhance generalization to complex scenes and improve reconstruction fidelity. In the following, we provide an overview of our approach and then introduce the technical details.

3.1 Overview

As shown in Figure 2, our model consists of an encoder, a latent extraction module, and a decoder.

Encoder. From an input image $\mathbf{I}$, the encoder extracts two sets of feature maps $f_g \in \mathbb{R}^{(H \cdot W) \cdot C_1}$ and $f_l \in \mathbb{R}^{(H \cdot W) \cdot C_2}$ using two sub-modules, E_1 and E_2, where H and W are the height and width of the feature maps, and C_1 and C_2 represent the number of channels. Specifically, E_1 extracts global features using a frozen DINOv2-ViT (Oquab et al., 2023) followed by two convolutional layers, while E_2 is a shallow U-Net that emphasizes local features, *i.e.*, $f_g = E_1(\mathbf{I})$ and $f_l = E_2(\mathbf{I})$.

Latent Extraction Module. The latent extraction module infers the latent representations and positions of objects in the underlying 3D scene from the obtained feature map. We assume the scene is composed of a background environment and no more than K foreground objects. Therefore, the output consists of a background latent $\mathbf{z}_0$ and a set of paired foreground latents and positions $\{\mathbf{z}_i, p_i^{\text{wd}}\}_{i=1}^K$, where $p_i^{\text{wd}} \in \mathbb{R}^3$ denotes a position in the world frame.

Decoder. Our decoder employs the conditional NeRF formulation $g(\mathbf{x}|\mathbf{z})$, comprising two MLPs: g_b for the background environment and g_f for foreground objects. We render each foreground object within its object-centric frame and then compose them into a holistic scene.

3.2 Object-Centric Modeling

Our latent inference module is built upon the Slot Attention (SA) (Locatello et al., 2020) framework enhanced with background awareness (Yu et al., 2022) (B-SA). B-SA features an iterative competition between a set of slot latent ($\text{slot}^b \in \mathbb{R}^{1 \times D_s}$, $\text{slots}^f \in \mathbb{R}^{K \times D_s}$) over the input queries (inputs $\in$

$\mathbb{R}^{N \times D_i}, N = H \cdot W$). In each training step, B-SA begins by sampling the background and foreground slot latents from two Gaussian distributions, each with a learnable mean and variance:

$$\text{slot}^b \sim \mathcal{N}(\mu^b, \text{diag}(\sigma^b)), \quad \text{slot}_i^f \sim \mathcal{N}(\mu^f, \text{diag}(\sigma^f)), \text{ and } \text{slots}^f = \begin{bmatrix} \text{slot}_1^f \\ \cdots \\ \text{slot}_K^f \end{bmatrix}. \tag{1}$$

The competition between slots is then represented via the attention mechanism:

$$\text{attn}_{i,j} = \frac{\exp(M_{i,j})}{\sum_k \exp(M_{i,k})}, \quad \text{where } M = \frac{1}{\sqrt{D^s}} \mathcal{K}(\text{inputs}) \cdot \begin{bmatrix} \mathcal{Q}^b(\text{slot}^b) \\ \mathcal{Q}^f(\text{slots}^f) \end{bmatrix}^T \in \mathbb{R}^{N \times (K+1)}, \tag{2}$$

and with $\mathcal{K}$, $\mathcal{Q}^b$, $\mathcal{Q}^f$ being learnable linear functions. Based on the attention weights, the update signals for the slots are set as a weighted mean of the input tokens, *i.e.*,

$$\text{updates}^b = (W[:, 0])^T \cdot \mathcal{V}(\text{inputs}) \in \mathbb{R}^{1 \times D_s}, \; \text{updates}^f = (W[:, 1{:}])^T \cdot \mathcal{V}(\text{inputs}) \in \mathbb{R}^{K \times D_s}, \tag{3}$$

where $W_{i,j} = \frac{\text{attn}_{i,j}}{\sum_l \text{attn}_{l,j}}$ is the attention map normalized over the spatial dimension, and $\mathcal{V}$ is a linear function. Finally, slots are updated via the Gated Recurrent Unit (GRU) (Cho et al., 2014): $\text{slot}^b \leftarrow \text{GRU}(\text{slot}^b, \text{updates}^b)$ and $\text{slots}^f \leftarrow \text{GRU}(\text{slots}^f, \text{updates}^f)$. This procedure is repeated for T iterations, finally delivering the background latent $\mathbf{z}_0$ and foreground latents $\{\mathbf{z}_i\}_{i=1}^K$.

Object-Centric Latent Inference. We assign a normalized image coordinate $p_i^{\text{img}} \in [-1, 1]^2$ with learnable initialization to each slot, then iteratively update them to obtain the final slot positions. To better exploit the position information, we leverage the slot-specific positional encoding as proposed by Biza et al. (2023), which is defined as $\mathcal{E}_i^{\text{pos}} := \text{concat}([\mathcal{E}^{\text{abs}} - p_i^{\text{img}}, p_i^{\text{img}} - \mathcal{E}^{\text{abs}}])$, with $\mathcal{E}^{\text{abs}} \in \mathbb{R}^{N \times 2}$ being the normalized 2D grid. Then, we re-write M in Eq. (2) as:

$$M = \frac{1}{\sqrt{D^s}} \begin{bmatrix} \mathcal{Q}^b(\text{slot}^b) \cdot \mathcal{K}^b(\text{inputs} + g(\mathcal{E}^{\text{abs}}))^T \\ \mathcal{Q}^f(\text{slot}_1^f) \cdot \mathcal{K}^f(\text{inputs} + g(\mathcal{E}_1^{\text{pos}}))^T \\ \cdots \\ \mathcal{Q}^f(\text{slot}_K^f) \cdot \mathcal{K}^f(\text{inputs} + g(\mathcal{E}_K^{\text{pos}}))^T \end{bmatrix}^T \in \mathbb{R}^{N \times (K+1)}, \tag{4}$$

where $\mathcal{K}^b, \mathcal{K}^f$ are linear functions and g is an MLP. In this formulation, queries (slot^b and slots^f) are attended by slot-specific keys and values with position information included.

We compute the update signals following Eq. (3). Then, the slot's position on the image is formulated as the weighted mean of attention over the $\mathcal{E}^{\text{abs}}$, complemented by a bias term to handle occlusion:

$$p_i^{\text{img}} = \mathcal{E}^{\text{abs}} \cdot \text{attn}[:, i] + \tanh(h(\text{attn}[:, i])) * \alpha, \quad i = 1, \cdots, K \tag{5}$$

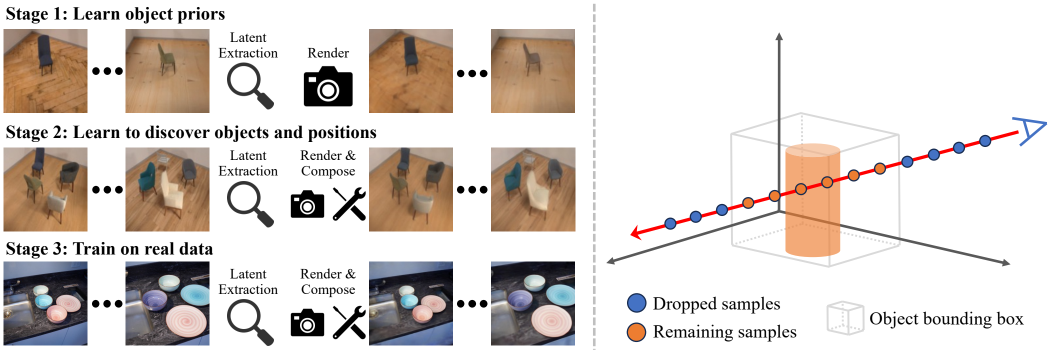
where $h : \mathbb{R}^N \to \mathbb{R}^2$ is a linear function, and $\alpha = 0.2$ serves as a scaling hyperparameter.

In each iteration, we first compute the positional encoding and the attention map, then update the slot latent and location. Finally, we project the slot location from the input image (p_i^{img}) onto the ground plane of the world coordinates to obtain p_i^{wd}. We provide more details in Appendix C.1.

Compositional Neural Rendering. With the positions of foreground slots, we put objects in the object-centric rather than the viewer-centric frame, thereby obtaining the object-centric neural fields. Technically, for each query point $\mathbf{x}$ in world coordinates, we translate it to the i^{th} slot's object-centric frame by first subtracting the slot position and then rotating it by the input camera view's rotation matrix R, *i.e.*, $\mathbf{x}^i = R \cdot (\mathbf{x} - p_i^{\text{wd}})$. We then retrieve the color and density of $\mathbf{x}$ in the foreground radiance fields as $(\mathbf{c}_i, \sigma_i) = g^f(\mathbf{x}^i | \mathbf{z}_i)$ and in the background radiance field as $(\mathbf{c}_0, \sigma_0) = g^b(\mathbf{x} | \mathbf{z}_0)$. These values are aggregated into the scene's composite density and color $(\overline{\mathbf{c}}, \overline{\sigma})$ using density-weighted means, defined as:

$$\overline{\sigma} = \sum_{i \geq 0} \omega_i \sigma_i, \quad \overline{\mathbf{c}} = \sum_{i \geq 0} \omega_i \mathbf{c}_i, \quad \text{where} \quad \omega_i = \frac{\sigma_i}{\sum_{j \geq 0} \sigma_j}. \tag{6}$$

Finally, the color $C(\mathbf{r})$ of a camera ray $\mathbf{r}(t) = \mathbf{o} + \mathbf{r}(t)$ is determined using the discretized volume rendering technique introduced by Mildenhall et al. (2020). Our pipeline is trivially differentiable, allowing backpropagation through all parameters simultaneously.



(a) Object-centric prior learning (b) Object-centric sampling

Figure 3: (a): The object-centric prior learning. We first train our model on single-object synthetic scenes to learn object intrinsic priors, then on multi-object synthetic scenes to learn to discover objects and their positions, and finally train it on real scenes. (b): The object-centric sampling. We drop out the samples distant from the object position for efficient sampling.

Discussion on Extrinsics Disentanglement. An object's canonical orientation is ambiguous without assuming its category (Wang et al., 2019). Thus, we choose not to disentangle objects' orientation since we target category-agnostic object discovery. Moreover, we empirically find that our model effectively learns objects' scale and orientation after disentangling their position (detailed below and in Appendix B). Therefore, we choose only to disentangle the position for generality.

Proof of Concept. We conduct a toy experiment to validate uOCF's invariant object representations. Further, we observe that uOCF has learned objects' scale and orientation, as demonstrated by interpolating the representations of two identical objects with different orientations and scales to obtain transitional results. Please refer to Appendix B for visualization and detailed analysis.

3.3 Object-centric prior learning and sampling

This section introduces object-centric prior learning to enhance generalizability to complex real scenes and object-centric sampling to improve reconstruction fidelity. Both techniques are enabled by object-centric modeling. Please refer to Appendix E.1 for ablation studies and detailed discussions.

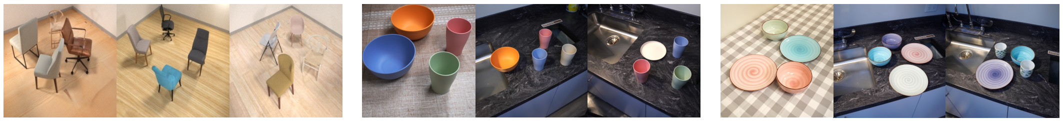
Object-Centric Prior Learning. The lack of inherent object priors complicates unsupervised object discovery in complex real-world scenes. To address this, we have developed a three-stage training pipeline that acquires object priors from synthetic images, as illustrated in Figure 3(a). Initially, the model learns the fundamentals of object-centric NeRFs, focusing on principles such as physical coherence (Spelke, 1990), which is vital for unsupervised segmentation (Chen et al., 2022). The next stage enhances the model's ability to predict object positions and segregate them into individual slots, preparing it for complex real-world scenarios. The final stage involves training on real-world scenes.

Notably, during the first two stages, the model is trained on synthetic datasets (which can be easily synthesized) to learn general object priors **agnostic to** the real-world dataset's object categories. For example, as detailed in Section 4, object priors learned from synthetic chairs effectively apply to dinnerware in real kitchen scenes. Further supporting this approach, in Appendix E.1, we demonstrate that using the simplistic CLEVR dataset (colored primitives like cubes and spheres) for object prior learning still yields satisfactory results.

Object-Centric Sampling. To further improve the reconstruction quality, we leverage the object-centric frame to concentrate the sampled points in proximity to objects, as illustrated in Figure 3(b). Specifically, after a few training epochs, we start dropping distant samples from the predicted object positions when the model learns to distinguish the foreground objects and predict their positions. This straightforward approach enables us to quadruple the number of samples with the same amount of computation, leading to significantly improved background reconstruction quality.

3.4 Model Training

Input Features. Recall that our encoder produces two sets of feature maps: f_g and f_l. Given that f_g is extracted by a pre-trained ViT and contains more global information, we employ f_g as the "inputs" for the latent extraction module. Then, we concatenate the output of the latent extraction module with the attention-weighted mean of f_l to derive the final slot latents $\{\mathbf{z}_i\}_{i=0}^{K}$. This design balances the global and local features while maximizing the utility of pre-trained models.



(a) Room-Texture (b) Kitchen-Easy (c) Kitchen-Hard

Figure 4: Samples from our datasets.

Table 1: Scene segmentation and novel view synthesis result on Room-Texture. Comparison methods are uORF (Yu et al., 2022), QBO (Jia et al., 2023), and COLF (Smith et al., 2023).

Method	Scene segmentation			Novel view synthesis		
	ARI↑	FG-ARI↑	NV-ARI↑	PSNR↑	SSIM↑	LPIPS↓
uORF	0.649	0.108	0.587	24.37	0.688	0.251
QBO	0.674	0.399	0.585	25.07	0.713	0.222
COLF	0.202	0.373	0.024	22.17	0.630	0.555
uOCF (ours)	**0.802**	**0.785**	**0.747**	**28.96**	**0.803**	**0.121**

Loss Functions. In all training stages, we train our model across a collection of scenes, each with sparse multi-view images. Specifically, the model receives an image as input, infers the objects' latent representations and positions, renders multiple views from the reference poses, and compares them to the reference images to calculate the losses. Model supervision is achieved using the reconstruction loss $\ell_{recon} = ||\boldsymbol{I} - \hat{\boldsymbol{I}}||_2^2$ and the perception loss ℓ_{perc} (Johnson et al., 2016) between the input image $\boldsymbol{I}$ and the reconstructed image $\hat{\boldsymbol{I}}$.

To minimize inconsistencies between the inferred positions of the same object when viewed from different directions, we introduce a position loss $\ell_{\text{pos}} = ||\mathbf{x}-\mathbf{x}'||_2^2$ during the first training stage, where $\mathbf{x}$ and $\mathbf{x}'$ denote the inferred positions of the foreground object from two different viewing directions. In addition, we incorporate optional depth ranking (Wang et al., 2023a) and occlusion (Yang et al., 2023) regularizers to mitigate the commonly observed floating artifacts in few-shot NeRFs.

The overall loss function is formulated as follows:

$$\mathcal{L}_{\text{first}} = \ell_{\text{recon}} + \lambda_{\text{perc}}\ell_{\text{perc}} + \lambda_{\text{pos}}\ell_{\text{pos}} \quad \text{(first stage)} \tag{7}$$

$$\mathcal{L}_{\text{second}} = \ell_{\text{recon}} + \lambda_{\text{perc}}\ell_{\text{perc}} \quad \text{(second stage)} \tag{8}$$

$$\mathcal{L}_{\text{third}} = \ell_{\text{recon}} + \lambda_{\text{perc}}\ell_{\text{perc}} + \lambda_{\text{reg}}\ell_{\text{reg}} \quad \text{(third stage)} \tag{9}$$

4 Experiments

We evaluate our method on object segmentation, novel view synthesis, and scene manipulation. Please refer to Appendix D for limitation analysis and Appendix E for more experimental results.

Data. We curate one synthetic and two real-world datasets. Each synthetic scene contains multiple objects with random sizes, positions, and orientations, rendered from 4 directions toward the center. Each real-world scene contains multiple objects at random positions and is captured from 3 poses (for tabletop scenes) or 2 poses (for kitchen backdrops) using mirrorless cameras.

Room-Texture. This synthetic dataset includes 324 chair models from the ABO dataset. Each scene contains four objects set against a background randomly chosen from a collection of floor textures. There are 5000 scenes for training and 100 scenes for evaluation.

Kitchen-Easy. This dataset features scenes with single-color, diffuse dinnerware. Scenes are divided into two categories: the first has a plain tabletop background, and the second introduces a more complex kitchen backdrop. There are 730 scenes for training and 100 for evaluation.

Kitchen-Hard. This dataset comprises scenes featuring textured, specular dinnerware. Similar to Kitchen-Easy, the first half presents a plain backdrop, while the latter has a complex kitchen background. There are 324 scenes for training and 56 for evaluation.

Figure 4 shows some samples drawn from our datasets and more details are provided in Appendix C.2.

Qualitative Metrics. We use three variants of the Adjusted Rand Index (ARI) for scene segmentation: the conventional ARI (calculated on all input image pixels), the Foreground ARI (FG-ARI, calculated

Table 2: Novel view synthesis results on Kitchen-Easy and Kitchen-Hard.

Method	Kitchen-Easy			Kitchen-Hard		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
uORF	26.38	0.816	0.089	19.23	0.602	0.336
QBO	27.30	0.826	0.073	19.78	0.639	0.318
COLF	20.96	0.662	0.333	18.30	0.561	0.397
uOCF (ours)	**28.68**	**0.856**	**0.051**	**28.29**	**0.842**	**0.069**

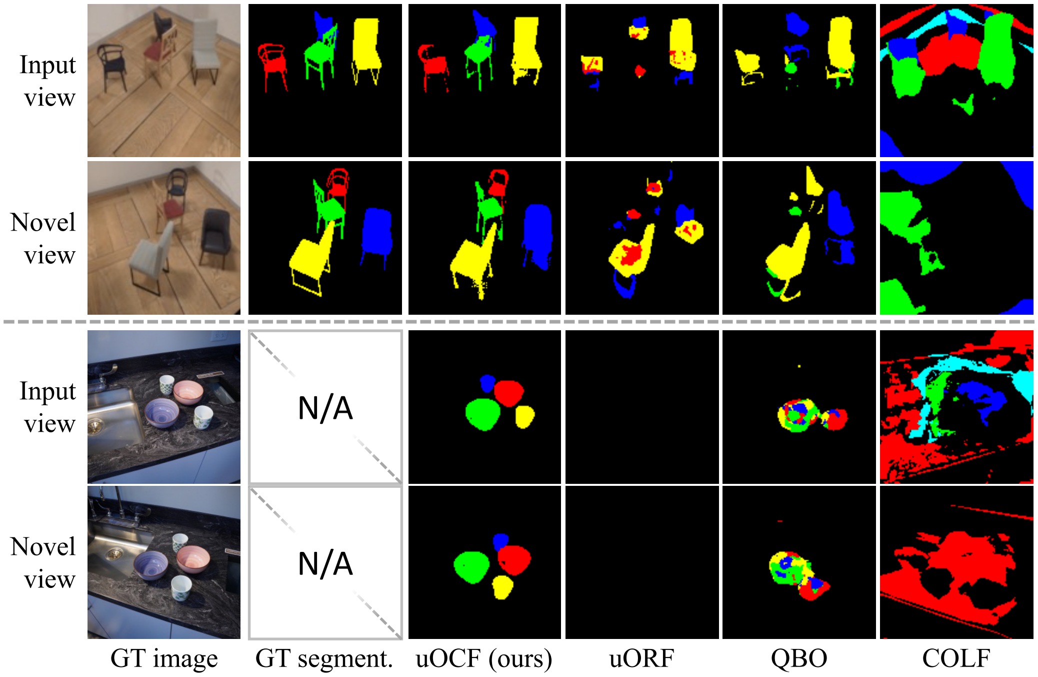


Figure 5: Scene segmentation qualitative results. Novel view images are for reference only.

on foreground input image pixels), and the Novel View ARI (NV-ARI, calculated on novel view pixels). We report the PSNR, SSIM, and LPIPS metrics for novel view synthesis. All scores are computed on images of resolution 128 × 128.

Baselines. We compare our method with uORF (Yu et al., 2022), QBO (Jia et al., 2023), and COLF (Smith et al., 2023). We train one model for each dataset. During testing, the model uses a single image with known camera intrinsics as input and outputs reconstruction/segmentation results from target poses. Further details are provided in Appendix C.3. We increase the latent dimensions and training iterations for the baselines for fair comparisons, thus maintaining similar computational costs across all models. We choose the number of foreground slots $K = 4$ across all methods.

4.1 Object Segmentation in 3D

We begin our evaluation by object segmentation. We render a density map $\mathbf{d}^i$ for each slot i, and then assign each pixel p a segmentation label $s_p = \arg\max_{i=0}^{K} \mathbf{d}_p^i$.

The results are presented in Table 1 and Figure 5. Notably, none of the previous methods could produce reasonable segmentation results in these complex scenes, resulting in extremely low FG-ARI scores. Specifically, uORF binds all objects to the background on Kitchen-Hard, resulting in empty foreground segmentation; COLF produces meaningless results on novel views as the light field does not guarantee multi-view consistency. In contrast, uOCF renders different slots in distinct object-centric frames, making it better distinguish the foreground objects. As a result, uOCF consistently produces satisfactory scene segmentation results, demonstrating its effectiveness in learning object-centric representations. Moreover, uOCF can even handle scenes where objects occlude each other. Please see Appendix E.7 for more results.

Table 3: Quantitative object translation and removal results on Room-Texture.

Method	Object translation			Object removal		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
uORF	23.65	0.654	0.284	23.81	0.664	0.282
QBO	25.21	0.700	0.226	24.58	0.698	0.247
uOCF (ours)	**28.74**	**0.805**	**0.126**	**30.04**	**0.829**	**0.109**

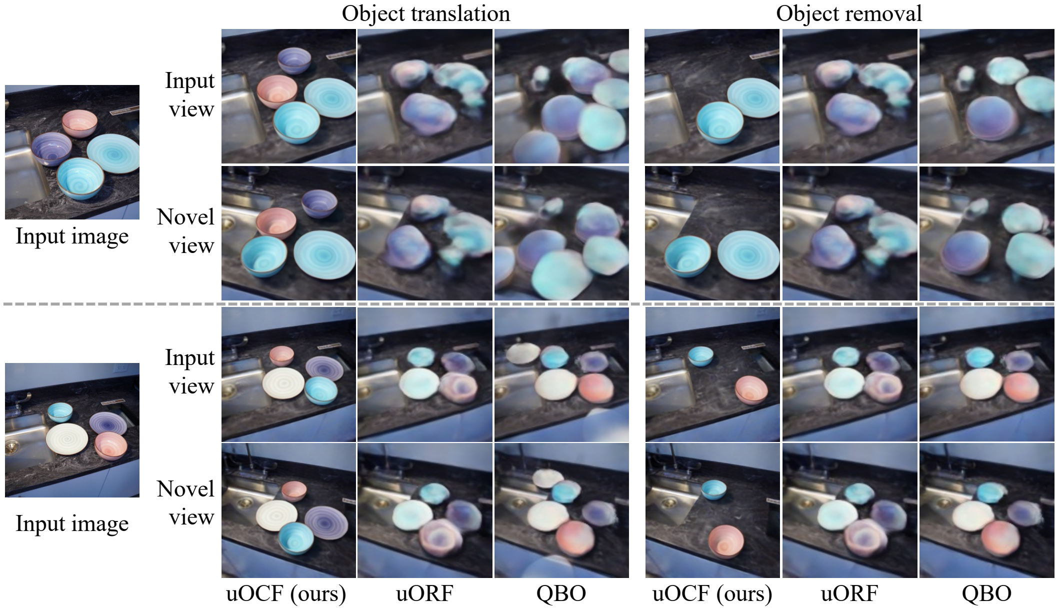


Figure 6: Qualitative results of single-image 3D scene manipulation.

4.2 Novel View Synthesis

We further evaluate our approach through novel view synthesis. For this evaluation, we input a single image from each test scene while the remaining images serve as references.

The results can be found in Tables 1, 2, and Figure 7. Our method outperforms them by a large margin across all metrics compared to existing approaches. Importantly, while previous methods often fail to recover foreground objects, our approach consistently provides accurate scene reconstructions.

4.3 Scene Manipulation in 3D

This section evaluates uOCF's ability in scene manipulation. Being able to infer objects' positions, uOCF readily supports the following scene editing functions: 1) object translation, achieved by modifying the object's position sent to the decoder, and 2) object removal, achieved by excluding certain objects during compositional rendering. In the following, we first quantitatively evaluate uOCF on the Room-Texture dataset, followed by qualitative evaluations on the kitchen datasets.

Quantitative Evaluation. For quantitative evaluation, we randomly select an object within the scene and relatively shift its position (object translation) or remove it (object removal). We render images from four viewpoints for every scene, along with the ground truth mask of the altered object. During testing, we determine the object to manipulate by selecting the object that has the highest IoU score with the ground truth mask. Note that, while existing methods are limited to relative position adjustment, our approach uniquely allows absolute translations due to the disentanglement of object positions and representations. As shown in Table 3, uOCF outperforms all compared methods across all metrics, justifying its effectiveness in scene manipulation.

Qualitative Evaluation. Figure 6 offers the qualitative scene manipulation results. As shown, uORF merges all objects into the background, making scene manipulation unfeasible (hence, we display the un-manipulated images); QBO fails to distinguish foreground objects correctly, resulting in messy manipulation results (see Figure 22 for a visual breakdown of each slot). In contrast, uOCF delivers much more reasonable and visually satisfactory results compared to prior methods. For additional visualization results, please refer to the supplementary video.

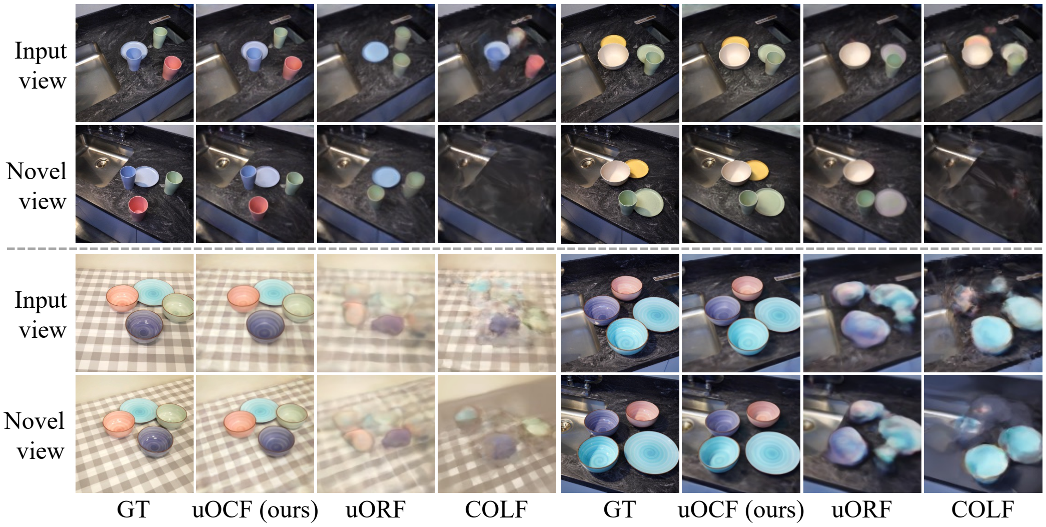


Figure 7: Novel view synthesis qualitative results on Kitchen-Easy (top rows) and Kitchen-Hard (bottom rows).

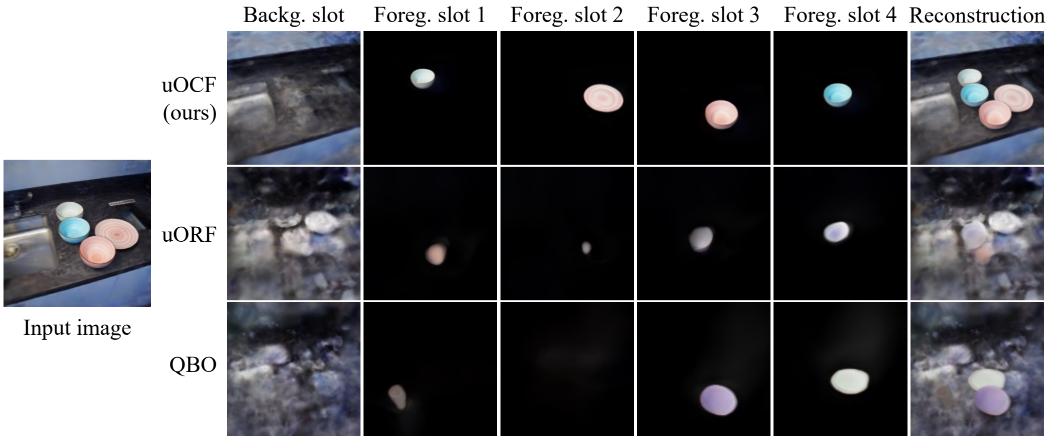


Figure 8: Ablation study on the zero-shot settings. We load the model trained on Kitchen-Easy and perform inference on an image from Kitchen-Hard after a fast test-time optimization.

4.4 Generalizability: Few-Shot and Zero-Shot Object Discovery

We finally explore the generalizability of uOCF. Unlike prior experiments where the entire dataset was used for training, we now consider two more challenging scenarios: 1) Few-shot object discovery, where we train the model using 10/50/100 scenes and evaluate on unseen scenes, and 2) Zero-shot object discovery, where we load the model trained on Kitchen-Easy and perform inference on an image from Kitchen-Hard after a fast test-time optimization for 500 iterations.

We present the zero-shot results in Figure 8 and defer further results and analysis on few-shot settings to Appendix E.2. Notably, whereas previous methods struggle to generalize to unseen objects, uOCF demonstrates outstanding generalizability, only requiring a very fast test-time optimization to generalize from one dataset to new, more complex objects.

5 Conclusion

In this paper, we identify the significance of object-centric modeling for unsupervised 3D object discovery, especially when generalizing to complex real-world scenarios, and propose unsupervised discovery of Object-Centric neural Fields (uOCF) to instantiate this concept. To evaluate our approach, we collect three datasets, each with scenes containing objects of multiple categories set against complex backgrounds. Our results indicate that the object-centric design, when combined with appropriate training strategies, can substantially improve the generalizability of unsupervised representation learning.

Reproducibility Statement

To promote reproducibility, upon the acceptance of this paper, we will make available a comprehensive suite of resources. This includes the repository containing both training and test code, the entirety of the three datasets, as well as the pre-trained models on these datasets. Additionally, we will provide detailed guidelines for utilizing our code on new datasets. For further in-depth understanding and ease of re-implementation, we have elaborated on the details for re-implementation in Appendix C.3.

References

Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021. 15, 16

Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv:2302.12288*, 2023. 14

Ondrej Biza, Sjoerd van Steenkiste, Mehdi SM Sajjadi, Gamaleldin F Elsayed, Aravindh Mahendran, and Thomas Kipf. Invariant slot attention: Object discovery with slot-centric reference frames. In *ICML*, 2023. 2, 4

Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv:1901.11390*, 2019. 2

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *arXiv*, 2015. 19

Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. Learning to balance specificity and invariance for in and out of domain generalization. In *ECCV*, 2020. 1

Chang Chen, Fei Deng, and Sungjin Ahn. Learning to infer 3d object models from images. *arXiv:2006.06130*, 2020. 2

Honglin Chen, Rahul Venkatesh, Yoni Friedman, Jiajun Wu, Joshua B Tenenbaum, Daniel LK Yamins, and Daniel M Bear. Unsupervised segmentation in real-world images via spelke object inference. In *ECCV*, 2022. 5, 17

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014. 4

Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, Matthieu Guillaumin, and Jitendra Malik. Abo: Dataset and benchmarks for real-world 3d object understanding. In *CVPR*, 2022. 15

Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. In *AAAI*, 2019. 2

Weijian Deng, Stephen Gould, and Liang Zheng. On the strong correlation between model invariance and generalization. In *NeurIPS*, 2022. 1

Aniket Didolkar, Anirudh Goyal, and Yoshua Bengio. Cycle consistency driven object discovery. *arXiv:2306.02204*, 2023. 2

Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. *arXiv:1907.13052*, 2019. 2

SM Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In *NeurIPS*, 2016. 2

SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 2018. 2

Kristen Grauman and Trevor Darrell. Unsupervised learning of categories from sets of partially matching image features. In *CVPR*, 2006. 2

Klaus Greff, Antti Rasmus, Mathias Berglund, Tele Hotloo Hao, Jürgen Schmidhuber, and Harri Valpola. Tagger: Deep unsupervised perceptual grouping. In *NeurIPS*, 2016. 2

Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. In *NeurIPS*, 2017. 2

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *ICML*, 2019. 2

Baoxiong Jia, Yu Liu, and Siyuan Huang. Improving object-centric learning with query optimization. In *ICLR*, 2023. 1, 2, 6, 7, 24

Jindong Jiang, Sepehr Janghorbani, Gerard de Melo, and Sungjin Ahn. Scalor: Generative world models with scalable object representations. In *ICLR*, 2020. 2

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 6

Armand Joulin, Francis Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010. 2

Adam R Kosiorek, Hyunjik Kim, Ingmar Posner, and Yee Whye Teh. Sequential attend, infer, repeat: Generative modelling of moving objects. In *NeurIPS*, 2018. 2

Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *CVPR*, 2018. 2

Bo Li, Zhengxing Sun, Qian Li, Yunjie Wu, and Anqi Hu. Group-wise deep object co-segmentation with co-attention recurrent neural network. In *CVPR*, 2019. 2

Shengnan Liang, Yichen Liu, Shangzhe Wu, Yu-Wing Tai, and Chi-Keung Tang. Onerf: Unsupervised 3d object segmentation from multiple views. *arXiv:2211.12038*, 2022. 2

Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. In *ICLR*, 2020. 2

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *NeurIPS*, 2020. 2, 3

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 3, 4

Tom Monnier, Elliot Vincent, Jean Ponce, and Mathieu Aubry. Unsupervised layered image decomposition into object prototypes. In *ICCV*, 2021. 2

Norman Müller, Andrea Simonelli, Lorenzo Porzi, Samuel Rota Bulo, Matthias Nießner, and Peter Kontschieder. Autorf: Learning 3d object radiance fields from single view observations. In *CVPR*, 2022. 2

Thu H Nguyen-Phuoc, Christian Richardt, Long Mai, Yongliang Yang, and Niloy Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. In *NeurIPS*, 2020. 3

Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021. 3

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv:2304.07193*, 2023. 3

Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *CVPR*, 2021. 2

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 3

Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv:2109.08238*, 2021. 18, 19

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *TPAMI*, 2022. 15

Bryan C Russell, William T Freeman, Alexei A Efros, Josef Sivic, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006. 2

Mehdi SM Sajjadi, Daniel Duckworth, Aravindh Mahendran, Sjoerd van Steenkiste, Filip Pavetic, Mario Lucic, Leonidas J Guibas, Klaus Greff, and Thomas Kipf. Object scene representation transformer. In *NeurIPS*, 2022. 2, 16

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *NeurIPS*, 2019. 3

Josef Sivic, Bryan C Russell, Alexei A Efros, Andrew Zisserman, and William T Freeman. Discovering objects and their location in images. In *ICCV*, 2005. 2

Cameron Smith, Hong-Xing Yu, Sergey Zakharov, Fredo Durand, Joshua B Tenenbaum, Jiajun Wu, and Vincent Sitzmann. Unsupervised discovery and composition of object light fields. *TMLR*, 2023. 1, 2, 6, 7

Elizabeth S Spelke. Principles of object perception. *Cognitive science*, 1990. 5, 17

Karl Stelzner, Kristian Kersting, and Adam R Kosiorek. Decomposing 3d scenes into objects via unsupervised volume segmentation. *arXiv:2104.01148*, 2021. 1, 2

Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. State of the art on neural rendering. *CGF*, 2020. 3

Huy V Vo, Patrick Pérez, and Jean Ponce. Toward unsupervised, multi-object discovery in large-scale image collections. In *ECCV*, 2020. 2

Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *ICCV*, 2023a. 6

He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *CVPR*, 2019. 5

Qian Wang, Yiqun Wang, Michael Birsak, and Peter Wonka. Blobgan-3d: A spatially-disentangled 3d-aware generative model for indoor scenes. *arXiv:2303.14706*, 2023b. 3

Jiajun Wu, Joshua B Tenenbaum, and Pushmeet Kohli. Neural scene de-rendering. In *CVPR*, 2017. 2

Qianyi Wu, Xian Liu, Yuedong Chen, Kejie Li, Chuanxia Zheng, Jianfei Cai, and Jianmin Zheng. Object-compositional neural implicit surfaces. In *ECCV*, 2022. 3

Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *CGF*, 2022. 3

Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *CVPR*, 2021. 3

Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *CVPR*, 2023. 6

Yafei Yang and Bo Yang. Promising or elusive? unsupervised object segmentation from real-world single images. In *NeurIPS*, 2022. 20

Shunyu Yao, Tzu Ming Harry Hsu, Jun-Yan Zhu, Jiajun Wu, Antonio Torralba, William T Freeman, and Joshua B Tenenbaum. 3d-aware scene manipulation via inverse graphics. In *NeurIPS*, 2018. 2

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 16

Hong-Xing Yu, Leonidas J Guibas, and Jiajun Wu. Unsupervised discovery of object radiance fields. In *ICLR*, 2022. 1, 2, 3, 6, 7, 15, 16, 19, 20, 24

Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2019. 1

A Supplementary Document Overview

This supplementary document is structured as follows: We commence with the proof of concept in Appendix B. Subsequently, Appendix C provides comprehensive implementation details. Our discussion then progresses to our approach's limitations and potential failure modes in Appendix D. Finally, we present additional experiments in Appendix E. Accompanying this document is an *overview video* attached in the supplementary file.

B Proof of Concept

We conduct a toy experiment (Figure 9) to demonstrate that our model has successfully learned object position, rotation, and scale. In this experiment, we begin with two images (input 1 and input 2) of a chair placed at the scene's center, exhibiting different sizes (on the left) or rotation angles (on the right), all captured from the same viewing direction.

We extract the object latents from these images, interpolate them, and then send the interpolated latents to the decoder. As shown between the two input images, we observe a smooth transition in both object size and rotation, indicating that the latent representation has effectively captured the scale and rotation of objects.

In the second row, we placed the chairs in different positions. As shown on the right, we obtained a smooth transition again, proving that our model could disentangle object positions from the latent representation.

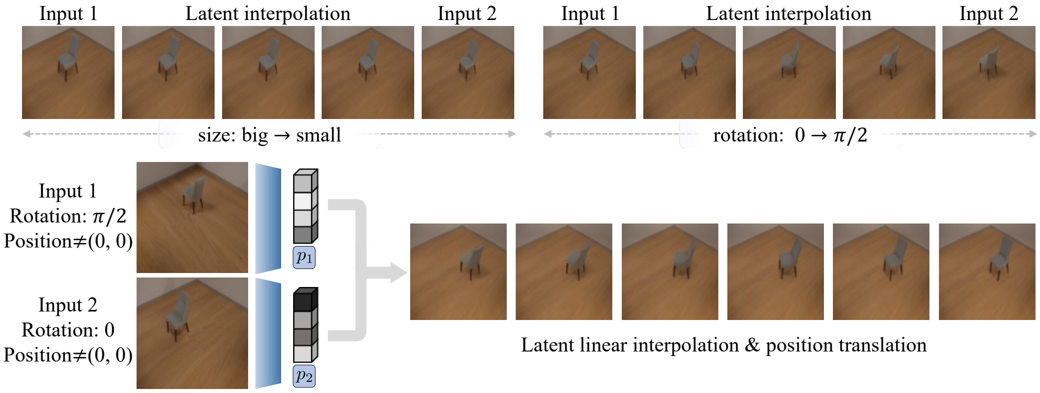


Figure 9: Proof of concept.

C Implementation

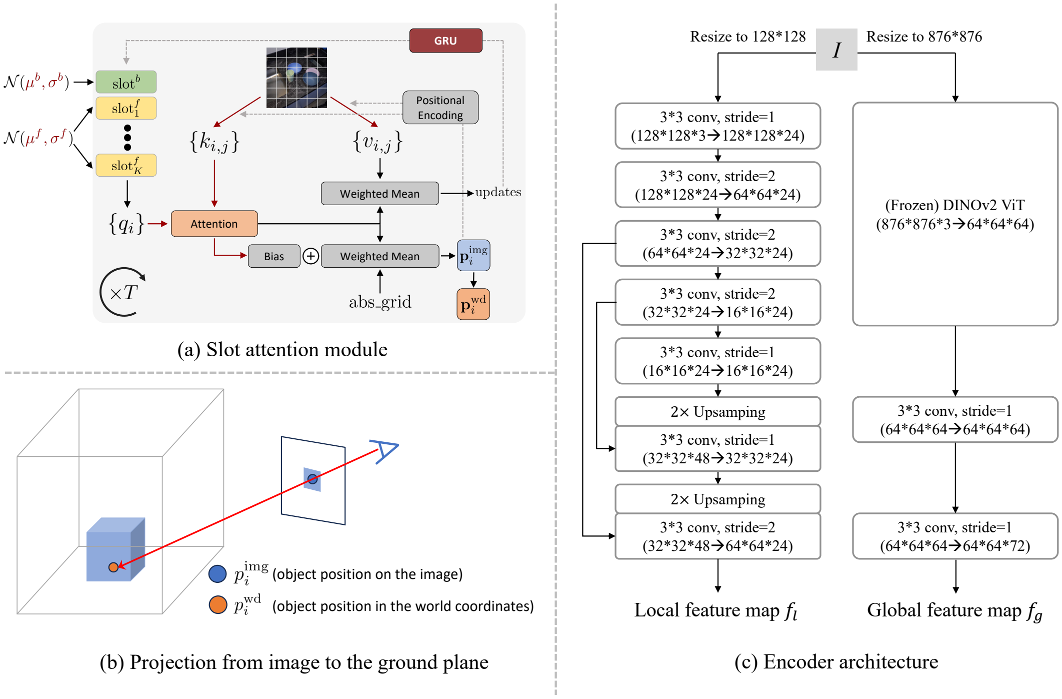
C.1 Model Architecture

Dual-Route Feature Map Extraction. We employ a dual-route encoder to extract both global and local features jointly. As shown in Figure 10(c), the local route is a trainable U-Net, while the global route incorporates a frozen DINOv2 ViT, followed by two convolutional layers. A ReLU activation, except the last one, follows each convolutional layer. Note that we use f_g as the "inputs" to the latent extraction module, and then concatenate the output of the latent extraction module with the attention weighted-mean of f_l to derive the final slot latents $\{\mathbf{z}_i\}_{i=0}^{K}$.

Latent Extraction Module. We provide an illustration of our latent extraction module in Figure 10(a). Note that we assume all objects are on a ground plane in the world frame. Thus, the object's position on the image plane can be directly mapped to its position in the world coordinates. See an illustration in Figure 10(b). This assumption is reasonable in many cases, such as outdoor scenes, indoor navigation scenes, and robotic tabletop manipulation scenes. Additionally, one may employ depth sensors or monocular metric depth estimators (Bhat et al., 2023) to predict the distance between the object and the camera, thus eliminating this assumption.

C.2 Data Collection

This section introduces the details of our datasets.


GRU
$\mathcal{N}(\mu^b, \sigma^b)$
slot^b
slot$_1^f$
$\mathcal{N}(\mu^f, \sigma^f)$
slot$_K^f$
Positional Encoding
$\{k_{i,j}\}$
$\{v_{i,j}\}$
Weighted Mean
updates
$\{q_i\}$
Attention
Bias
Weighted Mean
$\mathbf{p}_i^{\text{img}}$
$\mathbf{p}_i^{\text{wd}}$
$\times T$
abs_grid
p_i^{img} (object position on the image)
p_i^{wd} (object position in the world coordinates)
Resize to 128*128
I
Resize to 876*876
3*3 conv, stride=1 (128*128*3→128*128*24)
3*3 conv, stride=2 (128*128*24→64*64*24)
3*3 conv, stride=2 (64*64*24→32*32*24)
3*3 conv, stride=2 (32*32*24→16*16*24)
3*3 conv, stride=1 (16*16*24→16*16*24)
2× Upsampling
3*3 conv, stride=1 (32*32*48→32*32*24)
2× Upsampling
3*3 conv, stride=2 (32*32*48→64*64*24)
(Frozen) DINOv2 ViT (876*876*3→64*64*64)
3*3 conv, stride=1 (64*64*64→64*64*64)
3*3 conv, stride=1 (64*64*64→64*64*72)
Local feature map f_l
Global feature map f_g

(a) Slot attention module

(b) Projection from image to the ground plane

(c) Encoder architecture

Figure 10: (a): Illustration of our slot attention module. (b): Compute the object's position in the world coordinates based on its position on the image. (c): Detailed architecture of our dual-route encoder module.

Room-Texture. In Room-Texture, objects are chosen from 324 ABO objects (Collins et al., 2022) from the "armchair" category. In the single-object dataset, we create 4 scenes for each object, resulting in a dataset of 1296 scenes. The multiple-object dataset includes 5,000 scenes for training and 100 for evaluation, each containing 4 different objects.

Kitchen-Easy. In Kitchen-Easy, objects are diffuse and have no texture. The dataset comprises 16 objects and 6 tablecloths in total. We captured 3 images for each tabletop scene and 2 for each kitchen scene. This dataset contains 730 scenes for training and 100 for evaluation, with each containing 4 objects. We calibrate the cameras using the OpenCV library.

Kitchen-Hard. In Kitchen-Hard, objects are specular, and the lighting is more complex. The dataset comprises 12 objects and 6 tablecloths, and the other settings are identical to Kitchen-Easy. This dataset contains 324 scenes for training and 56 scenes for evaluation.

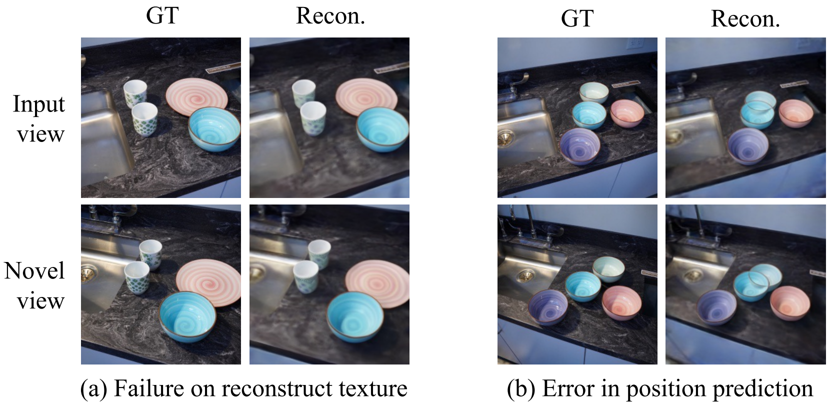
C.3 Training Configuration

This section briefly discusses the training configuration of uOCF.

We employ Mip-NeRF (Barron et al., 2021) as our NeRF backbone and estimate the depth maps by MiDaS (Ranftl et al., 2022). An Adam optimizer with default hyper-parameters and an exponential decay scheduler is used across all experiments. The initial learning rate for all stages is set to 0.0003. Loss weights are set to $\lambda_{\text{perc}} = 0.006, \lambda_{\text{pos}} = 0.1, \lambda_{\text{depth}} = 1.5$, and $\lambda_{\text{occ}} = 0.1$, where $\lambda_{\text{reg}}\ell_{\text{reg}} = \lambda_{\text{depth}}\ell_{\text{depth}} + \lambda_{\text{occ}}\ell_{\text{occ}}$.

Coarse-To-Fine Progressive Training. We employ a coarse-to-fine strategy in our second training stage to facilitate training at higher resolutions. Reference images are downsampled to a lower resolution during the coarse training stage and replaced by image patches randomly cropped from the original high-resolution images. Additionally, to stabilize the training procedure, we introduce ℓ_{perc} and ℓ_{depth} progressively to the training objective instead of incorporating them from the beginning.

Training Configuration on Room-Texture. During stage 1, we train the model for 200 epochs directly on images of resolution 128 × 128. We enforce the locality constraint (a bounding box for foreground slots) (Yu et al., 2022) during the first 50 epochs. We start with the reconstruction loss only, then add the perceptual and position losses at the 50th, 100th epoch, respectively. During stage



(a) Failure on reconstruct texture (b) Error in position prediction

Figure 11: Failure case visualizations. Our method may fail to reconstruct intricate object texture (a) or predict biased object position.

2, we train the model for 40 epochs on the coarse stage and another 40 on the fine stage. The two stages use input images of resolution 64 × 64 and 128 × 128, respectively, and both stages use patch size 64 × 64. We start with the reconstruction loss only, then add the perceptual loss at the 10th epoch, and start the object-centric sampling from the 20th epoch.

Training Configuration on Kitchen-Easy and Kitchen-Hard. On both kitchen datasets, we follow the three-stage pipeline *i.e.*, load the model pre-trained on Room-Texture. We train the model for 1250 epochs, where the fine stage starts at the 250th epoch. The coarse stage uses images of resolution 64 × 64. For the fine stage, we use an input resolution of 128 × 128 and a patch size of 64 × 64 for the first 750 epochs. Subsequently, we use a resolution of 256 × 256 and a patch size of 80 × 80 for the next 250 epochs. We add the perceptual loss at the 100th epoch, start the object-centric sampling from the 200th epoch, and add the depth loss at the 1000th epoch.

D Limitations and Failure Cases

Limitation of Real Datasets. The kitchen datasets we have collected are still not complex enough compared to in-the-wild scenes, as they all have a clean background with minimal noise. Besides, the real objects have relatively simple textures and shapes. To address these drawbacks, future work may incorporate more advanced 3D object representations to represent the objects in a more compact and memory-friendly way. Generalizing our method to more complicated scenes is an interesting research direction.

Limitation on Reconstruction Quality. Scene-level generalizable NeRFs (Yu et al., 2021; Sajjadi et al., 2022; Yu et al., 2022) commonly face challenges in accurately reconstructing detailed object textures. To address this limitation, we integrated the MipNeRF (Barron et al., 2021) backbone into our approach and included the perceptual and depth losses. However, despite these enhancements, our approach still encounters difficulty capturing extremely high-frequency details. As shown in Figure 11(a), our method fails to replicate the mug's detailed texture. Future research may benefit from exploring more advanced NeRF backbones to further improve texture detail reconstruction.

Failure in Position Prediction. Our three-stage training pipeline, despite its robustness in many situations, is not immune to errors, particularly in object position prediction. Complexities arise due to occlusions between objects, where using the attention-weighted mean for determining object positions can lead to inaccuracies. Although a bias term can rectify this in most instances (Figure 7), discrepancies persist under a few conditions, as depicted in Figure 11(b).

E Additional Experiments

E.1 Ablation Studies

Encoder Design. In this ablation study, we address two questions concerning our encoder design: (1) Is the modification of the encoder (replacing the shallow U-Net with DINO) adequate for unsupervised object discovery in complex real-world scenes? (2) Given the integration of the powerful DINO

Method	Scene segmentation			Novel view synthesis		
	ARI↑	FG-ARI↑	NV-ARI↑	PSNR↑	SSIM↑	LPIPS↓
uORF	0.649	0.108	0.587	24.37	0.688	0.251
uORF-DINO	0.688	0.677	0.656	25.33	0.739	0.240
uORF-DR	0	0	0	25.38	0.698	0.322
uOCF-DINO	0.652	0.346	0.621	26.06	0.723	0.251
uOCF-IM	**0.806**	0.749	**0.752**	27.77	0.753	0.182
uOCF (ours)	0.802	**0.785**	0.747	**28.96**	**0.803**	**0.121**

Table 4: Ablation study for different encoder designs on Room-Texture. uORF-DINO and uORF-DR modify the standard uORF by replacing its shallow U-Net encoder with the DINO encoder and our dual-route encoder, respectively. The uOCF-DINO drops the U-Net route and utilizes the DINO encoder only, and uOCF-IM incorporates DINO's intermediate layer features instead of uOCF's shallow encoder. Note that uORF-DR binds all foreground objects to the background, leading to an ARI score of zero.

Method	PSNR↑	SSIM↑	LPIPS↓
uORF (baseline)	19.23	0.602	0.336
Omit stage 1&2	19.67	0.565	0.568
Omit stage 2	24.52	0.769	0.157
Omit object-centric sampling	27.89	**0.843**	0.083
Adapt from CLEVR	27.32	0.833	0.092
uOCF (full)	**28.29**	0.842	**0.069**

Table 5: Ablation study for the training pipeline on Kitchen-Hard.

encoder, what is the rationale behind retaining a U-Net route? The answers to these questions are presented in Table 4 and Figure 12.

Addressing the first question, our findings show that despite DINO's efficacy in feature extraction, merely replacing the shallow U-Net with DINO does not overcome the inherent limitation of uORF in entangling the foreground from background elements. This is exemplified in uORF-DINO, which tends to mix different objects' parts into a single foreground slot akin to the original uORF. Meanwhile, uORF-DR, featuring our dual-route encoder, incorrectly binds all foreground objects into the background, leading to zero ARI scores. These results highlight the necessity of disentangling object position from its intrinsic properties proposed in our method.

As for the second question, employing DINO exclusively for feature extraction (uOCF-DINO) degrades both scene segmentation performance and scene reconstruction quality. This can be attributed to the loss of object-specific details in the final DINO features, which fails to provide the decoder with sufficient information about intra-category object differences. We also experimented with substituting the U-Net route in our encoder with DINO's intermediate features (uOCF-DR) but still obtained comparably inferior results. As depicted in the third and fourth columns of Figure 12, these approaches resulted in degraded object reconstruction visual qualities.

Training Pipeline. We then validate the efficacy of our three-stage object-centric training pipeline. The first stage instructs the model on the fundamentals of an object-centric NeRF, such as the physical coherence (Spelke, 1990)–a crucial aspect in unsupervised segmentation (Chen et al., 2022). The subsequent stage refines the model's ability to predict object positions and segregate them into individual slots, thereby equipping it for handling complex real-world scenes.

In other words, although the objects in the two datasets might share a large domain gap, the learned object priors are transferable. The main paper already presents the model's adaptability from synthetic chairs to actual dinnerware. To further validate this concept, we consider using the simple CLEVR shapes (balls, cubes, and cylinders) for object prior learning, which maintains fairly good results, as shown in Table 5.

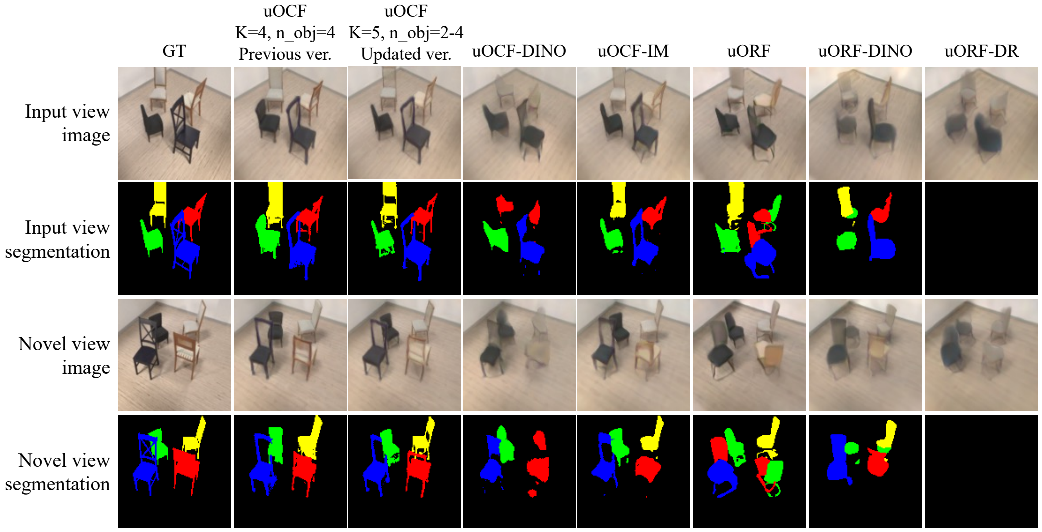


Figure 12: Ablation study for the encoder design. Results show that merely integrating DINO fails to address the limitations of existing methods in complex scenes.

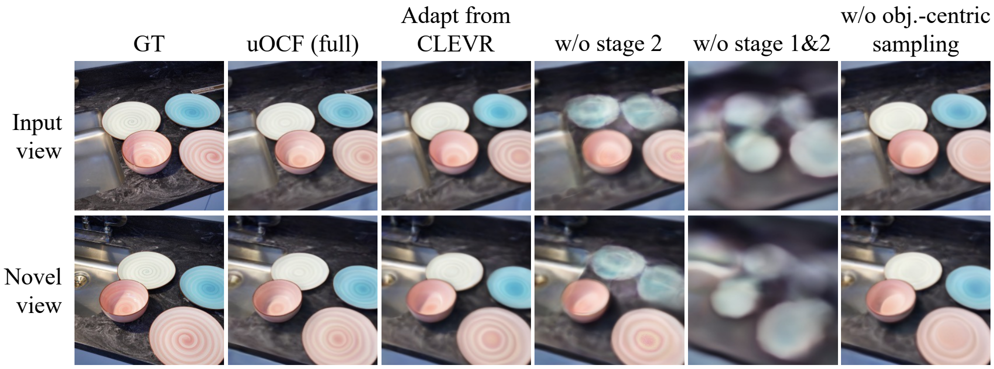


Figure 13: Qualitative ablation study for the training pipeline on Kitchen-Hard. Removing either training stage on the synthetic dataset significantly impairs performance, while omitting the object-centric sampling technique results in a noticeable decline in visual quality.

As illustrated in Table 5 and Figure 13, omitting both synthetic stages leads to the model erroneously associating the entire scene with the background slot, yielding significantly poorer results. Similarly, excluding the second stage hampers the model's ability to accurately identify all objects in a scene, resulting in a substantial performance decline. We also present results from eliminating the object-centric sampling strategy, which slightly negatively impacts the visual quality.

E.2 Generalizability: Few-Shot and Zero-shot Object Discovery

Extending the discussions from Sec. 4.4, this section delves into the generalizability of uOCF. In contrast to previous experiments that utilized the full Kitchen-Harddataset for training, we now embark on a more challenging scenario. Here, the model is trained with a limited number of scenes—specifically, 10, 50, or 100—and then evaluated on unseen scenes.

Figure 14 illustrates a notable trend: the reconstruction quality of uOCF deteriorates in correlation with the reduction in the number of training scenes. Particularly, with an extremely limited training set, the model struggles to differentiate objects from the background. This underscores the significance of a substantial dataset size for effective unsupervised object discovery.

Furthermore, to assess the zero-shot generalizability of our model, we render images from the complex indoor scene dataset HM3D (Ramakrishnan et al., 2021) and conduct zero-shot inference

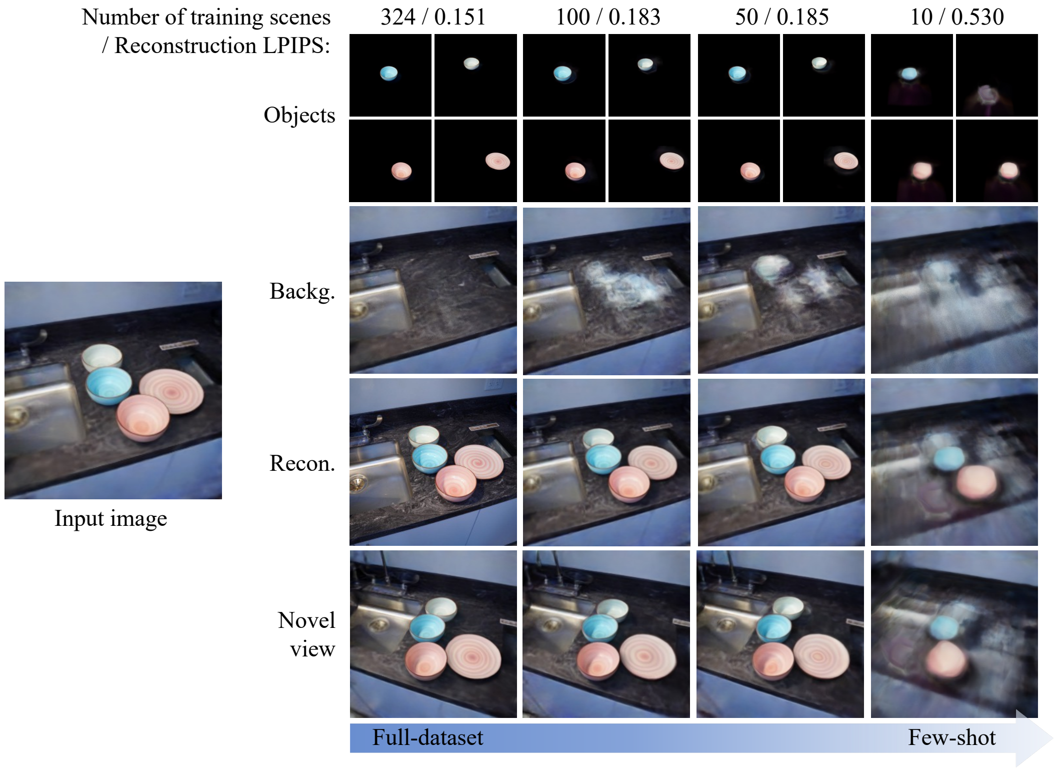


Figure 14: Ablation study on the few-shot settings. We decrease the number of scenes available during training. The LPIPS values are computed between images of resolution 256 × 256.

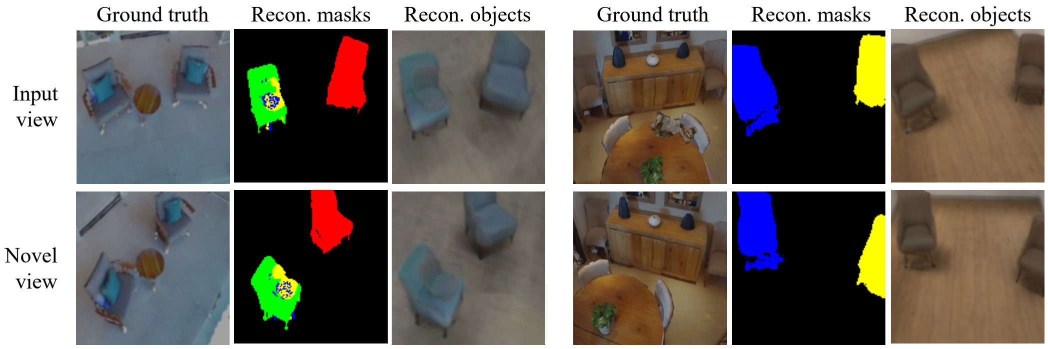


Figure 15: Zero-shot inference on the HM3D (Ramakrishnan et al., 2021) dataset. Our method can discover and segment the chair instances in the scene, thereby delivering plausible reconstruction results.

using the model pre-trained on Room-Texture. The results, as depicted in Figure 15, show that our method can accurately identify and segment the chair instances in the scene and deliver plausible reconstruction results.

E.3 Results on Room-Diverse

To comprehensively compare established methods, we further include results on the Room-Diverse dataset as introduced in (Yu et al., 2022). This dataset is comprised of synthetic scenes featuring a variety of foreground object shapes against diverse background visuals. Each scene is constructed with four distinct chairs, whose shapes are randomly selected from a pool of 1,200 ShapeNet chair models (Chang et al., 2015). These chairs vary in size, position, and orientation. The backgrounds are sampled from 50 different floor textures sourced from the web. The dataset encompasses 5,000 scenes for training and 100 for testing. While we follow the official rendering

Table 6: Scene segmentation and novel view synthesis result on Room-Diverse. Comparison methods are uORF, uORF with DINO encoder (uORF-D), uORF with our dual-route encoder (uORF-DR), and QBO.

Method	Scene segmentation			Novel view synthesis		
	ARI↑	FG-ARI↑	NV-ARI↑	PSNR↑	SSIM↑	LPIPS↓
uORF	0.638	0.705	0.494	25.11	0.683	0.266
uORF-DINO	0.692	0.555	0.633	25.50	0.698	0.239
uORF-DR	0.742	0.653	0.680	26.00	0.707	0.209
QBO	0.724	0.716	0.618	24.49	0.680	0.182
uOCF (ours)	**0.769**	**0.828**	**0.688**	**27.31**	**0.751**	**0.141**

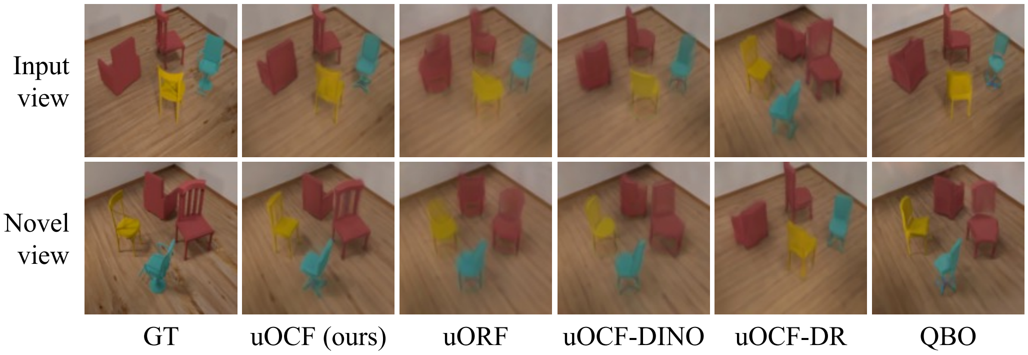


Figure 16: Qualitative comparison results on Room-Diverse.

script provided by Yu et al. (2022), we introduce a greater degree of variation in the sizes of the objects.

The quantitative results in Table 6 and the qualitative results in Figure 16 clearly demonstrate that our method significantly surpasses current approaches, further justifying the effectiveness of our approach.

E.4 Additional Evaluation Metrics

Following the analysis by Yang & Yang (2022), we extend our evaluation of scene segmentation by employing the Average Precision (AP) metric. Specifically, we consider two kinds of APs: Input view-AP for the input view and Novel view-AP for novel views, with the results detailed in 7. In contrast to existing methods like uORF, which demonstrate notably low AP scores due to their inability to accurately segregate foreground objects (as visualized in Figure 5), our method achieves notably higher scores, highlighting the efficacy of our approach.

E.5 Discussion on the Number of Slots

The above experiments are all conducted on scenes with exactly 4 objects for both training and evaluation, with the number of slots K also set to 4. In this section, we first explore how uOCF trained on scenes with 4 objects behaves on test scenes with less than 4 objects. Then, we discuss the effect of K (the number of slots) in uOCF. Specifically, we consider randomizing the number of objects in stage 2 (each scene contains 2$\sim$4 chair instances) and setting the number of slots K exceeds the maximum number of objects in the scene ($K = 5$). We use a new Room-Texture test set for the latter two configurations for evaluation, with each scene containing 2$\sim$4 objects.

Testing on Scenes with Fewer Objects. We train our model on scenes with exactly 4 objects and try testing it on scenes with 3 objects. As depicted in Figure 17, in scenarios where the scene comprises 3 objects, it is observed that multiple slots may bind to a single object. This might be attributed to all training scenes containing exactly 4 objects. However, the reconstructions maintain notably high fidelity.

Randomized Number of Objects in Training Scenes. As shown in Tables 8, 9 and Figure 18, uOCF performs reasonably well when trained on scenes with $\leq K(= 4)$ objects, justifying that the

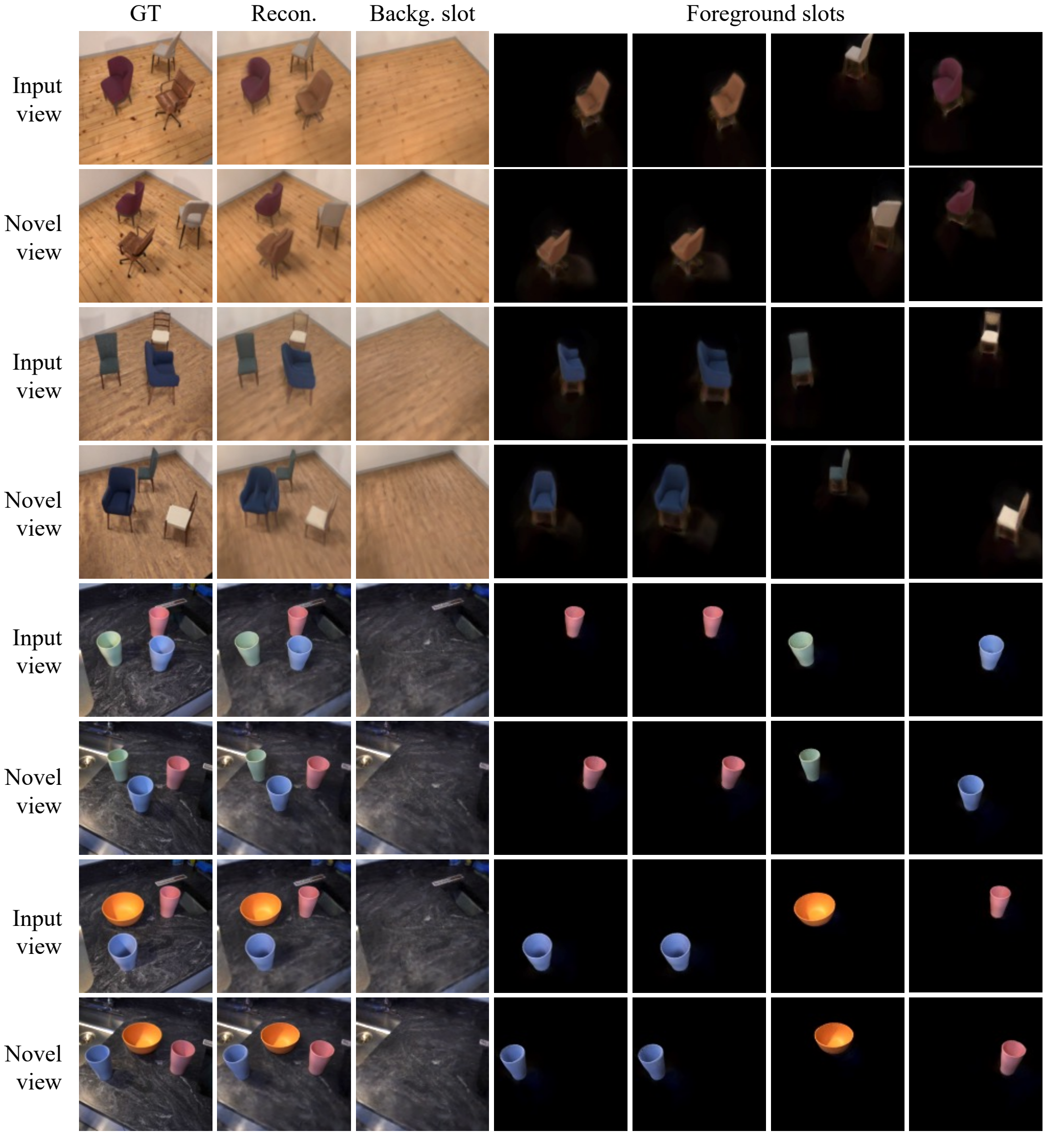


Figure 17: Qualitative results of uOCF on scenes with less than four objects.

Metric	uORF	QBO	COLF	uOCF (ours)
Input view-AP	0.005	0.359	0.315	**0.782**
Novel view-AP	0.001	0.195	0.015	**0.770**

Table 7: Quantitative scene segmentation results using the AP metric on Room-Texture.

outstanding performance of uOCF in real-world scenes is not because of the identical number of objects between stages 2/3. In other words, applying the "object prior" learned from synthetic scenes to real-world scenes neither requires two datasets to have the same number of objects nor K to equal the number of objects in the scene. We can still assume a shared maximum number of objects of the two datasets, similar to previous unsupervised object discovery literature.

Moreover, empty slots now appear after using a training set consisting of a randomized number of objects (please see Figure 18), overcoming the over-segmentation problem. Regarding under-segmentation, in Figure 12, we observe that only our model (both the previously trained model and the newly trained model) can overcome under-segmentation, whereas previous methods fail to segment all the chair details.

Table 8: Quantitative scene segmentation and novel view synthesis result on Room-Diverse with a different number of slots (K) and a random number of objects in the scene. The line with gray denotes the original configuration. Each scene for evaluation contains 2∼4 objects. We compare the training results of 15 epochs (75000 iterations), where coarse and fine training stages last 10 and 5 epochs, respectively.

Method	Scene segmentation			Novel view synthesis		
	ARI↑	FG-ARI↑	NV-ARI↑	PSNR↑	SSIM↑	LPIPS↓
$K = 4, \text{n\_obj} = 4$	0.819	0.643	0.743	28.68	0.803	0.139
$K = 4, \text{n\_obj} \in \{2, 3, 4\}$	0.828	0.743	0.756	30.11	0.831	0.112
$K = 5, \text{n\_obj} \in \{2, 3, 4\}$	0.819	0.559	0.769	28.72	0.805	0.132

Table 9: Quantitative scene segmentation and novel view synthesis result on Kitchen-Hard with a different number of slots (K) and a random number of objects in the synthetic scenes of stage 2. The line with gray denotes the original configuration. We compare the training results of 150 epochs (48600 iterations) of the coarse training stage.

Method	PSNR↑	SSIM↑	LPIPS↓
$K = 4, \text{n\_obj} = 4$	27.36	0.820	0.140
$K = 4, \text{n\_obj} \in \{2, 3, 4\}$	27.10	0.825	0.131
$K = 5, \text{n\_obj} \in \{2, 3, 4\}$	28.26	0.837	0.120

E.6 Additional Real-World Results

Besides Kitchen-Easy and Kitchen-Hard, we gather an additional real-world dataset of relatively low difficulty, namely "Planters". Sample images of this dataset are shown in Figure 19.

Planters. This dataset includes scenes with four plant pots or vases arranged on a tabletop decorated with tablecloths. We gathered 9 plant pots, 8 vases, and 6 tablecloths and captured 745 scenes for training and 140 for evaluation, each with 3 images from different poses. We train the model for 750 epochs, where the fine stage starts at the 250th epoch. The coarse stage and fine stage use images of resolution 64×64 and 128×128, respectively, and both stages use patch size 64×64. We add the perceptual loss from the 100th epoch and start the object-centric sampling from the 200th epoch.

As shown by the quantitative in Table 10 and qualitative results in Figure 20, our method significantly surpasses previous methods and delivers scene reconstruction and novel view synthesis results with much higher visual quality.

E.7 Additional Qualitative Results

Transparent Objects. We first apply our model to handle (semi-)transparent objects during test time.

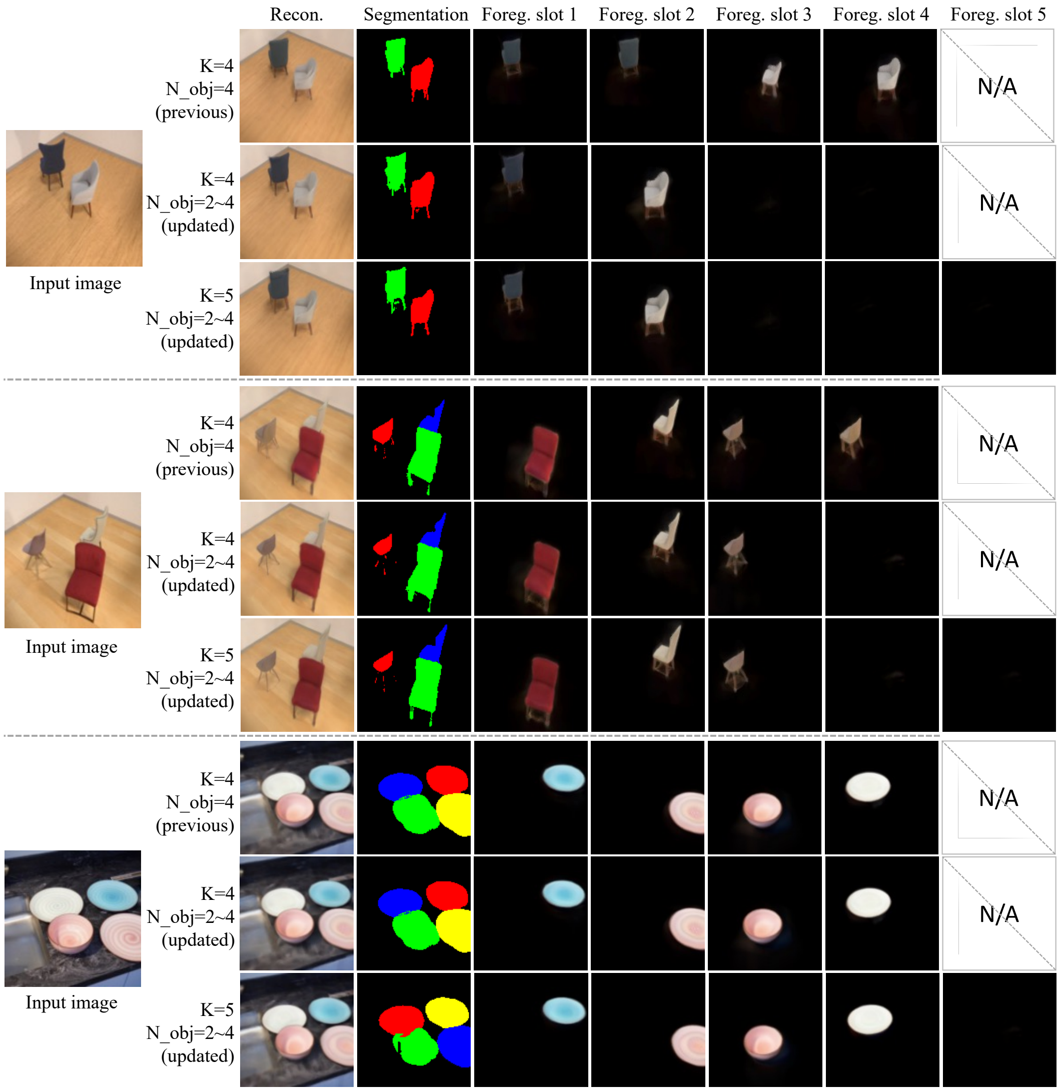


Figure 18: Qualitative scene segmentation and novel view synthesis result on Room-Texture and Kitchen-Hard with a different number of slots (K) and a random number of objects in the synthetic dataset.

Method	PSNR↑	SSIM↑	LPIPS↓
uORF	24.49	0.748	0.163
uORF-QBO	28.09	0.847	0.108
COLF	19.22	0.588	0.464
uOCF (ours)	**29.00**	**0.864**	**0.062**

Table 10: Quantitative results on the Planters dataset.

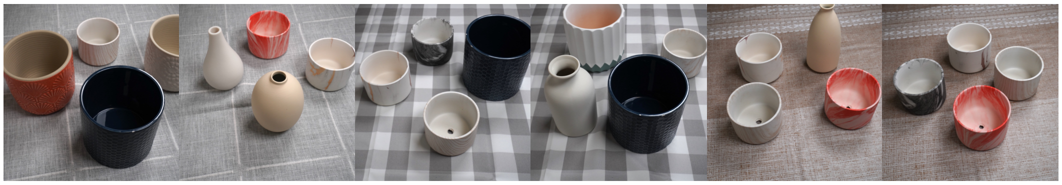


Figure 19: Sample images of the Planters dataset.

As shown in Figure 21, our model ignores transparency due to the absence of transparent objects in its training dataset. However, it still demonstrates reasonable object segmentation and reconstruction capabilities.

Single-Slot Visualizations. Next, we present the single-slot visualization for our scene manipulation experiment (Sec. 4.3) in Figure 22. Notably, uORF (Yu et al., 2022) puts all objects within the background slot, whereas QBO (Jia et al., 2023) produces identical results across all foreground slots. Contrasting these, uOCF accurately differentiates between the foreground objects and the background.

Scene Segmentation Visualizations. Afterward, we provide scene segmentation results on the kitchen datasets in Figure 23. Unlike previous methods that merge all objects into the background (uORF) or yield cluttered outcomes (QBO and COLF), uOCF consistently yields high-fidelity segmentation results.

Novel View Synthesis Visualizations. Finally, we offer more qualitative results for novel view synthesis in Figures 24 and 25. Our method produces much better results than previous methods regarding visual quality.

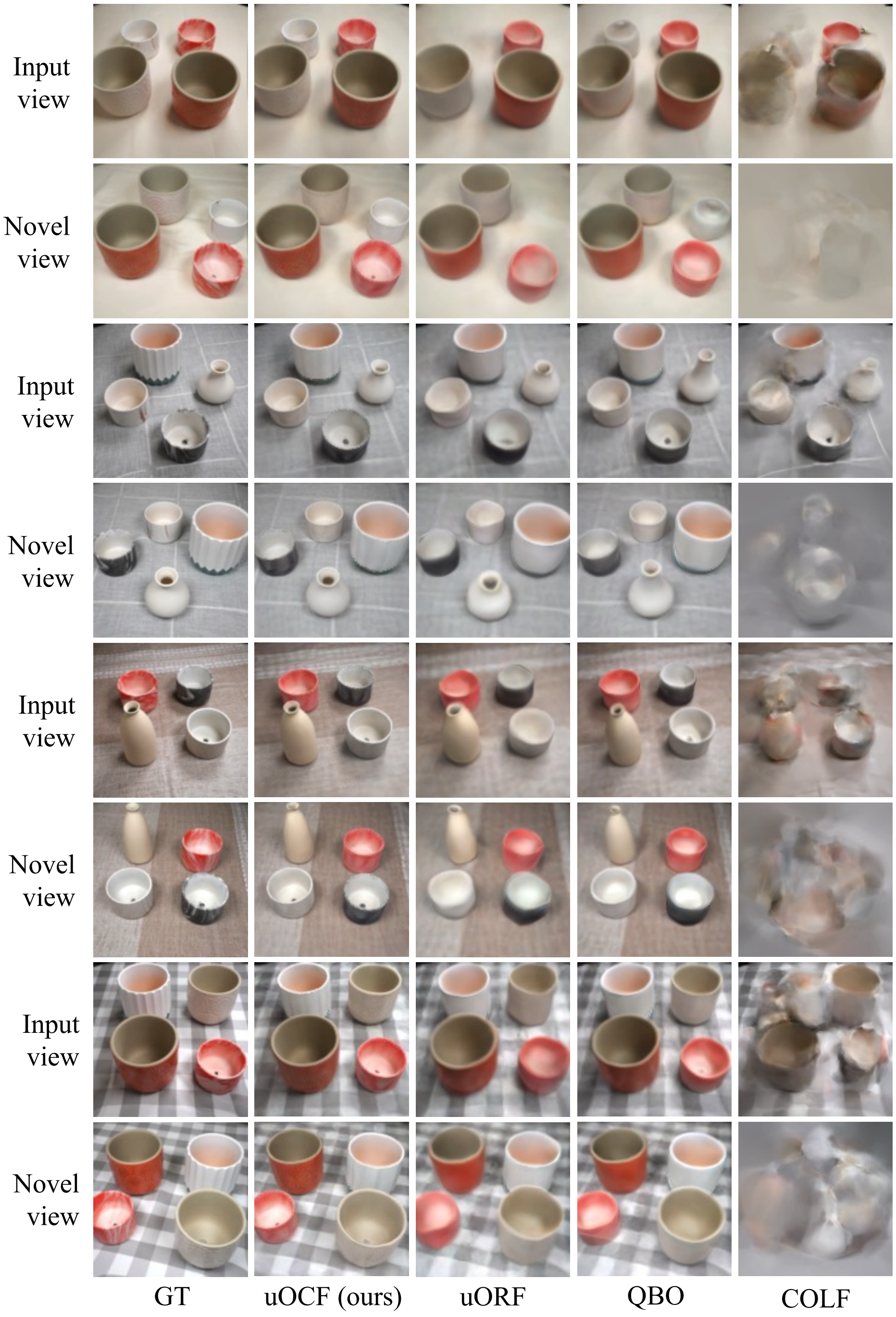


Figure 20: Qualitative comparison results on the Planters dataset.

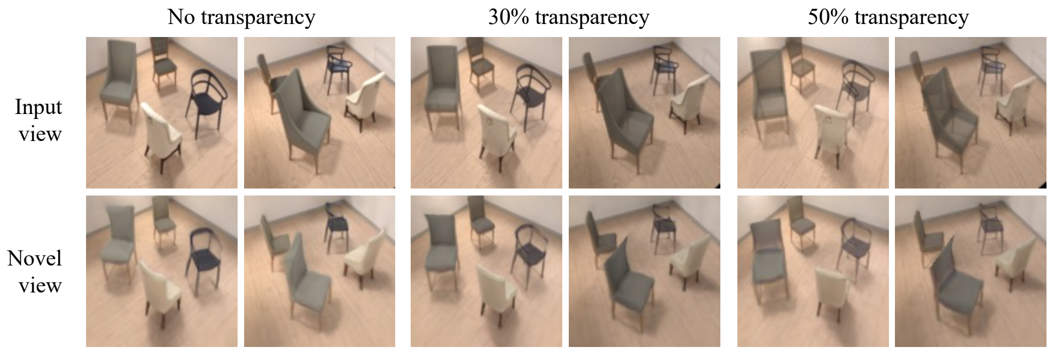


Figure 21: Reconstruction results of uOCF for transparent objects.

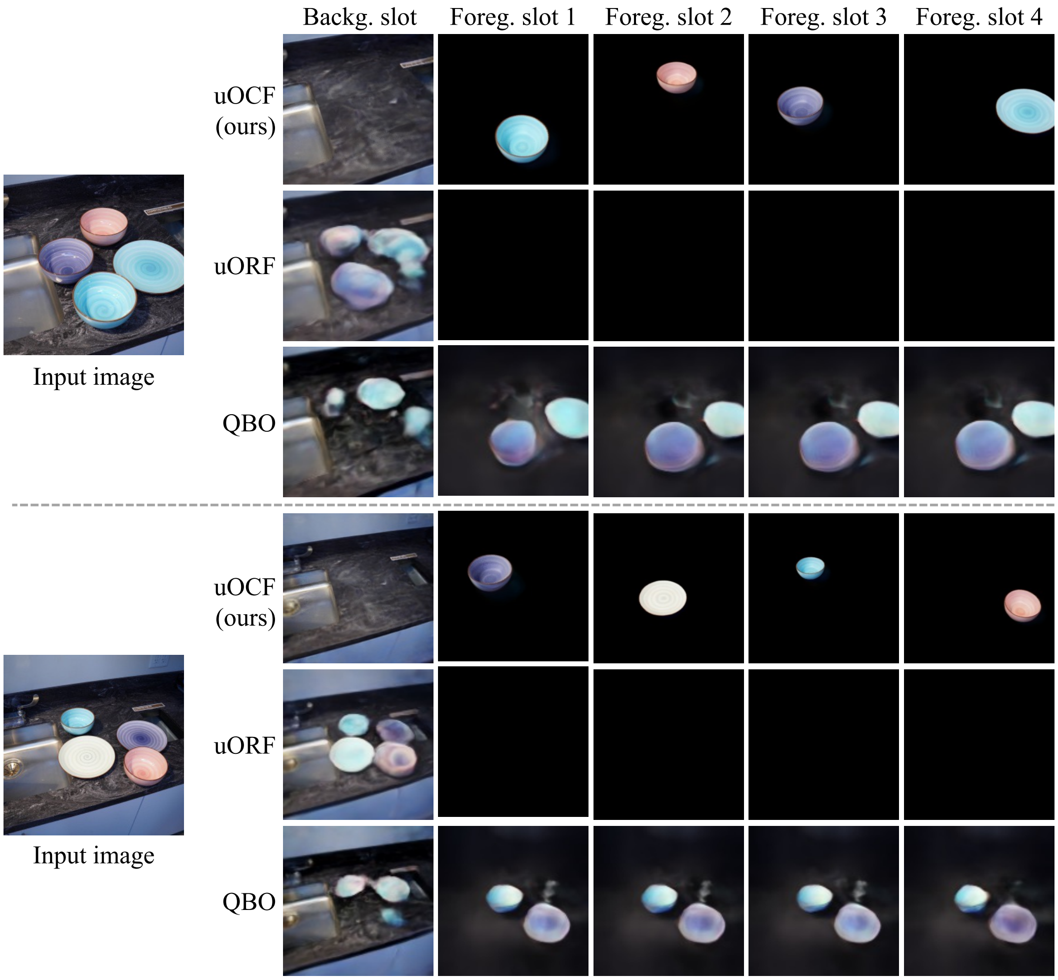


Figure 22: Single slot reconstruction comparison results on Kitchen-Hard.

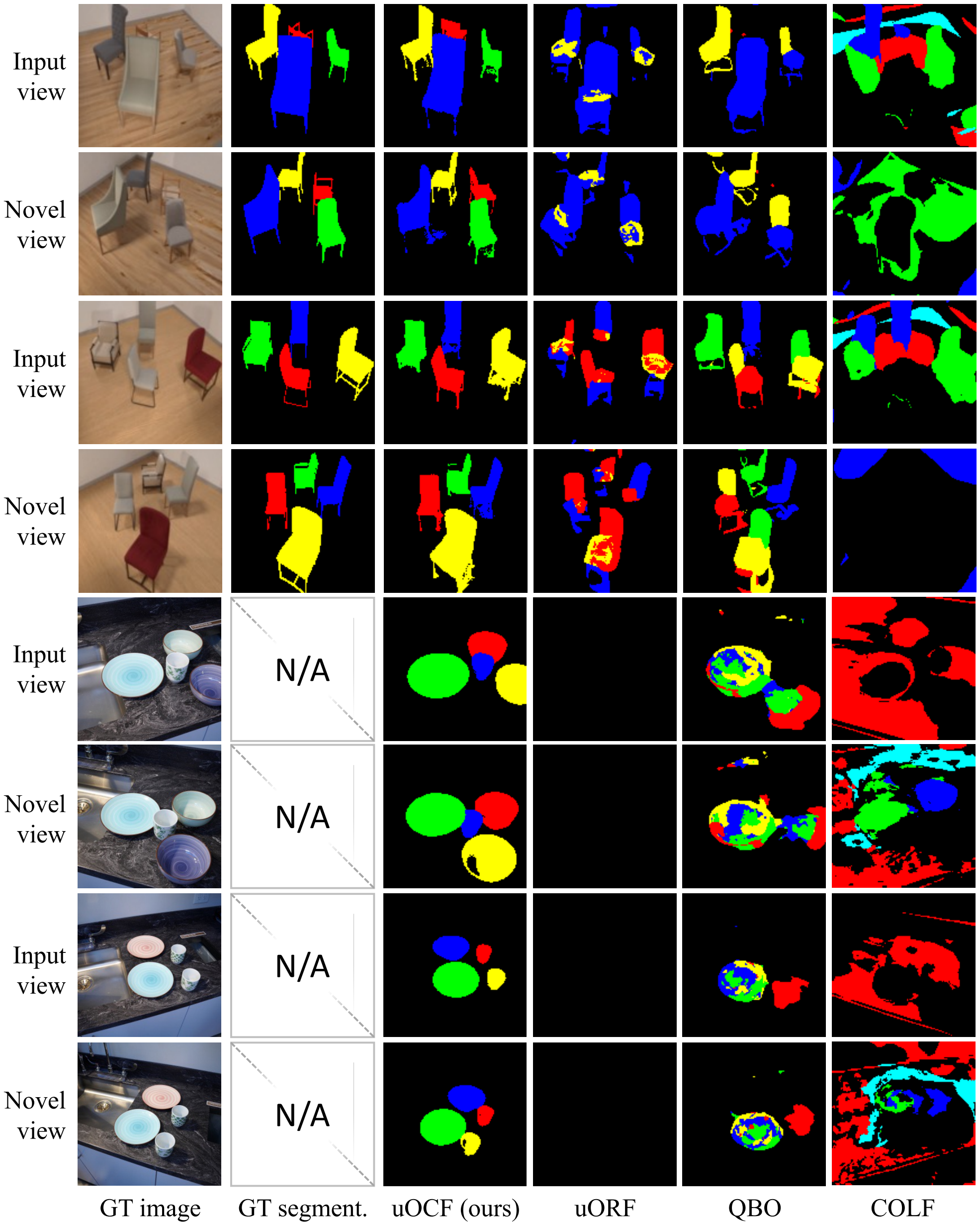


Figure 23: Scene segmentation comparison results on the Room-Texture and Kitchen-Hard datasets.

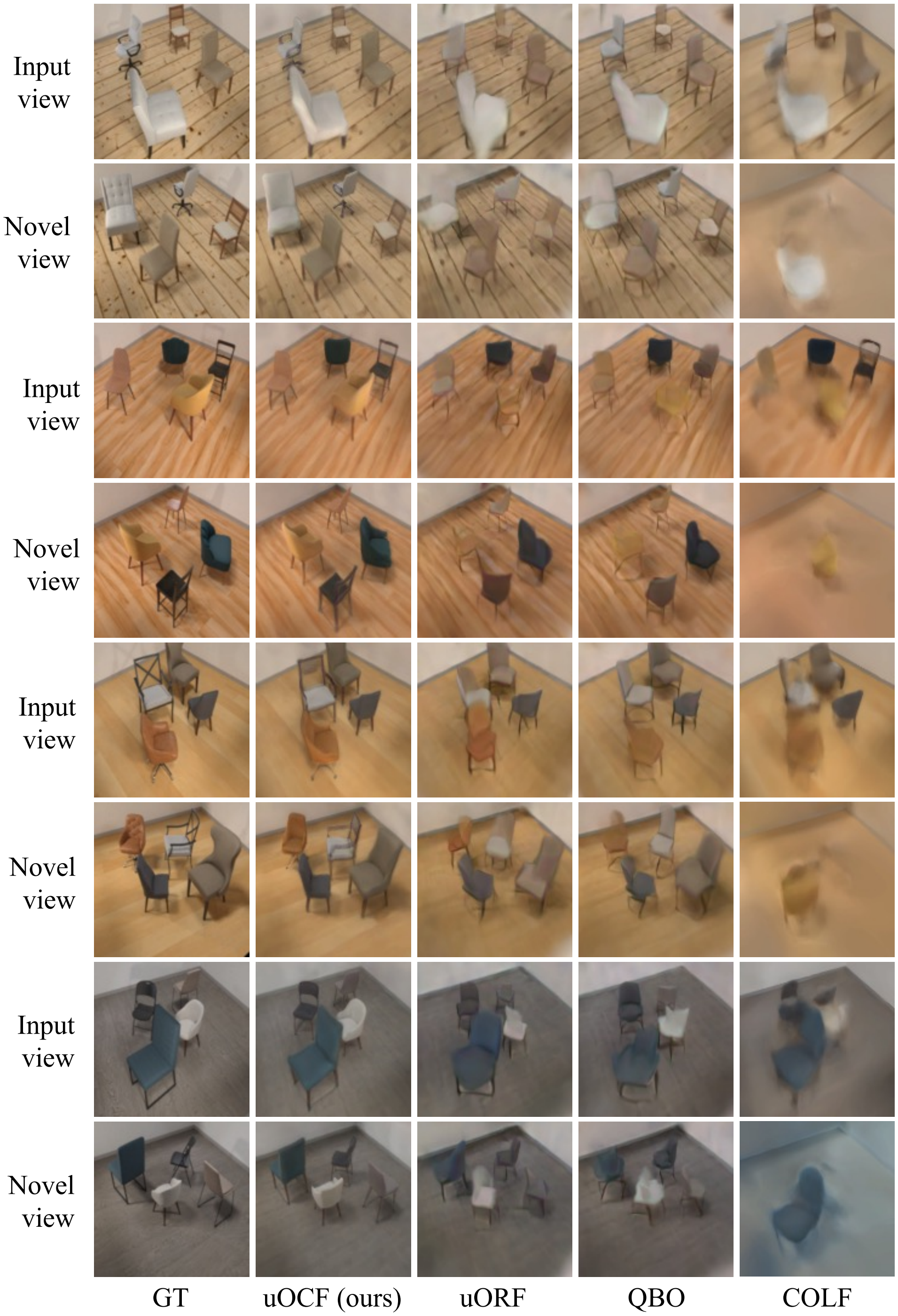


Figure 24: Additional qualitative comparison results on the Room-Texture dataset.

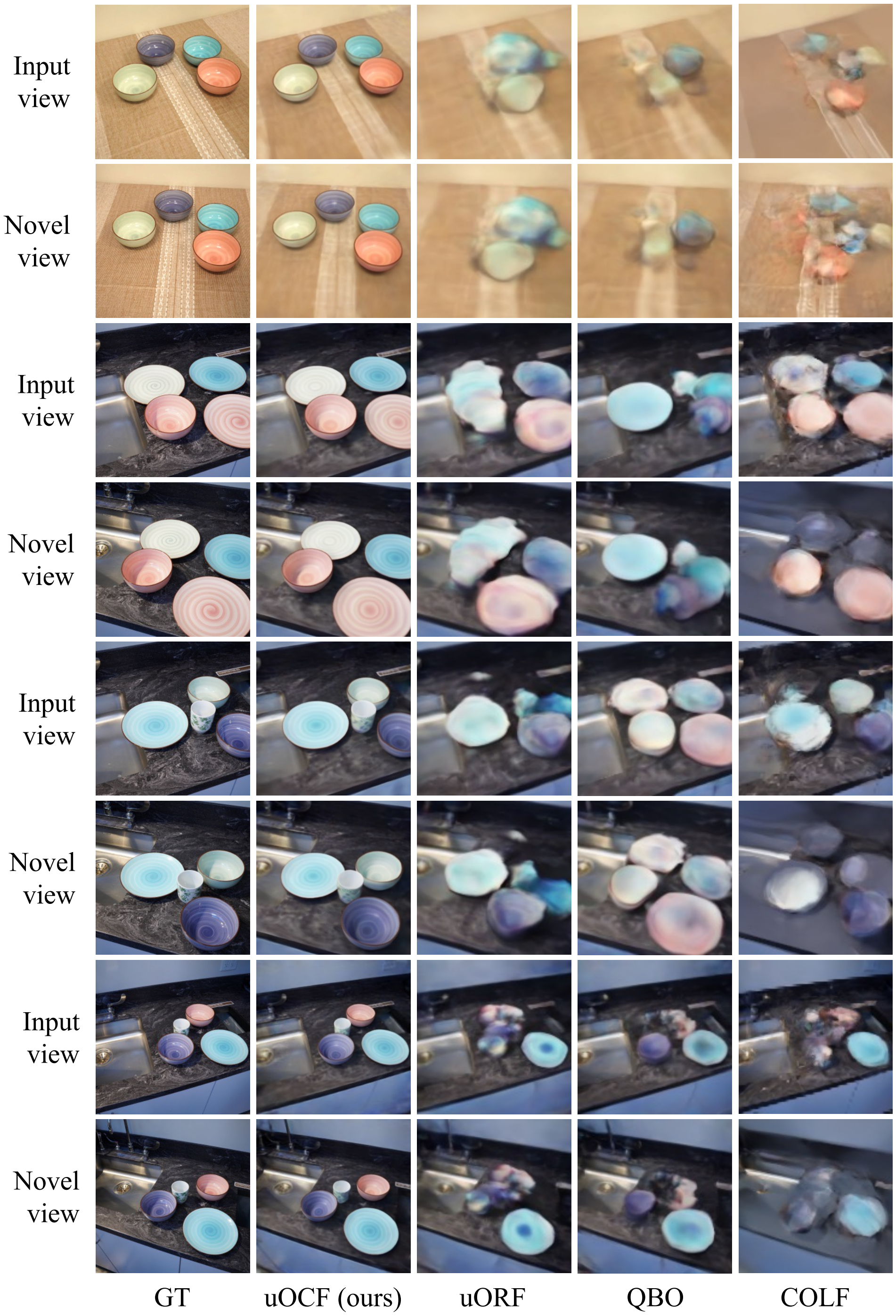


Figure 25: Additional qualitative comparison results on the Kitchen-Hard dataset.