Supplemental Material for Unsupervised Discovery and Composition of Object Light Fields

Anonymous authors
Paper under double-blind review

Abstract

In this supplementary document, we first describe details on architecture in Sec. 1, dataset construction in Sec. 2, and training protocol in Sec. 3. We then provide additional algorithm pseudocode in Sec. 4 and qualitative results in Sec. 5.

1 Architecture

1.1 Encoder

1.1.1 Feature Extractor

We use the same encoder as (Yu et al., 2021) — 6 convolutional layers with bilinear upsampling applied to the last 3. Each layer has kernel size 3, padding of 1, and stride of 1 except for the second and third layers which use 2. Pixel coordinates are normalized to the range of [-1,1] in both directions, leading to 4 additional input channels to concatenate with the 3 image color channels.

1.1.2 Background-Aware Slot Encoder

We use the same slot encoder parameters as (Yu et al., 2021) — a slot dimension of 128 and and 3 iterations of slot competition.

1.2 Decoder

1.2.1 Light Field Networks

For both the foreground and background light fields, we implement them as MLP's of 2 hidden layers with 256 hidden features and ReLU activations. They yield ray-features of dimensionality 64.

1.2.2 Color, Depth, and Visibility Networks

The color generator and depth decoder map the ray feature from an object-LFN into color and depth, and the visibility network assigns visibility weight based on the LFN's decoded depth and relative depth; all three networks are implemented as MLPs of 3 hidden layers with 128 hidden features and ReLU activations.

1.2.3 HyperNetwork

We map the slot latent codes of dimensionality 128 to the weights of each light field MLP via a hypernetwork (Ha et al., 2017), as performed by (Sitzmann et al., 2019; 2020; 2021). The hypernetwork is implemented as an MLP with 1 hidden layer which accepts a 128-dimensional slot latent code and predicts all the weights of the corresponding light field.

2 Dataset Details

2.1 Room-Scenes (Clevr-567, Room-Chair, Room-Diverse)

We refer the reader to the supplement of (Yu et al., 2021) as they produced these datasets and detail their creation there.

2.2 City-Block

The city block is constructed with one road and several buildings, obtained from an online 3D-assets website. The camera is always facing forward in the middle of the lane, with height sampled from a range of near the ground to slightly above the car height, and depth in the scene sampled from near the front to the end of the block. Context views are always captured at the same distance from each row of cars. The training scenes always consist of two rows of cars with the back row placed a fixed distance from the front row. The cars in each row have a small difference in position.

3 Training

We optimize our models' parameters using the ADAM solver with learning rate of 5×10^{-5}. We report the training supervision schedules for each dataset evaluation below.

3.1 CLEVR-567

We train first at a resolution of 64x64 for 150k iterations and at a resolution of 128x128 for 60k iterations. We supervise with a combination of $L1$,$L2$, and perceptual loss (Zhang et al., 2018).

3.2 Room-Chair and Room-Diverse

We initialize the model with the weights of the CLEVR-567 model and train for 128k iterations at 64x64, then for 90k iterations at 128x128 resolution. We supervise with a combination of $L1$,$L2$, and perceptual loss (Zhang et al., 2018).

3.3 Multi-Lane Highway

We initialize the model with the weights of the CLEVR-567 model. We then train with the $L2$ reconstruction loss at a resolution of 64x64 for 145k iterations.

3.4 City Block

We pretrain the static background network on the city block dataset at 64x64 resolution for 800k iterations with $L2$ loss. Then, foreground slots are introduced, initialized with weights from the CLEVR-567 model, and trained for 60k iterations at 64x64 resolution and 50k iterations at 128x128 resolution, supervised with $L2$ loss.

4 Background-Aware Slot Encoder Pseudocode

The pseudocode for (Yu et al., 2021)'s background-aware modification to the slot attention algorithm is presented below at the courtesy of its authors:

Algorithm 1: Object-centric latent inference with background-aware slot attention.

```
Input: $\texttt{feat} \in \mathbb{R}^{N \times D}$
Learnable: $\mu^b, \sigma^b, \mu^f, \sigma^f$: prior parameters, $k, q^b, q^f, v^b, v^f$: linear mappings, $\texttt{GRU}^b$, $\texttt{GRU}^f$, $\texttt{MLP}^b$, $\texttt{MLP}^f$

    $\texttt{slot}^b \sim \mathcal{N}^b \in \mathbb{R}^{1 \times D}$
    $\texttt{slots}^f \sim \mathcal{N}^f \in \mathbb{R}^{K \times D}$
    for $t = 1, \cdots, T$ do
        $\texttt{slot\_prev}^b = \texttt{slot}^b, \quad \texttt{slots\_prev}^f = \texttt{slots}^f$
        $\texttt{attn} = \texttt{Softmax}\left(\frac{1}{\sqrt{D}} k(\texttt{feat}) \cdot \begin{bmatrix} q^b(\texttt{slot}^b) \\ q^f(\texttt{slots}^f) \end{bmatrix}^T, \texttt{dim='slot'}\right)$
        $\texttt{attn}^b = \texttt{attn[0]}, \quad \texttt{attn}^f = \texttt{attn[1:end]}$
        $\texttt{updates}^b$= WeightedMean(weights=$\texttt{attn}^b$, values=$v^b$(inputs))
        $\texttt{updates}^f$= WeightedMean(weights=$\texttt{attn}^f$, values=$v^f$(inputs))
        $\texttt{slot}^b = \texttt{GRU}^b$(state=$\texttt{slot\_prev}^b$, inputs=$\texttt{updates}^b$)
        $\texttt{slots}^f = \texttt{GRU}^f$(state=$\texttt{slots\_prev}^f$, inputs=$\texttt{updates}^f$)
        $\texttt{slot}^b += \texttt{MLP}^b(\texttt{slot}^b)$,
        $\texttt{slots}^f += \texttt{MLP}^f(\texttt{slots}^f)$
    end
    return  $\texttt{slot}^b$, $\texttt{slots}^f$
```

5 Additional Results

5.1 Comparison to Baseline on Unbounded Scene

We evaluate the baseline model (Yu et al., 2021) on the unbounded City Block scene and illustrate a sample in Fig. 1. Employing a volumetric decoder, the model is forced by its memory constraints and the large bounds of the scene to sample coarsely between the near and distant far plane. As a result of their coarse sampling, their model is unable to learn to render any foreground elements of the scene.

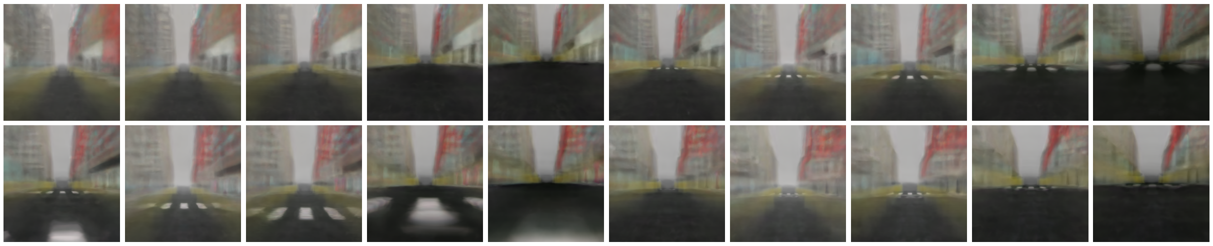


Figure 1: The baseline model's (Yu et al., 2021) reconstruction on the City Block scene. Using a volumetric decoder, the model is unable to render smaller foreground objects due to its coarse sampling through the long scene.

5.2 Additional Scene Manipulation Results

We perform additional scene manipulation demonstrations in Fig. 2. In section 5.3 we describe the limitations of our representation's support for object manipulation.

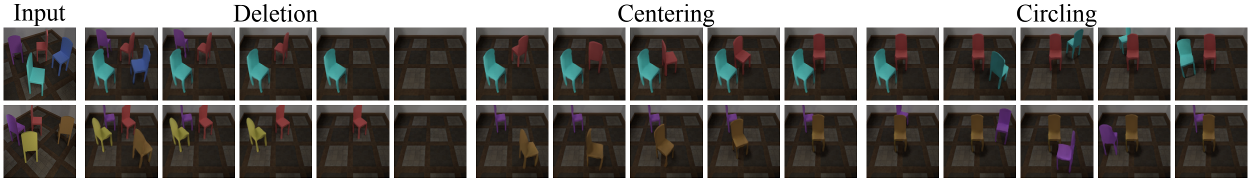


Figure 2: We demonstrate object-level scene editing tasks of object deletion, centering, and circling, on scenes from the chairs dataset.

5.3 Scene Editing Limitation

As our model reasons on the space of camera rays, instead of 3D points as in Yu et al. (2021), scene editing manipulations which move the camera beyond the space of observed camera rays during training may result in view-inconsistent behavior. That is, our volumetric predecessor uORF (Yu et al., 2021) demonstrates scene editing with perfect generalization due to the inherent multiview consistency of volumetric rendering, whereas our light-field representation does not afford this guarantee. We demonstrate such an object manipulation generating camera rays outside the training distribution and yielding view-inconsistent behavior in Fig 3.

Therefore, for scene-editing applications, volumetric representations may be preferable in the setting where the distribution of novel query viewpoints is significantly different than the distribution of training viewpoints.

5.4 Out-of-Distribution Novel View Synthesis

We render views from camera viewpoints outside of the training distribution of the room-scenes dataset in Fig 4. This generalization ability is due to the fact that our method operates on camera rays instead of camera positions, and so these out-of-distribution camera viewpoints generate rays which are mostly still in the training distribution. Camera positions which generate rays outside of the training distribution, such as lowering the camera to the floor level, have no such generalization guarantee.

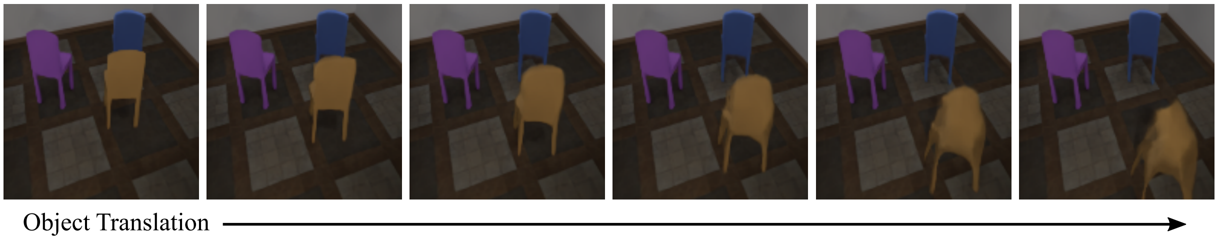


Figure 3: Our scene editing capabilities are limited to object manipulations which remain near the distribution of observed camera rays during training, and editing beyond this range may result in view-inconsistent predictions.

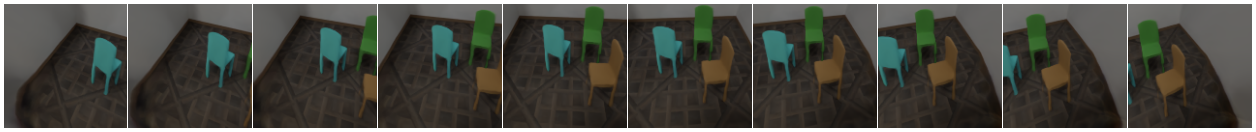


Figure 4: We render from camera viewpoints outside of the training distribution.

5.5 Complex Occlusion Rendering

We conduct an experiment to more rigorously evaluate the ability of our model to reason about and render complex occlusions — specifically, rendering which cannot be solved by naive "ordering" of the objects. We render out a small dataset of interlocked rings, and supervise our model with ground-truth segmentation. Novel views and segmentation is shown in Fig 5, demonstrating the ability of our model to render non-trivial occlusions.

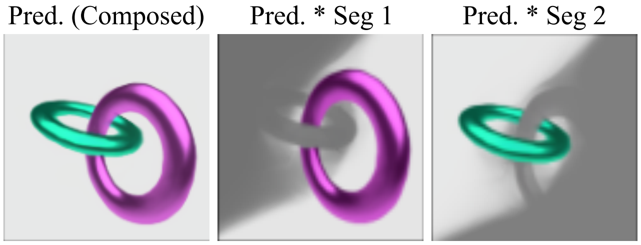


Figure 5: We demonstrate the ability of our model to render complex occlusions by rendering interlocked rings, where a simple ordering of the objects is insufficient to represent the scene. We show the render composited as well as multiplied by the predicted per-slot masks.

5.6 Additional Decomposition and Novel View Synthesis Results

We provide additional novel view synthesis results and decomposition on four scenes from each room dataset in Fig. 6 and the composited city block scene in Fig. 7.

References

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. In *Proc. ICLR*, 2017.

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, pp. 1121–1132, 2019.

Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020.

Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In *Proc. NeurIPS*, 2021.

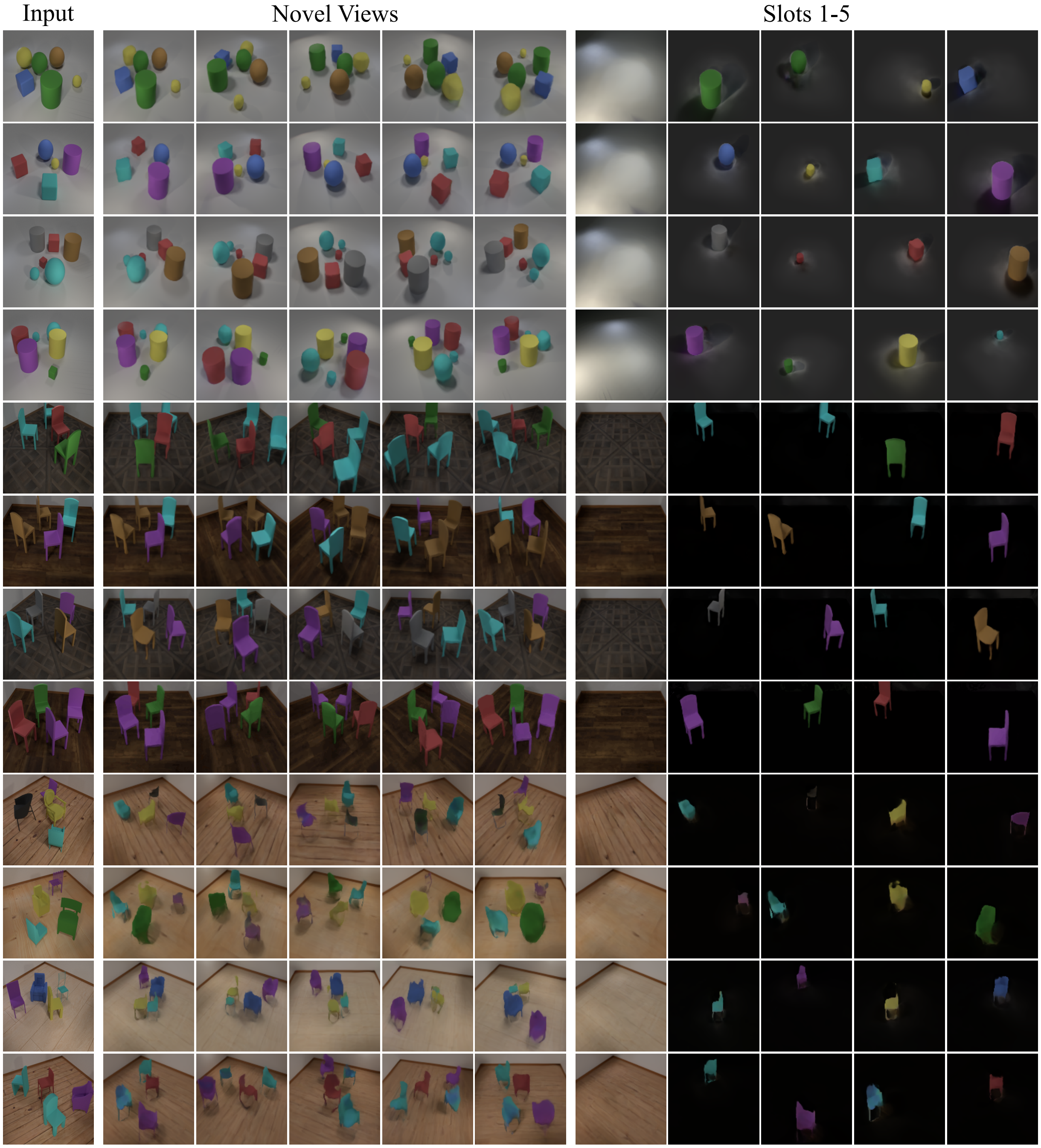
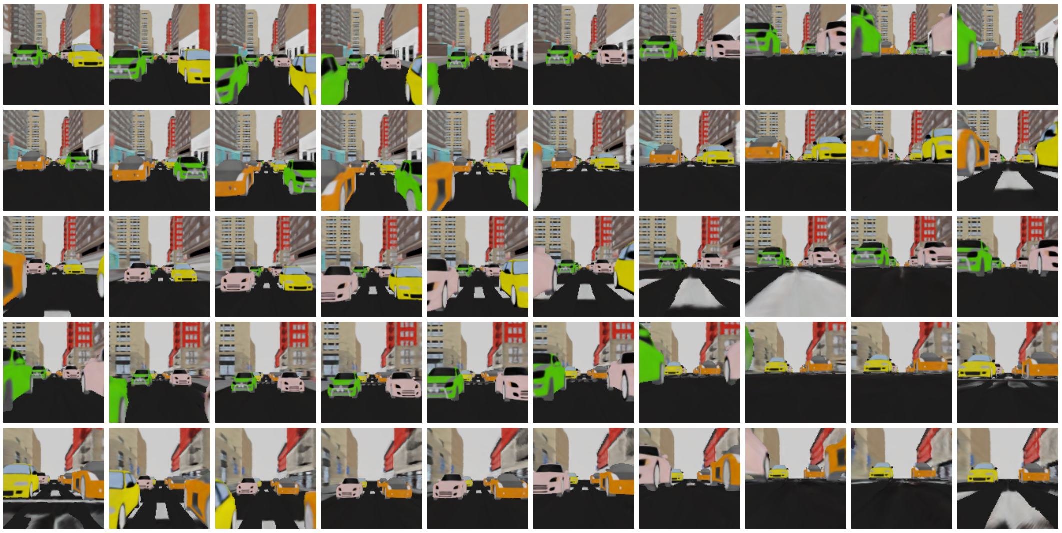


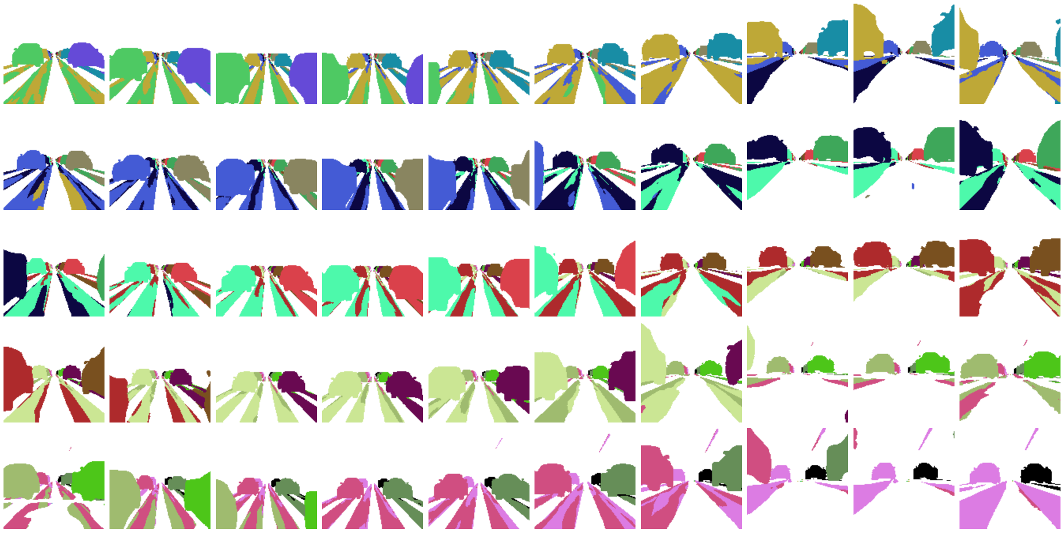
Figure 6: We demonstrate additional qualitative results for the room-scene datasets on tasks of novel view synthesis and scene decomposition. We show five novel views at the scene level (middle) and five slots from the first novel view (right).

Hong-Xing Yu, Leonidas J Guibas, and Jiajun Wu. Unsupervised discovery of object radiance fields. *arXiv preprint arXiv:2107.07905*, 2021.

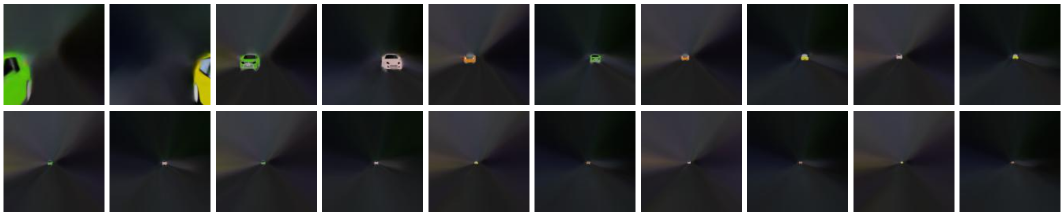
Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.



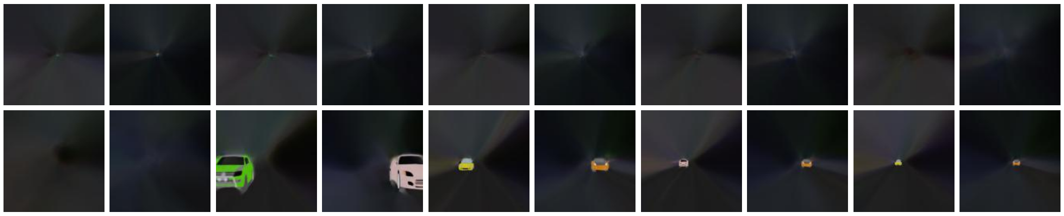
(a) Novel views of composite scene (left to right, top to bottom)



(b) Novel segmentation from composite scene (left to right, top to bottom)



(c) Object light fields at frame 4



(d) Object light fields at frame 35

Figure 7: We render a sequence of novel views from the cross-scene composition application described (first), as well as each view's segmentation (second), and per-object contributions from two frames (third, fourth).