Supplementary Material for Unsupervised Discovery of Object Radiance Fields

Hong-Xing Yu **Leonidas J. Guibas** **Jiajun Wu**
Stanford University

Abstract

In this **supplementary document**, we first provide implementation details on unsupervised discovery of Object Radiance Fields in Section 1. We then describe details on datasets and baseline architectures in Section 2. We further show additional results in Section 3. All mathematical and algorithmic notations are the same as those in the main manuscript. In the **supplementary video**, we provide an overview of our paper.

1 Implementation

In this section we provide implementation details of our unsupervised discovery of Object Radiance Fields. Each following subsection corresponds to that in the main manuscript.

1.1 Object-centric Encoding

Convolutional feature extraction. Our convolutional encoder is a simple U-net. We show our encoder architecture in Table 1 and Table 2. In our experiments we assume fixed camera focal length. In this case, the ray direction does not provide additional information to the pixel coordinates, and thus we drop the ray direction input and only feed pixel coordinates as input channels in addition to the input RGB image. Each of the XY pixel coordinates is normalized to $[-1, 1]$ in both directions, leading to 4 additional channels.

Background-aware slot attention. We show a pseudo code of background-aware slot attention in Algorithm 1. We encourage readers to compare it with the original slot attention algorithm [4] for better understanding. For CLEVR-567 dataset we set $D = 40$ and $K = 8$. For Room-Chair and Room-Diverse we set $D = 64$ and $K = 5$.

1.2 Compositional Neural Rendering

Coordinate space. We represent foreground objects in the viewer space. Regarding background, since it is difficult to estimate full geometry from a single view (e.g., the geometry behind the camera), we assume fixed background geometry and represent it in the world space. Incorporating multi-view images as inference input might solve this problem [7], but we leave it as future exploration. This design also encourages the disentanglement between foreground objects and background by preventing the background slot from decoding foreground objects, because the positional information provided in the encoder is represented in viewer space. To further encourage the disentanglement, we add a locality constraint during early training. Considering that "foreground" objects should be largely visible in sight, we set a foreground box and enforce that every foreground-querying point outside the box has zero density. The foreground box is defined such that its projection in image space can engage roughly 90% pixels.

Decoder architecture. We show our foreground decoder architecture in Figure 1.

Preprint. Under review.

Layer name	Input shape	Output shape	Stride	Note
`Conv1`	64×64×7	64×64×64	2	Skip to `Conv6`
`Conv2`	64×64×64	32×32×64	2	Skip to `Conv5`
`Conv3`	32×32×64	16×16×64	2	
`Conv4`	16×16×64	16×16×64	1	
`Upsample`	16×16×64	32×32×64		Bilinear upsampling
`Conv5`	32×32×128	32×32×64	1	
`Upsample`	32×32×64	64×64×64		Bilinear upsampling
`Conv6`	64×64×128	64×64×64	1	

Table 1: Encoder architecture for the CLEVR-567 dataset and the Room-Chair dataset. All convolutional kernel sizes are 3×3. All activation functions for convolutional layers are ReLU.

Layer name	Input shape	Output shape	Stride	Note
`Conv0`	128×128×7	128×128×64	1	
`Conv1`	128×128×64	64×64×64	2	Skip to `Conv6`
`Conv2`	64×64×64	32×32×64	2	Skip to `Conv5`
`Conv3`	32×32×64	16×16×64	2	
`Conv4`	16×16×64	16×16×64	1	
`Upsample`	16×16×64	32×32×64		Bilinear upsampling
`Conv5`	32×32×128	32×32×64	1	
`Upsample`	32×32×64	64×64×64		Bilinear upsampling
`Conv6`	64×64×128	64×64×64	1	

Table 2: Encoder architecture for the Room-Diverse dataset. All convolutional kernel sizes are 3×3. All activation functions for convolutional layers are ReLU.

1.3 Model Learning

Loss functions. We set $\lambda_{\text{percept}} = 0.006$, $\lambda_{\text{adv}} = 0.01$, $\lambda_R = 10$. For perceptual loss, we implement the feature extractor p by using the output of the 4-th convolutional block in a VGG16 [6] pretrained on ImageNet. For the adversarial discriminator, we follow the architecture of StyleGAN2 [3] with slight modification such that the maximum channel number is 128. We also use the lazy R1 regularization [3]. We use Adam optimizer for discriminator with learning rate 0.001, $\beta_1 = 0$ and $\beta_2 = 0.9$. The adversarial loss is incorporated after 100k iterations. Since shape uncertainty only appears in the Room-Diverse dataset, we only impose the adversarial loss on the Room-Diverse dataset but not on CLEVR-567 or Room-Chair. Both perceptual loss and adversarial loss are added after the first 100k iterations.

Coarse-to-fine progressive training. For coarse training, we bilinearly downsample supervision images to 64×64. The coarse training lasts for 600K iterations. For fine training, we randomly crop 64×64 patches from 128×128 images. The fine training lasts for 600K iterations. Our model is trained on a single Nvidia RTX 3090 GPU for about 6 days. For all networks except discriminator, we use Adam optimizer with learning rate 0.0003, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Learning rate is exponentially decreased by half for every 200K iterations until after 600K iterations. We also adopt the learning rate warm-up from the slot attention paper [4] for the first 1K iterations. We initialize decoder networks with Xavier's initialization. In each batch, we input one image and neurally render 4 images for supervision. We render each pixel with 64 samples.

2 Experiments

In this section we provide further details on experiment settings.

2.1 Data

CLEVR-567. In the CLEVR-567 dataset, each object's shape is randomly chosen from three geometric primitives (i.e., cylinder, cube and sphere). The color is randomly chosen from

Algorithm 1: Background-aware slot attention module.

Input: $\texttt{inputs} \in \mathbb{R}^{N \times D}, \texttt{slot}^b \sim \mathcal{N}^b \in \mathbb{R}^{1 \times D}, \texttt{slots}^f \sim \mathcal{N}^f \in \mathbb{R}^{K \times D}$

Layer Params: k, q^b, q^f, v^b, v^f: linear mappings, $\texttt{GRU}^b$, $\texttt{GRU}^f$, $\texttt{MLP}^b$, $\texttt{MLP}^f$, $\texttt{LayerNorm}$ (x5)

$\texttt{inputs} = \texttt{LayerNorm}(\texttt{inputs})$

for $t = 1, \cdots, T$

$\texttt{slot\_prev}^b = \texttt{slot}^b, \quad \texttt{slots\_prev}^f = \texttt{slots}^f$

$\texttt{slot}^b = \texttt{LayerNorm}(\texttt{slot}^b), \quad \texttt{slots}^f = \texttt{LayerNorm}(\texttt{slots}^f)$

$\texttt{attn} = \texttt{Softmax}\left(\frac{1}{\sqrt{D}} k(\texttt{inputs}) \cdot \begin{bmatrix} q^b(\texttt{slot}^b) \\ q^f(\texttt{slots}^f) \end{bmatrix}^T, \texttt{dim='slot'}\right)$

$\texttt{attn}^b = \texttt{attn[:, 0]}, \quad \texttt{attn}^f = \texttt{attn[:, 1:end]}$

$\texttt{updates}^b = \texttt{WeightedMean}(\texttt{weights=attn}^b, \ \texttt{values=}v^b\texttt{(inputs)})$

$\texttt{updates}^f = \texttt{WeightedMean}(\texttt{weights=attn}^f, \ \texttt{values=}v^f\texttt{(inputs)})$

$\texttt{slot}^b = \texttt{GRU}^b(\texttt{state=slot\_prev}^b, \ \texttt{inputs=updates}^b)$

$\texttt{slots}^f = \texttt{GRU}^f(\texttt{state=slots\_prev}^f, \ \texttt{inputs=updates}^f)$

$\texttt{slot}^b += \texttt{MLP}^b(\texttt{LayerNorm}(\texttt{slot}^b)), \quad \texttt{slots}^f += \texttt{MLP}^f(\texttt{LayerNorm}(\texttt{slots}^f))$

return $\texttt{slot}^b, \texttt{slots}^f$

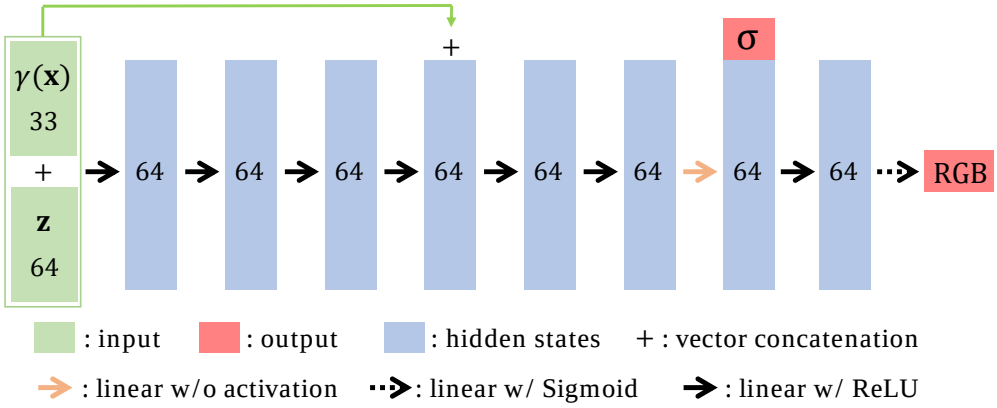

Figure 1: Illustration for foreground decoder architecture. We follow the architecture in NeRF [5] but with fewer parameters to decrease space demand. We set the highest positional embedding frequency to 5, so that the positional embedding input dimension is $5 \times 2 \times 3 + 3 = 33$. The background decoder is slightly different in that it does not have the second last layer and third last layer. Density σ is activated by ReLU. Since estimating specularity from a single image is intrinsically ambiguous, we assume Lambertian surfaces and do not use the ray direction as input.

$\{\texttt{red}, \texttt{blue}, \texttt{purple}, \texttt{gray}, \texttt{cyan}, \texttt{yellow}, \texttt{green}, \texttt{brown}\}$. There are two possible sizes for each object. When rendering images, we use the same camera intrinsic as original CLEVR dataset [2]. We do not use the visibility check due to our 360 degree multi-view setting, so we increase elevation angle by $\pi/15$ to increase the chance of object visibility. Rendering setting is the same for all datasets.

Room-Chair. For the object shape we use a chair model* from ShapeNet [1]. We use the same material and colors as CLEVR-567.

Room-Diverse. All object shapes are randomly chosen from 1,200 ShapeNet chairs. For each shape, we normalize it into a unit cube according to vertex coordinates. We also use 8 colors $\{\texttt{red}, \texttt{blue}, \texttt{purple}, \texttt{gray}, \texttt{cyan}, \texttt{yellow}, \texttt{green}, \texttt{black}\}$ with diffuse material. Since shape uncertainty only appears in this dataset, we only impose the adversarial loss on this dataset.

*Model ID: 3ffd794e5100258483bc207d8a5912e3

2.2 Baseline Architectures

Slot attention. We use the encoder-decoder architecture in the slot attention paper [4] used for object discovery experiments on the CLEVR dataset. Basically it has 6 convolutional layers for encoder and 6 convolution-transpose layers for decoder. The number of channels for each layer is 64. All models are trained on 128×128 images.

NeRF-AE. We follow NeRF implementation without view direction as input and set the highest frequency to 5. The encoder is similar to ours in Figure 2, but the basic number of channels is increased from 64 to 256 (and thus the number of channels of inputs to `Conv5` and `Conv6` is 512). The number of slot is set to 1.

3 Additional Results

In this section we show additional qualitative results for our experiments in the main manuscript. We show additional examples for 3D scene segmentation in Figure 2 and Figure 3, for novel view synthesis in Figure 4, Figure 5 and Figure 6, for scene manipulation in Figure 7 and Figure 8, for evaluating losses in Figure 9, for generalization to challenging spatial arrangement in Figure 10 (note that in the packed-CLEVR-11 dataset we only use a single size for higher object visibility), and for generalization to unseen object appearance in Figure 11.

References

[1] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 3

[2] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017. 3

[3] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 2

[4] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *arXiv preprint arXiv:2006.15055*, 2020. 1, 2, 4

[5] 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. *arXiv preprint arXiv:2003.08934*, 2020. 3

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[7] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. *arXiv preprint arXiv:2012.02190*, 2020. 1

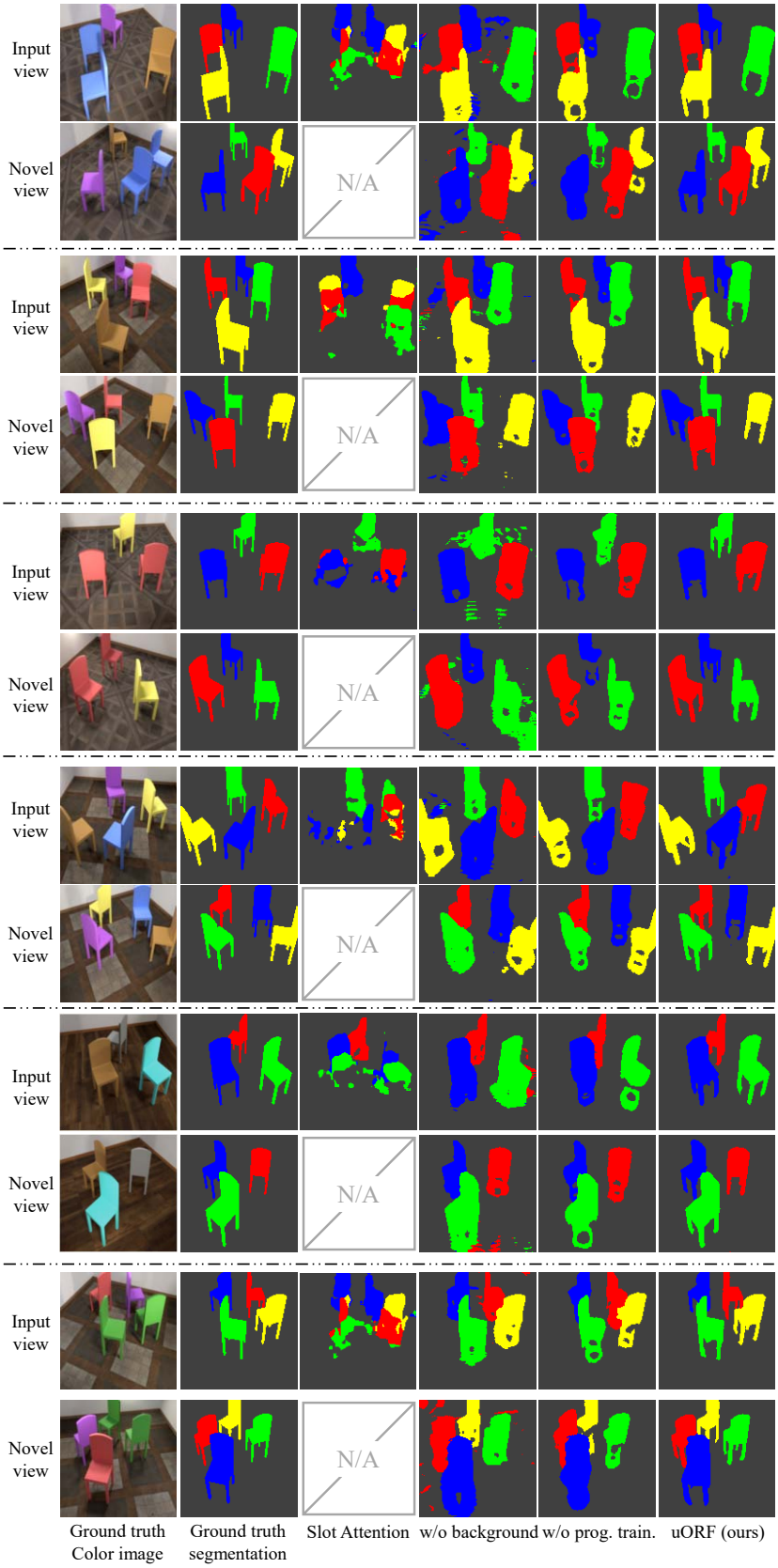

Figure 2: Additional qualitative results for 3D segmentation on Room-Chair dataset.

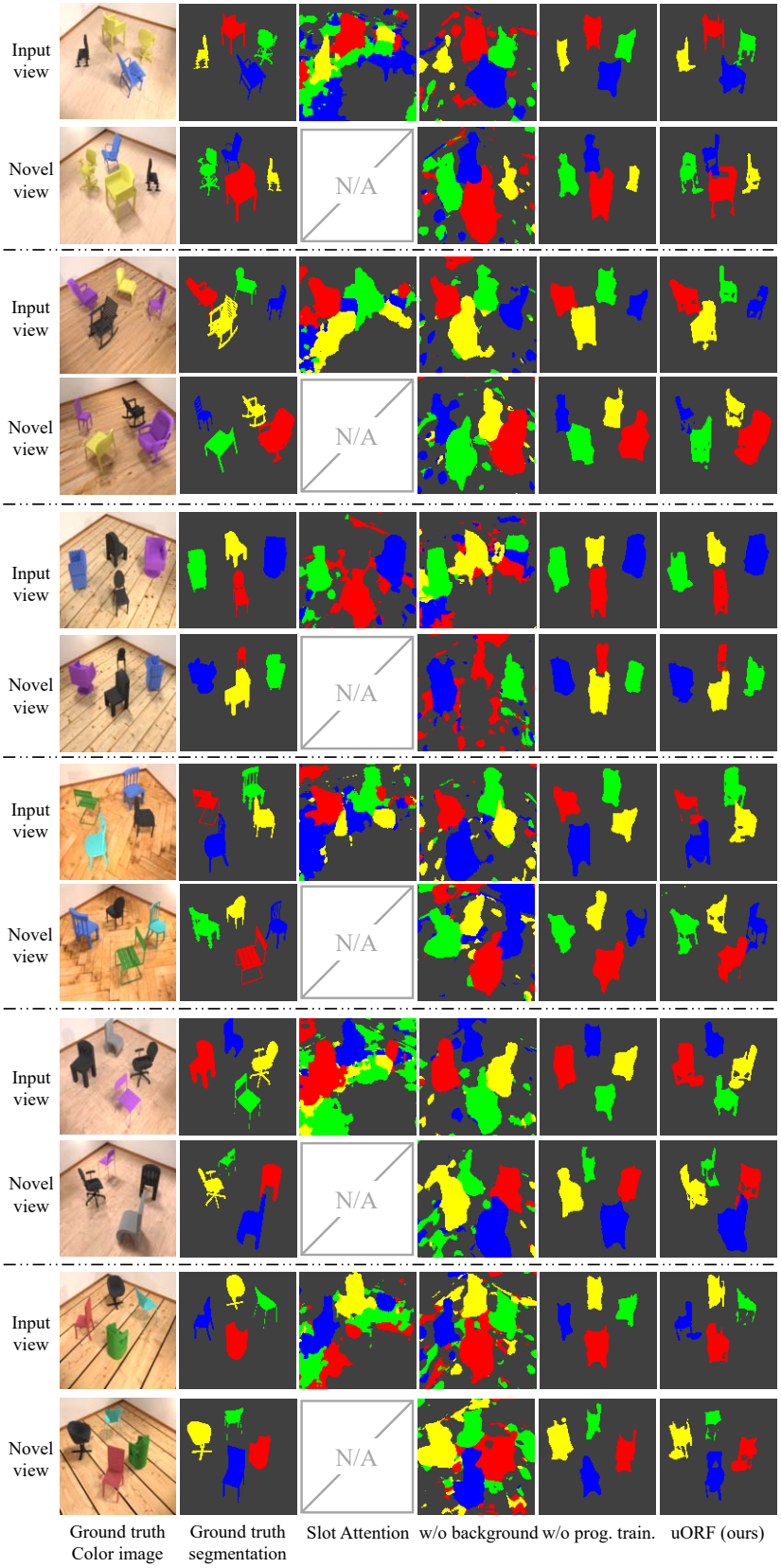

Figure 3: Additional qualitative results for 3D segmentation on Room-Diverse dataset.

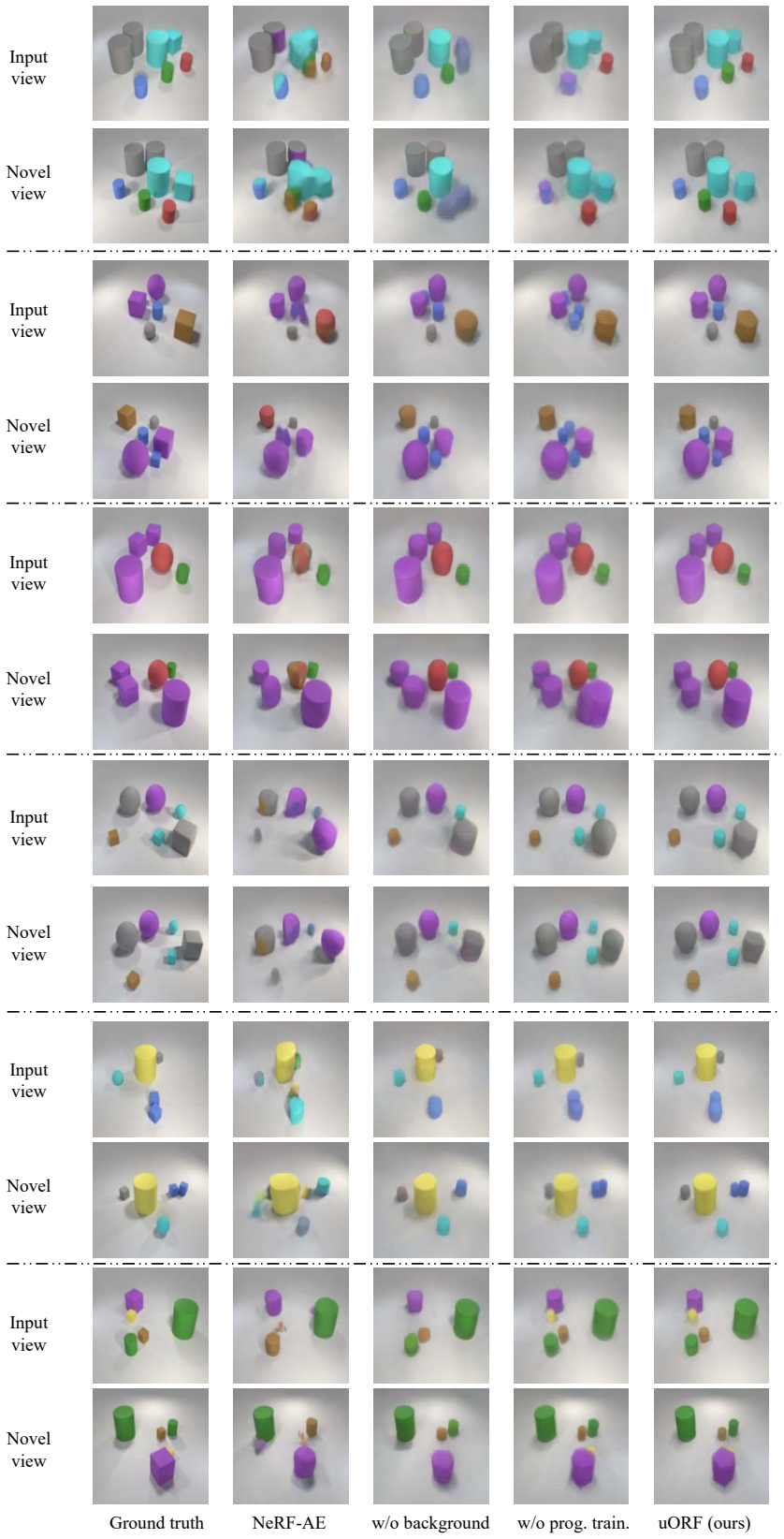

Figure 4: Additional qualitative results for novel view synthesis on CLEVR-567 dataset.

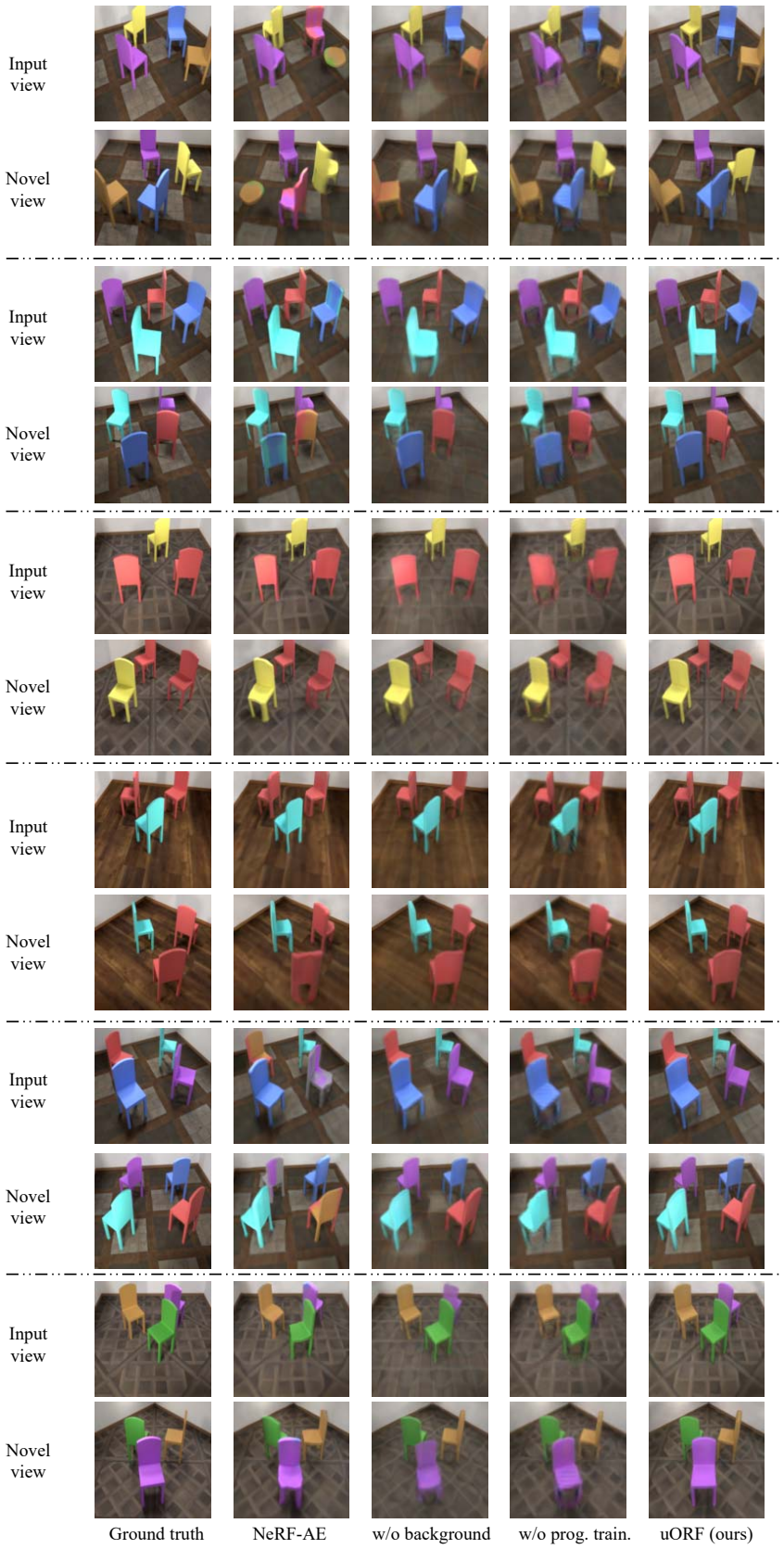

Figure 5: Additional qualitative results for novel view synthesis on Room-Chair dataset.

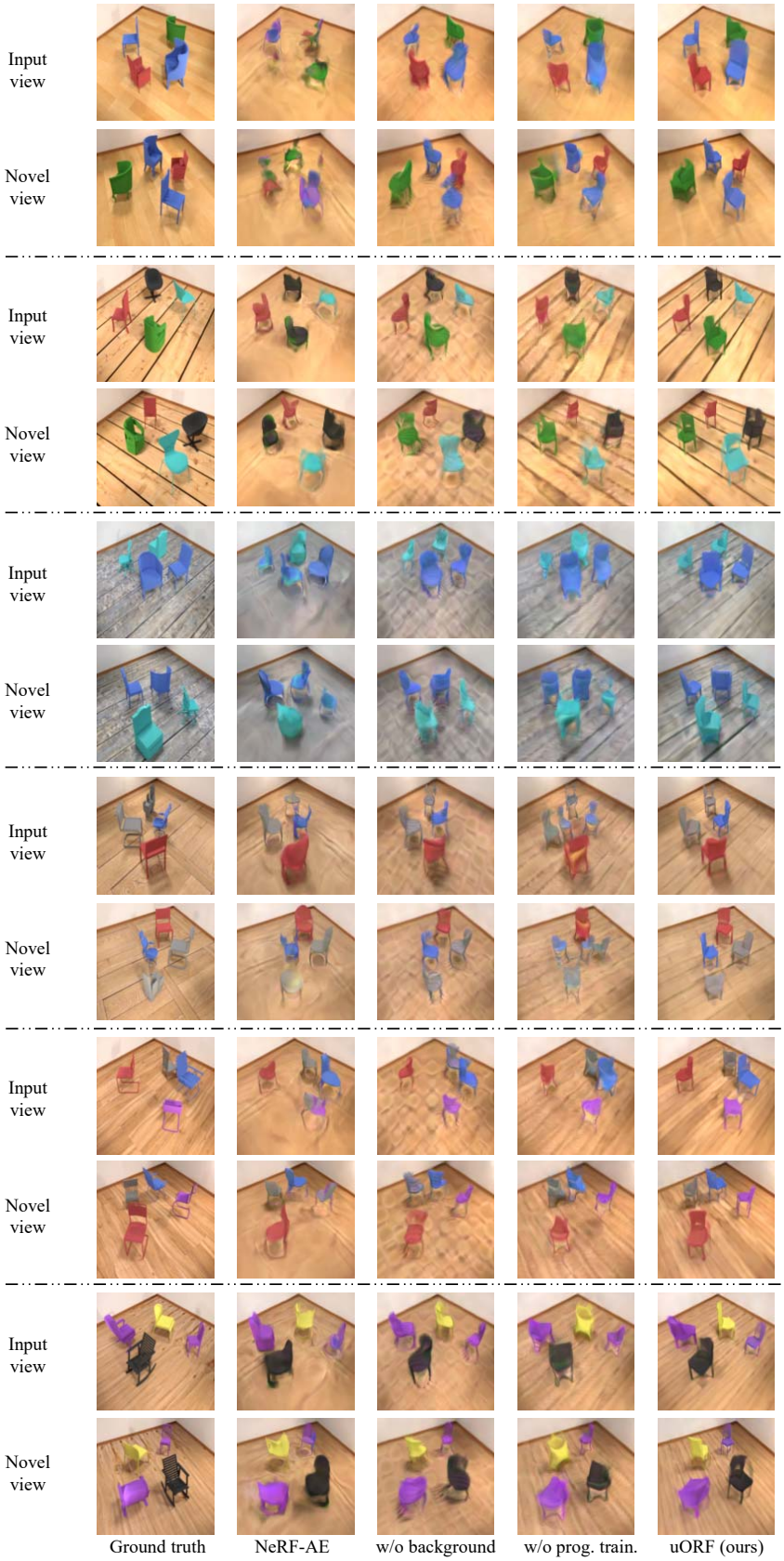

Figure 6: Additional qualitative results for novel view synthesis on Room-Diverse dataset.

Figure 7: Additional qualitative results for scene manipulation.

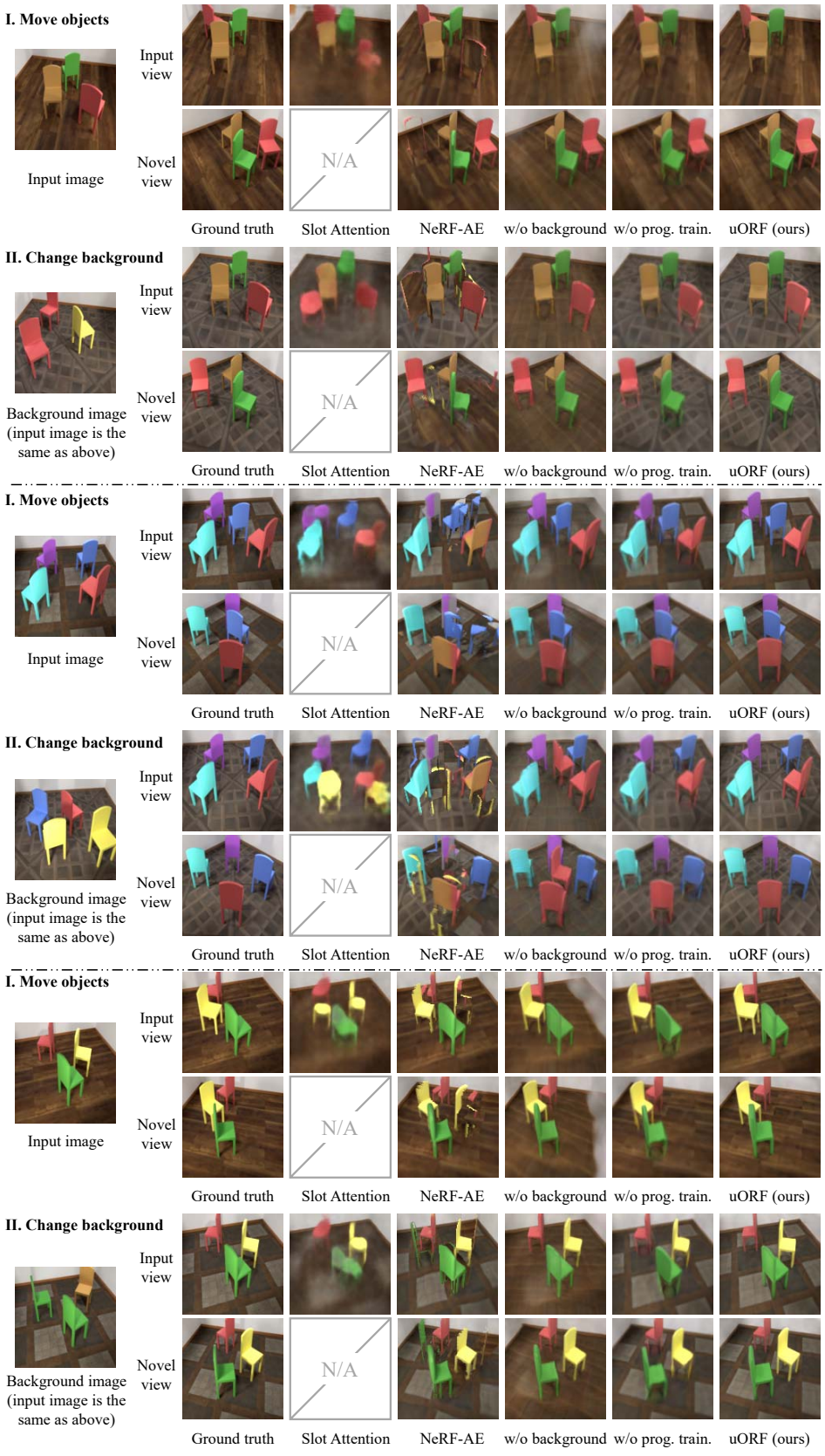

Figure 8: Additional qualitative results for scene manipulation.

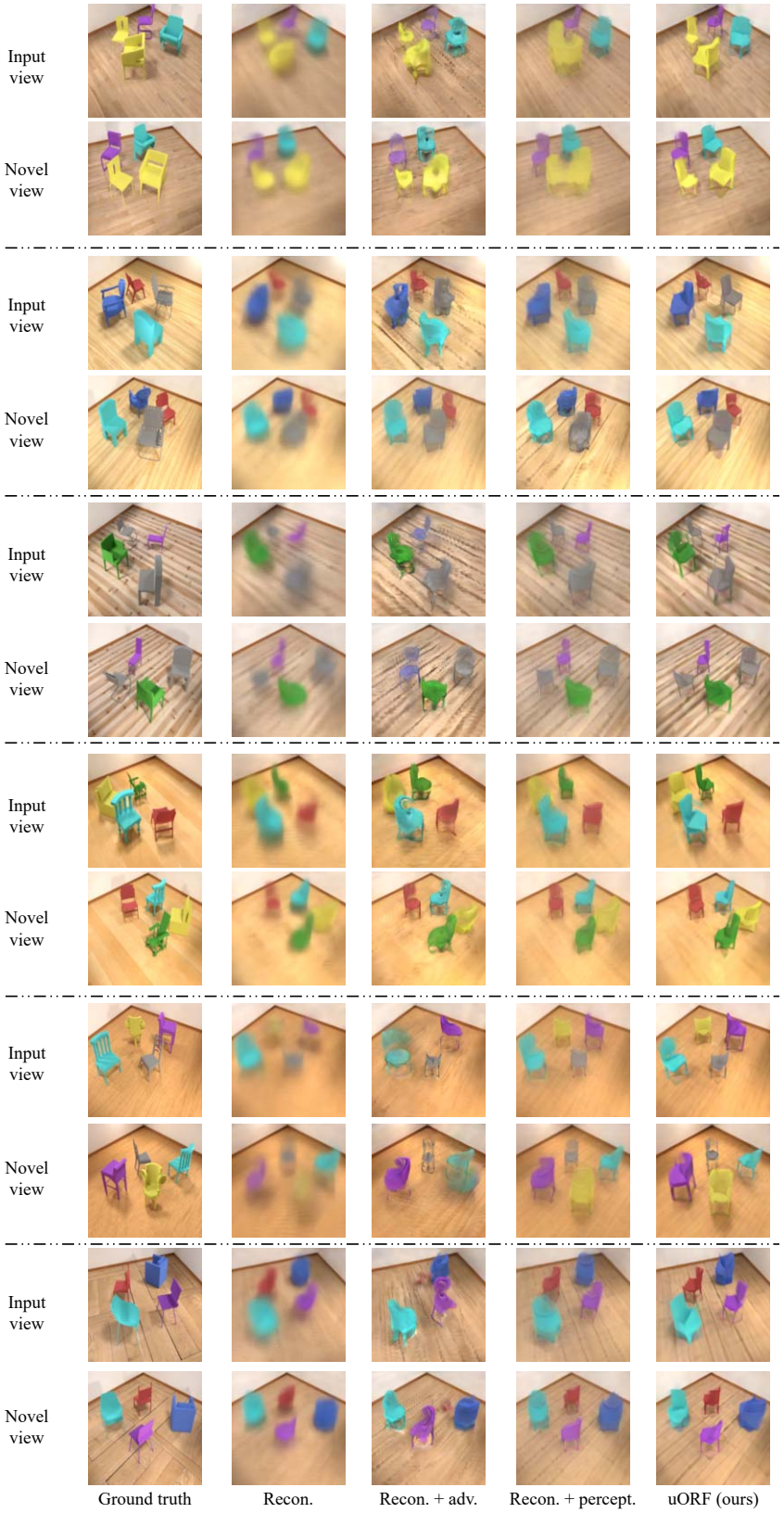

Figure 9: Qualitative results for loss evaluations. Using both perceptual loss and adversarial loss improves image quality.

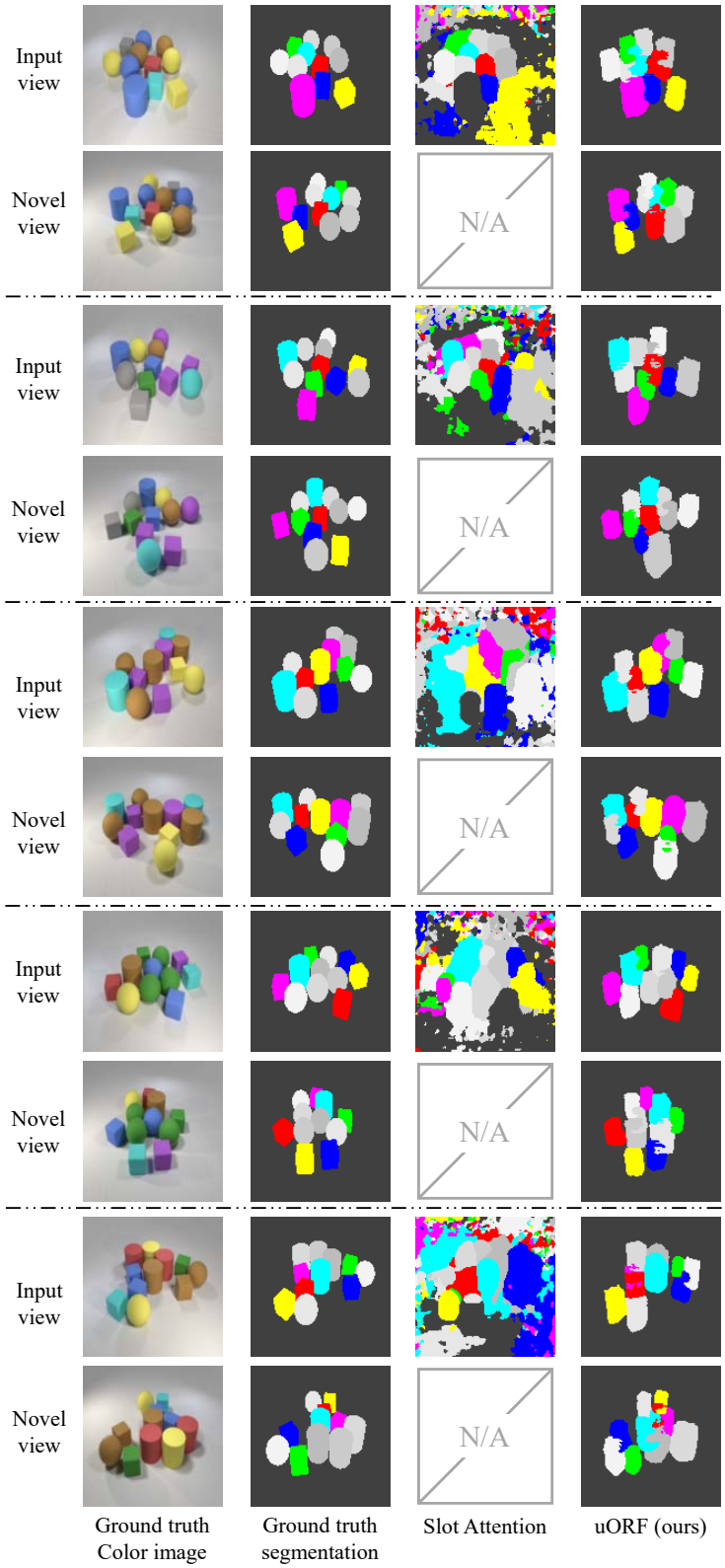

Figure 10: Qualitative results for generalization to unseen spatial arrangement.

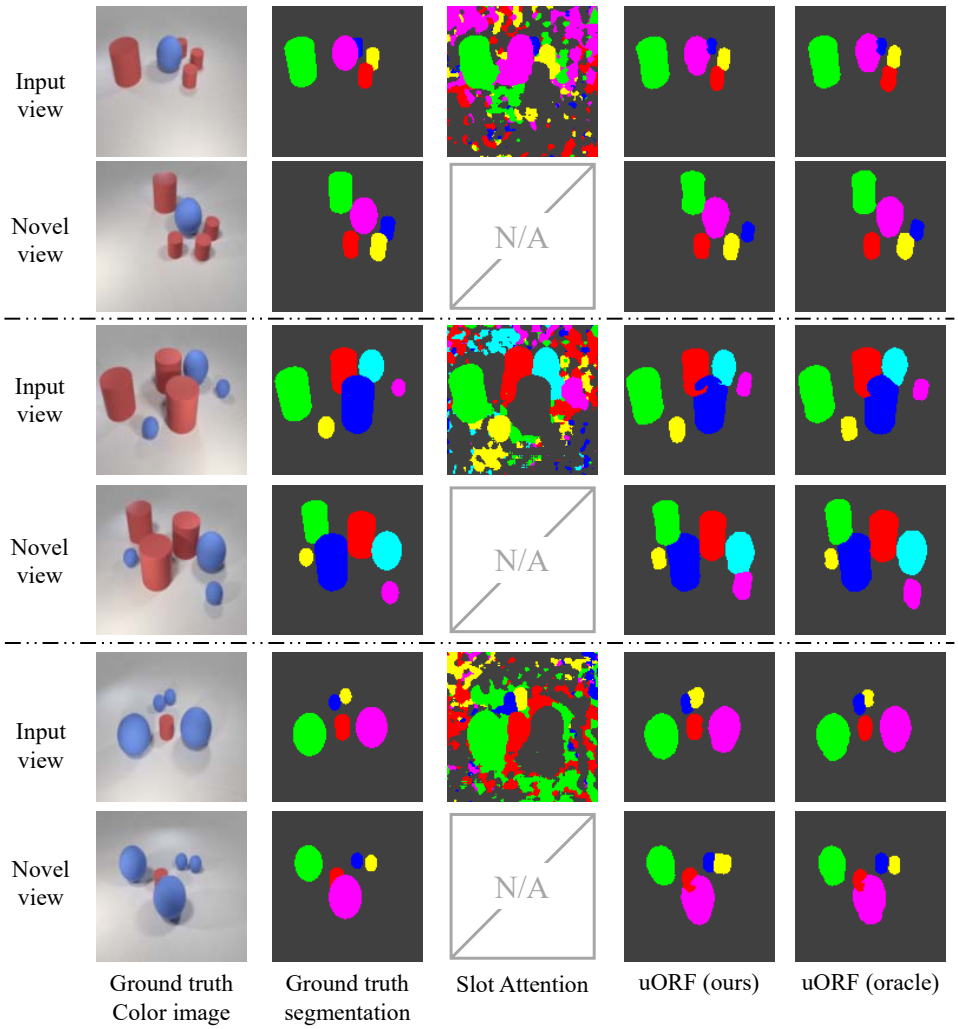

Figure 11: Qualitative results for generalization to unseen combination of color and shape.