MOVIS: Enhancing Multi-Object Novel View Synthesis for Indoor Scenes

Supplementary Material

A. Model details

A.1. DINO patch feature and camera view embedding

The original image encoder of Stable Diffusion is CLIP, which excels at aligning images with text. Other image encoders like DINO-v2 [14] or ConvNeXtv2 [22] may provide denser image features that may benefit generation tasks as mentioned by previous works [7, 9]. Therefore, we opt to use the DINO feature instead of the original CLIP feature in our network following [7]. To inject the DINO patch feature into our network, we encode the input view image using DINO-v2 [14] "norm patchtokens", whose shape dimension is $(b, 16, 16, 1024)$. We will simply flatten it into $(b, 256, 1024)$ to apply cross-attention, and b means batch size here.

As for the camera view embedding, we choose to embed it using a 6 degrees of freedom (6DoF) representation. To be specific, let E_i be the extrinsic matrix under the input view and E_j be the extrinsic matrix under the output view, we represent relative camera pose change as $E_i^{-1}E_j$. We will also flatten it into 16 dimensions to concatenate it to the image feature. Afterwards, we will replicate the 16-dimension embedding 256 times to concatenate the embedding to every channel of the DINO feature map. A projection layer will later be employed to project the feature map into $(b, 256, 768)$ to match the dimension of the CLIP encoder, which was originally used by Stable Diffusion so that we can fine-tune the pre-trained checkpoint. It is worth noting that we also tried other novel view synthesizer's camera embedding like Zero-1-to-3 [11] using a 3DoF spherical coordinates in early experiments, but we found that it does not make much of a difference.

A.2. Depth and mask condition

In this section, we will explain how input view depth and mask are incorporated as additional conditioning inputs. For depth maps, regions with infinite depth values are assigned a value equal to twice the maximum finite depth value in the rest of the image. After this adjustment, we apply a normalization technique to scale the depth values to the range of $[-1, 1]$, enabling the use of the same VAE architecture as for images.

For mask images, we assign unique values to different object instances in the input view. For instance, if there are four objects in the multi-object composite, they will be labeled as 1, 2, 3, and 4, respectively, while the background will be assigned a value of 0. The same normalization technique used for depth maps is applied to these mask images. These mask images, like all other inputs, are processed by the VAE, with all images set to a resolution of 256×256.

A.3. Supervision for auxiliary mask prediction task

To implement the auxiliary mask prediction task, we encode the output view mask images into the same latent space as the input view mask images. Object instances viewed from different angles will be assigned the same value, which is ensured during the curation of our compositional dataset. Supervision is directly applied to the latent mask features extracted from the final layer of the denoising U-Net. Only the input view mask images are required during inference, simplifying the process while preserving consistency across views.

A.4. Timestep scheduler

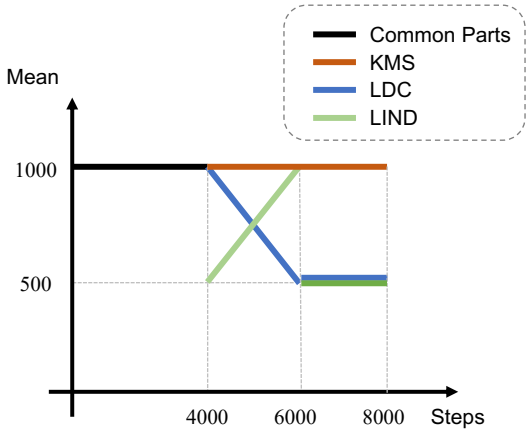

Figure S.1. **Illustration of different timestep sampling strategies.**

Table S.1. **Ablation on different strategies.** Incorporating sampling strategies significantly improves the model performance, while the linear decline (LDC) achieves the best.

Dataset	Method	Novel View Synthesis		
		PSNR(↑)	SSIM(↑)	LPIPS(↓)
C3DFS	w/o sch.	16.166	0.808	0.212
	KMS	17.148	0.823	0.175
	LIND	17.279	0.824	0.172
	LDC	**17.432**	**0.825**	**0.171**

Though we finally employed a linearly declining strategy, we experimented with several alternatives. Specifically, we tested linearly declining the mean of the Gaussian

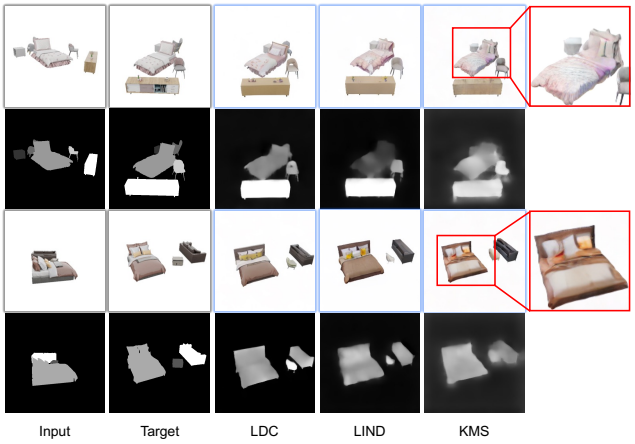

Figure S.2. **Comparison of different strategies**. The predicted images and mask images under novel views using different strategies are visualized. We can observe that images predicted by the KMS strategy possess weird and blurry color while LDC strategy seems to be slightly better than LIND.

distribution (LDC), linearly increasing the mean after a sudden drop (LIND), and keeping the mean constant (KMS). These strategies are illustrated in Fig. S.1. The metrics on the test set of our C3DFS are provided in Tab. S.1, with some visual comparisons in Fig. S.2. *w/o sch.* in Tab. S.1 refers to applying a uniform sampler, same as the one in the main paper. From the results, we observe that LDC achieves slightly better performance than LIND and KMS, largely outperforming *w/o sch.*

However, we observed significant visual artifacts such as weird colors and extremely blurry mask images when combining the auxiliary mask prediction task with the KMS sampling strategy, as demonstrated in Fig. S.2. For example, the bed in the second example possesses unclear object boundaries and distorted texture. We believe this is due to KMS focusing primarily on denoising at larger timesteps, which provides limited guidance for recovering mask images and refining fine-grained geometry and appearance. Consequently, without a dedicated period for denoising smaller timesteps, the per-object shape and appearance appear distorted and unrealistic.

B. Experiment Details

B.1. Implementation Details

We solely utilize the data from C3DFS as the training source for our model. The training process takes around 2 days on 8 NVIDIA A100 (80G) GPUs, employing a batch size of 172 per GPU. The exact training steps are 8,000 steps. During the inference process, we apply 50 DDIM steps and set the guidance scale to 3.0. We use DepthFM [6] and SAM [8] to extract the depth maps and object masks when they are not available, as well as for all real-world images.

B.2. Datasets

C3DFS We use the furniture models from the 3D-FUTURE dataset [5] to create our synthetic multi-object compositional data. The 3D-FUTURE dataset contains 9,992 detailed 3D furniture models with high-resolution textures and labels. Following previous work [1], we categorize the furniture into seven groups: bed, bookshelf, cabinet, chair, nightstand, sofa, and table. To ensure unbiased evaluation, we further split the furniture into distinct training and test sets, ensuring that none of the test set items are seen during training.

After filtering the furniture, we first determine the number of pieces to include in each composite, which is randomly selected to be between 3 and 6. Next, we establish a probability distribution based on the different types of furniture items and sample each piece according to this distribution. To prevent collisions and penetration between furniture items, we employ a heuristic strategy. Specifically, for each furniture item to be added, we apply a random scale adjustment within the range of $[0.95, 1.05]$, as the inherent scale of the furniture models accurately reflects real-world sizes. We also rotate each model by a random angle to introduce additional variability. Once these adjustments are complete, we begin placing the furniture items in the scene. The first item is positioned at the center of the scene at coordinates $(0, 0, 0)$. Subsequent objects are added one by one, initially placed at the same central location. Since this results in inevitable collisions, we randomly sample a direction and gradually move the newly added item along this vector until there is no intersection between the bounding boxes of the objects. By following these steps, we generate a substantial number of multiple furniture items composites, ultimately creating a training set of 100,000 composites and a test set of 5,000 to evaluate the capabilities of our network.

After placing all the furniture items, we render multi-view images to facilitate training, using Blender [2] as our renderer due to its high-quality output. We first normalize each composite along its longest axis. To simulate real-world camera poses and capture meaningful multi-object compositions, we employ the following method for sampling camera views.

Cameras are randomly sampled using spherical coordinates, with a radius range of $[1.3, 1.7]$ and an elevation angle range of $[2°, 40°]$. There are no constraints on the azimuth angle, allowing the camera to rotate freely around multiple objects. The chosen ranges for the radius and elevation angles are empirical. In addition to determining the camera positions, we establish a "look-at" point to compute the camera pose. This point is randomly selected on a spherical shell with a radius range of $[0.01, 0.2]$.

To enhance the model's compositional structural aware-

Table S.2. **Availability of conditions in different datasets.**

	C3DFS	Room-Texture	Objaverse	SUNRGB-D	3D-FRONT
depth	✓	×	✓	×	×
mask	✓	✓	✓	×	×

ness, we also render depth maps and instance masks (both occluded and unoccluded) from 12 different viewpoints. The unoccluded instance mask ensures that if one object is blocked by another, the complete amodal mask of the occluded object is still provided, regardless of any obstructions. Although these unoccluded instance masks are not currently necessary for our network, we render them for potential future use.

Objaverse To evaluate our network's generalization capability, we create a small dataset comprising 300 composites sourced from Objaverse [3]. Specifically, we utilize the provided LVIS annotations to select categories that are commonly found in indoor environments, such as beds, chairs, sofas, dressers, tables, and others. Since the meshes from Objaverse vary in scale, we rescale each object based on reference object scales from the 3D-FUTURE dataset [5]. The composition and rendering processes follow the same strategy employed in C3DFS.

Algorithm 1 Hit Rate Computation

// Obtain image-matching pairs using MASt3R and save in a list
$\text{Pairs}_{\text{gt}} = \text{MASt3R}(\text{GT})$
$\text{Pairs}_{\text{ours}} = \text{MASt3R}(\text{Ours})$
// Each element in the list is a four-element tuple $\text{p} = (\mathbf{x}^0, \mathbf{y}^0, \mathbf{x}^1, \mathbf{y}^1)$
// $(\mathbf{x}^0, \mathbf{y}^0)$ refers to the point in the input view image and $(\mathbf{x}^1, \mathbf{y}^1)$ the point in output view image
$\text{Hits} = 0$
For p_{gt} **in** Pairs_{gt}
    // p^i_{ours} is the i-th element of $\text{Pairs}_{\text{ours}}$
    // p[:2] refers to the first two element in the tuple and p[2:] the last two
    $i^\star = \arg\min_i(\mathbf{L}_2(\text{p}_{\text{gt}}[:2], \text{p}^i_{\text{ours}}[:2]))$
        IF $\mathbf{L}_2(\text{p}_{\text{gt}}[:2], \text{p}^{i^\star}_{\text{ours}}[:2]) < 20$ **and** $\mathbf{L}_2(\text{p}_{\text{gt}}[2:], \text{p}^{i^\star}_{\text{ours}}[2:]) < 20$
            // Successfully hit one, delete it from gt pairs and ours pairs
            $\text{Hits} \leftarrow \text{Hits} + 1$
            POP$(\text{Pairs}_{\text{ours}}, \text{p}^{i^\star}_{\text{ours}})$
return $\text{Hits}/\text{len}(\text{Pairs}_{\text{gt}})$

Inference Details Since our model requires input-view depth map and mask images as additional inputs, we need to use DepthFM [6] and SAM [8] to extract the depth maps and object masks when they are not available, as well as for all real-world images. We show whether all the used datasets have provided depth maps and mask images in Tab. S.2. '×' means they do not provide such conditions while '✓' means they do provide such conditions.

Algorithm 2 Nearest Matching Distance Computation

// The notations are the same as the one in Algorithm 1
$\text{Pairs}_{\text{gt}} = \text{MASt3R}(\text{GT})$
$\text{Pairs}_{\text{ours}} = \text{MASt3R}(\text{Ours})$
$\text{Dist} = \text{EmptyList}()$
For p_{gt} **in** Pairs_{gt}
    $i^\star = \arg\min_i(\mathbf{L}_2(\text{p}_{\text{gt}}[:2], \text{p}^i_{\text{ours}}[:2]))$
    IF $\mathbf{L}_2(\text{p}_{\text{gt}}[:2], \text{p}^{i^\star}_{\text{ours}}[:2]) < 20$
        Append$(\text{Dist}, \mathbf{L}_2(\text{p}_{\text{gt}}[2:], \text{p}^{i^\star}_{\text{ours}}[2:]))$
        POP$(\text{Pairs}_{\text{ours}}, \text{p}^{i^\star}_{\text{ours}})$
return $\text{Mean}(\text{Dist})$

B.3. Metrics

Intersection over Union (IoU) Since all baseline methods do not possess the concept of every object instance, we compute a foreground-background IoU for comparison. This metric can provide a rough concept of the overall placement alignment with ground truth images. We extract the foreground object mask by converting the generated image to grayscale (I_L). Given that the generated image has a white background, we compute the foreground mask $\mathbf{M}$ as $\mathbf{M} = I_L < \beta_{th}$, where β_{th} is a threshold that is fixed as 250.

Cross-view Matching As outlined in the main paper, we introduce two metrics to systematically evaluate cross-view consistency with the input view: **Hit Rate** and **Nearest Matching Distance**. Since direct assessment of cross-view consistency is not feasible by merely evaluating the success matches between each method's predicted novel view images and the input view image, we opt to how far the predicted matches deviate from the ground-truth matches.

We first compute ground-truth matching points and every model's matching points using MASt3R [10] upon the input view image and the output view image (ground truth or predicted). Each matching pair is represented as a four-element tuple $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{x}^1, \mathbf{y}^1)$, where $(\mathbf{x}^0, \mathbf{y}^0)$ corresponds

Figure S.3. **Visualized results on *in-the-wild* datasets**

to the point on the input-view image, and $(\mathbf{x}^1, \mathbf{y}^1)$ corresponds to the point on the output-view image.

For each ground-truth matching pair $(\mathbf{x}^0_{gt}, \mathbf{y}^0_{gt}, \mathbf{x}^1_{gt}, \mathbf{y}^1_{gt})$, we find the nearest predicted matching pair in each model's results, denoted as $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{x}^1, \mathbf{y}^1)$, based on the Euclidean distance between points in the input view image. If both $\mathbf{L}_2||(\mathbf{x}^0_{gt}, \mathbf{y}^0_{gt}), (\mathbf{x}^0, \mathbf{y}^0)||$ and $\mathbf{L}_2||(\mathbf{x}^1_{gt}, \mathbf{y}^1_{gt}), (\mathbf{x}^1, \mathbf{y}^1)||$ is smaller than a fixed threshold 20, the match is considered a successful hit. The **Hit Rate** is then calculated as the ratio of successful hits to the total number of ground-truth matches.

For Nearest Matching Distance, we examine whether $\mathbf{L}_2||(\mathbf{x}^0_{gt}, \mathbf{y}^0_{gt}), (\mathbf{x}^0, \mathbf{y}^0)||$ is within the threshold. For those passing this check, we compute the mean distance $\mathbf{L}_2||(\mathbf{x}^1_{gt}, \mathbf{y}^1_{gt}), (\mathbf{x}^1, \mathbf{y}^1)||$ as the **Nearest Matching Distance**, averaging over all successful hits. A detailed pseudo-code explanation can be found in Algorithm 1 and Algorithm 2.

B.4. Results

We show more visualized results of our own methods along with ground truth on C3DFS in Fig. S.11, on Objaverse [3] in Fig. S.12, and on Room-Texture [13] in Fig. S.13. More visualized comparisons with baselines on Room-Texture [13], SUNRGB-D [18] and 3D-FRONT [4] are shown in Fig. S.4. More results on in-the-wild datasets are shown in Fig. S.3.A more complete ablation study on other datasets including Objaverse and Room-Texture is shown in Tab. S.3. Some continuous rotation examples on SUNRGB-D and 3D-FRONT are shown in Fig. S.5, and more cross-view matching results without ground-truth pairs as reference are shown in Fig. S.6.

B.5. Applications

Object Removal Since we can predict mask images under novel views, we can support simple image editing tasks like novel view object removal by simply setting a threshold value in the mask image and mask out corresponding pixels to achieve object removal. An example is shown in Fig. S.9.

Reconstruction The capability to synthesize novel view images that are consistent with the input view image demonstrates that the model possesses 3D-awareness, which can assist downstream tasks such as reconstruction. We leverage an off-the-shelf reconstruction method DUSt3R [21] using the input-view image and novel view images predicted by our method. Two visualized examples are shown in Fig. S.10.

B.6. Mutual Occlusion

In multi-object compositions, mutual occlusion between objects is very common. Although we did not specifically design the method to make the model aware of mutual occlusion, the model has learned some understanding of these occlusion relationships. A series of research efforts [15, 19, 23, 25–27] specifically focus on addressing mutual occlusion relationships by predicting the amodal masks or synthesizing amodal appearance, but these models typically do not consider scenarios involving camera view change. Moreover, there may not be a well-established metric to measure how well the model understands mutual occlusion from novel viewpoints. We provide a simple experiment and discussion in this section to illustrate model's comprehension of mutual occlusion.

First, in the context of novel view synthesis, the comprehension of occlusion relationships can be divided into

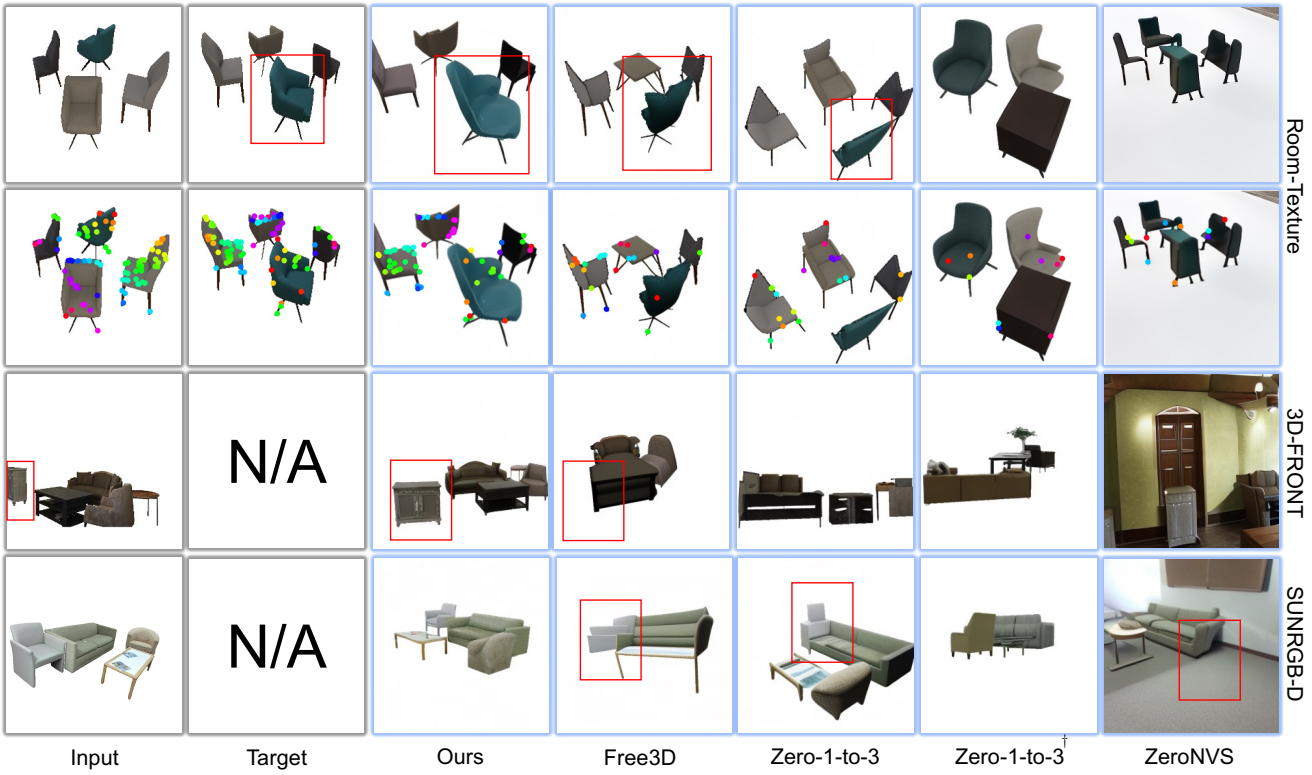

Figure S.4. **Visualized comparison on Room-Texture [13], SUNRGB-D [18], and 3D-FRONT [5].**

Table S.3. **Ablation study on various datasets.**

Dataset	Method	Novel View Synthesis			Placement	Cross-view Consistency	
		PSNR(↑)	SSIM(↑)	LPIPS(↓)	IoU(↑)	Hit Rate(↑)	Dist(↓)
C3DFS	w/o depth	17.080	0.819	0.178	57.2	18.6	**42.6**
	w/o mask	16.914	0.818	0.187	54.7	14.7	49.6
	w/o sch.	16.166	0.808	0.212	49.1	9.7	48.6
	Ours	**17.432**	**0.825**	**0.171**	**58.1**	**19.3**	44.9
Room-Texture	w/o depth	9.829	0.705	**0.365**	**25.7**	**5.0**	**76.1**
	w/o mask	9.576	0.699	0.384	24.2	2.9	91.4
	w/o sch.	9.173	0.689	0.392	22.4	2.8	92.0
	Ours	**10.014**	**0.718**	0.366	24.2	4.4	78.1
Objaverse	w/o depth	17.457	0.835	0.178	50.5	15.9	**45.5**
	w/o mask	17.176	0.834	0.187	47.3	12.8	53.6
	w/o sch.	16.642	0.825	0.210	43.2	9.6	51.7
	Ours	**17.749**	**0.840**	**0.169**	**51.3**	**17.0**	47.2

two parts. The first is the ability to synthesize parts that were occluded in the input view. The second is the ability to synthesize new occlusion relationships under the novel view. We show several examples of synthesizing occluded parts and synthesizing new occlusions in Fig. S.8. We believe this capability is learned in a data-driven way since the multi-object composites are physically plausible regarding these occlusion relationships.

Secondly, we now propose a new metric to evaluate the capability of understanding mutual occlusion under this setting. We first use visible mask and amodal mask in the input-view image to determine how heavily an object is occluded:

1. If an object's visible mask is exactly its full mask, there

Figure S.5. **Continuous rotation examples on SUNRGB-D and 3D-FRONT**. We rotate the camera around the multi-object composites, successfully synthesizing plausible novel-view images across a wide range of camera pose variations. This first five examples are from SUNRGB-D, and the last three examples are from 3D-FRONT.

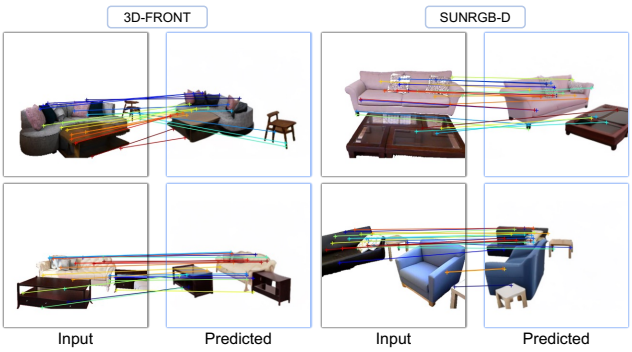

Figure S.6. **Visualized cross-view matching results**. Since we do not have ground truth image for 3D-FRONT and SUNRGB-D, we only visualize cross-view matching results using our predicted images. But we can still observe a strong cross-view consistency from the accurate matching results.

exists no occlusion.

2. If an object's visible mask is more than 70% of its full mask, the object is occluded.
3. If an object's visible mask is less than 70% of its full mask, the object is heavily occluded.

Afterward, we segment the predicted view image with ground truth per-object visible mask. We calculate the specific region's PSNR, SSIM, and LPIPS metrics as shown in Tab. S.4. It can reflect how well our model and baseline models are at synthesizing novel view plausible images that are originally occluded under the input view. There are 10903 fully visible objects, 6058 occluded objects, and 2215 heavily occluded objects. This experiment is conducted on our own C3DFS.

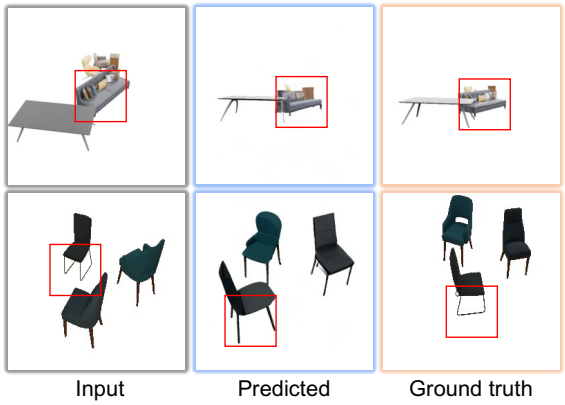

Figure S.7. **Failure Cases**. It is hard for our model to learn extremely fine-grained consistency on objects with delicate structure and texture.

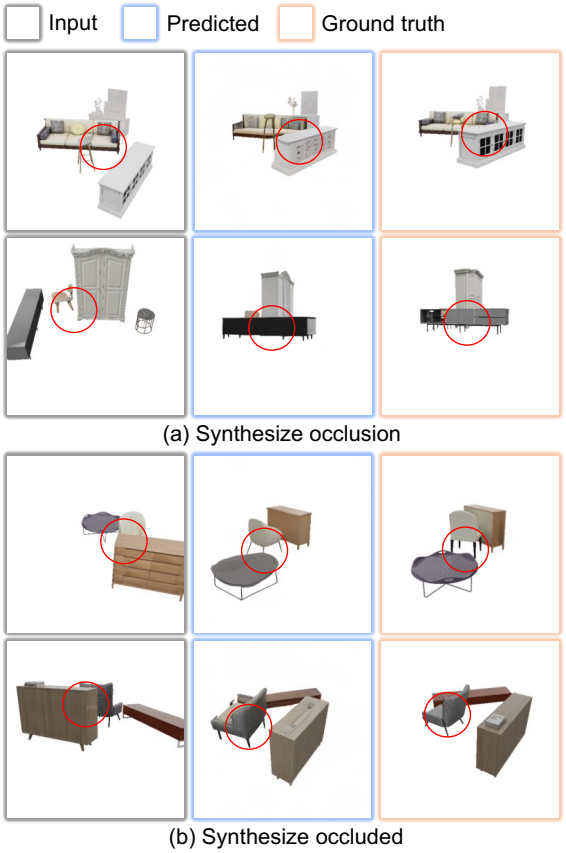

Figure S.8. **Occlusion Synthesis Capability**. Our proposed method can synthesize new occlusion relationship under novel views as shown in the highlighted area of sofa or cabinet in (a). Our method can also hallucinate occluded parts as shown in the highlighted area of chairs in (b).

C. Failure Cases and Limitations

Failure Cases We showcase two failure cases in Fig. S.7. We can observe that delicate structure and texture like colorful pillows on the sofa or slim legs of chairs are hard for our model to learn. Though object placement is approximately accurate, more fine-grained consistency is not quite ideal in these cases. We believe training with a higher resolution and incorporating epipolar constraints will mitigate this problem in the future.

Limitations We identify two limitations of our work. Firstly, though we achieve stronger cross-view consistency with the input view image, our model does not guarantee the multi-view consistency between our synthesized images. It is plausible to synthesize any results in areas with ambiguity, leading to potential multi-view inconsistency in our model. We believe incorporating multi-view awareness techniques [9, 12, 16, 17, 20, 24] can mitigate this problem. Secondly, we do not model background texture in our framework due to difficulty of realistically mimicking real-

Table S.4. **Evaluation on objects with varying extents of occlusion.**

Method	Visible			Occluded			Heavily Occluded		
	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)
Ours	**11.45**	**0.56**	**0.13**	**11.33**	**0.55**	**0.14**	**10.57**	**0.55**	**0.14**
Zero-1-to-3	9.46	0.54	0.16	9.33	0.52	0.17	9.00	0.53	0.16
Zero-1-to-3†	9.68	0.55	0.14	9.54	0.52	0.15	9.26	0.53	0.15

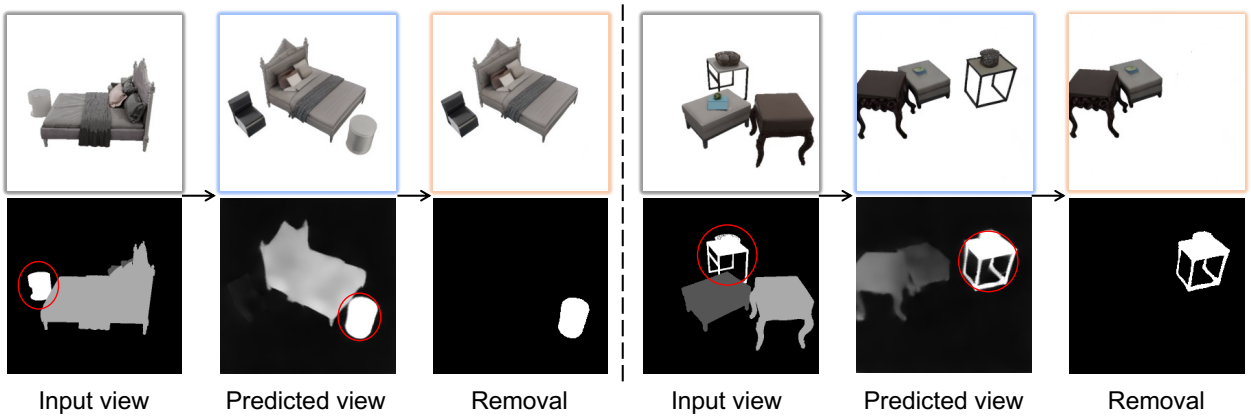

Figure S.9. **Object Removal Example**. We can remove an object under novel views by setting a threshold to the predicted mask image and delete corresponding pixels.

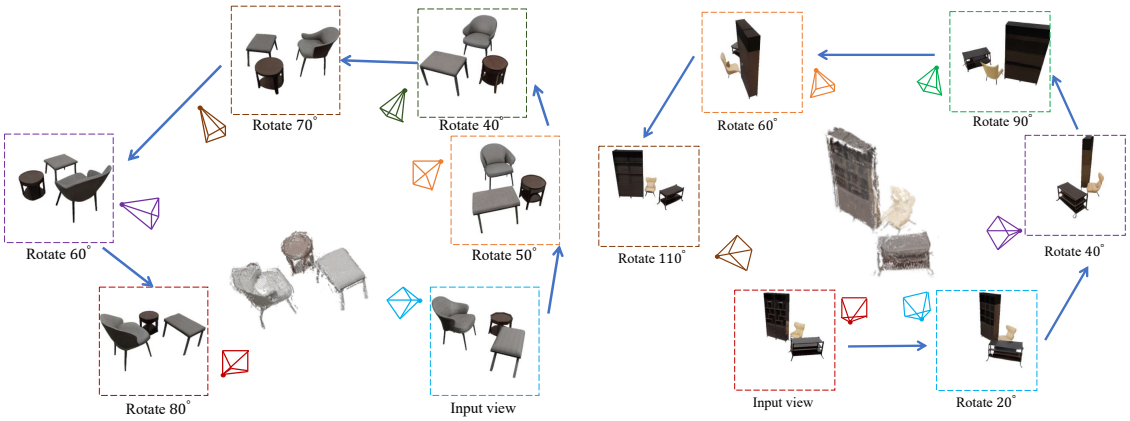

Figure S.10. **Reconstruction results using DUSt3R.** We rotate our camera around the multi-object composite and use the predicted images along with the input-view image for reconstruction.

world background texture, making it less convenient to directly apply our method to in-the-wild images. We believe training on more realistic data with background in the future can make our model more convenient to use.

D. Potential Negative Impact

The use of diffusion models to generate compositional assets can raise ethical concerns, especially if used to create realistic yet fake environments. This could be exploited for misinformation or deceptive purposes, potentially leading to trust issues and societal harm. Additionally, hallucinations from diffusion generation models can produce misleading or false information within generated images. This is particularly concerning in applications where accuracy and fidelity to the real world are critical.

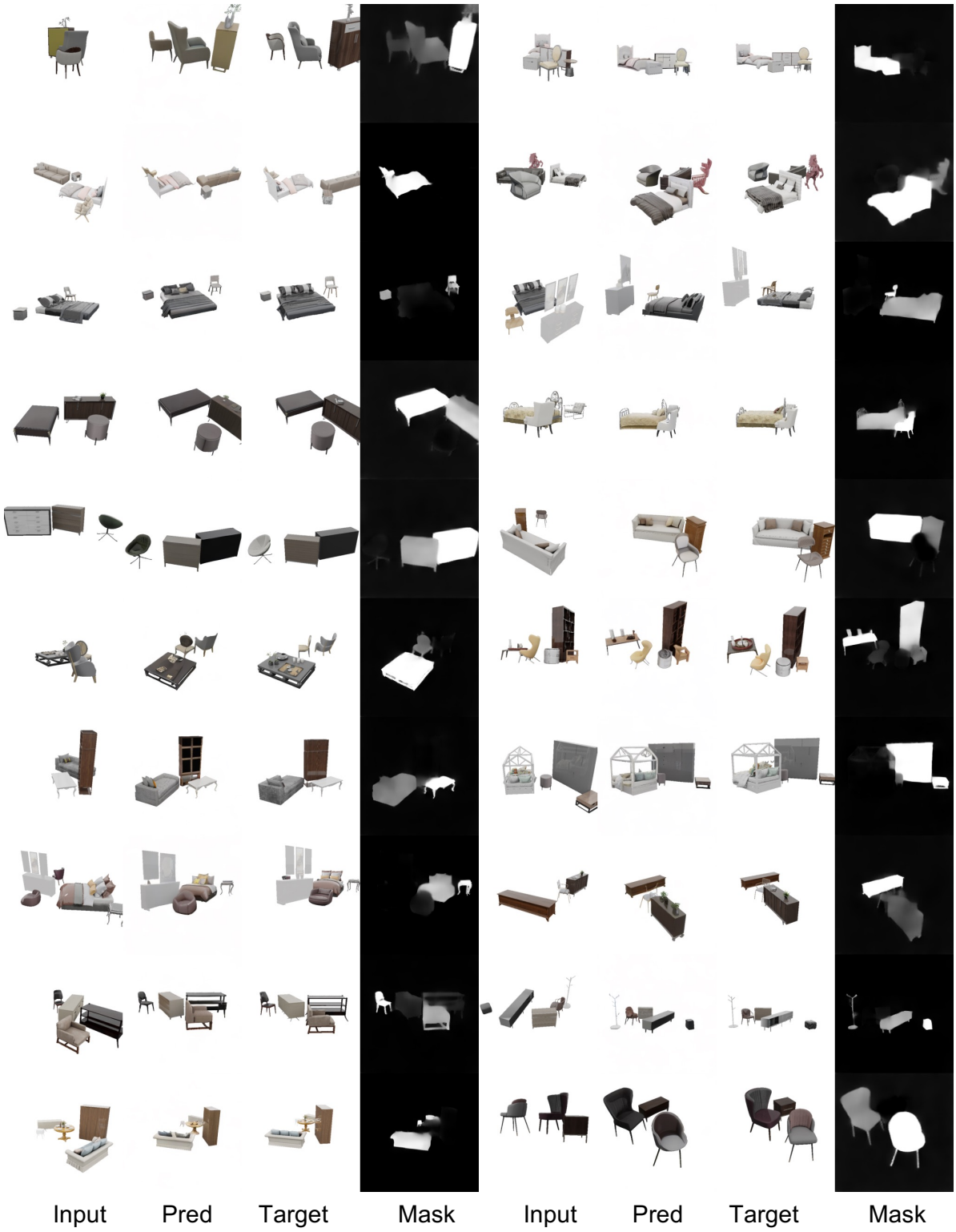

Figure S.11. **More visualized results on C3DFS dataset.**

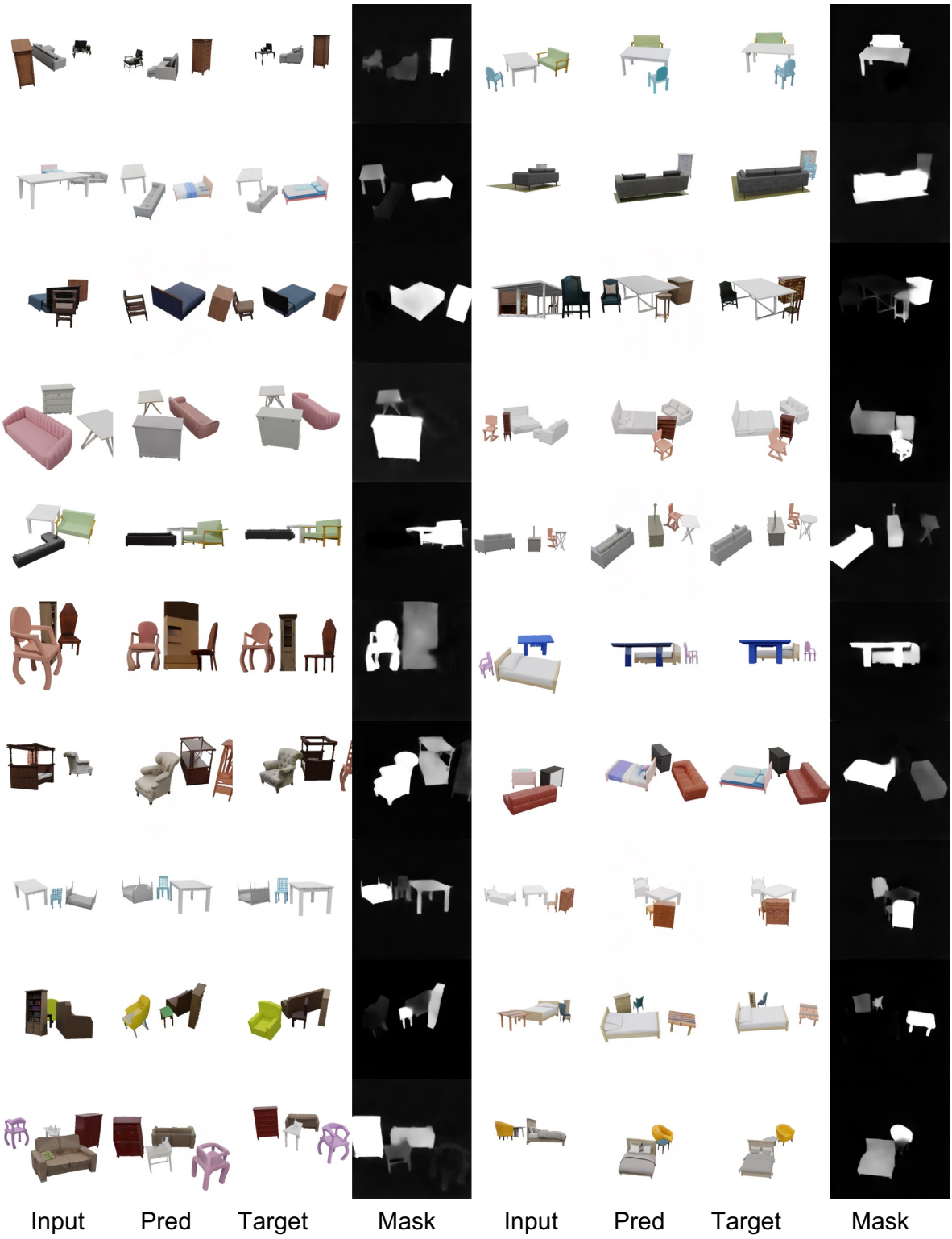

Figure S.12. **More visualized results on Objaverse dataset.**

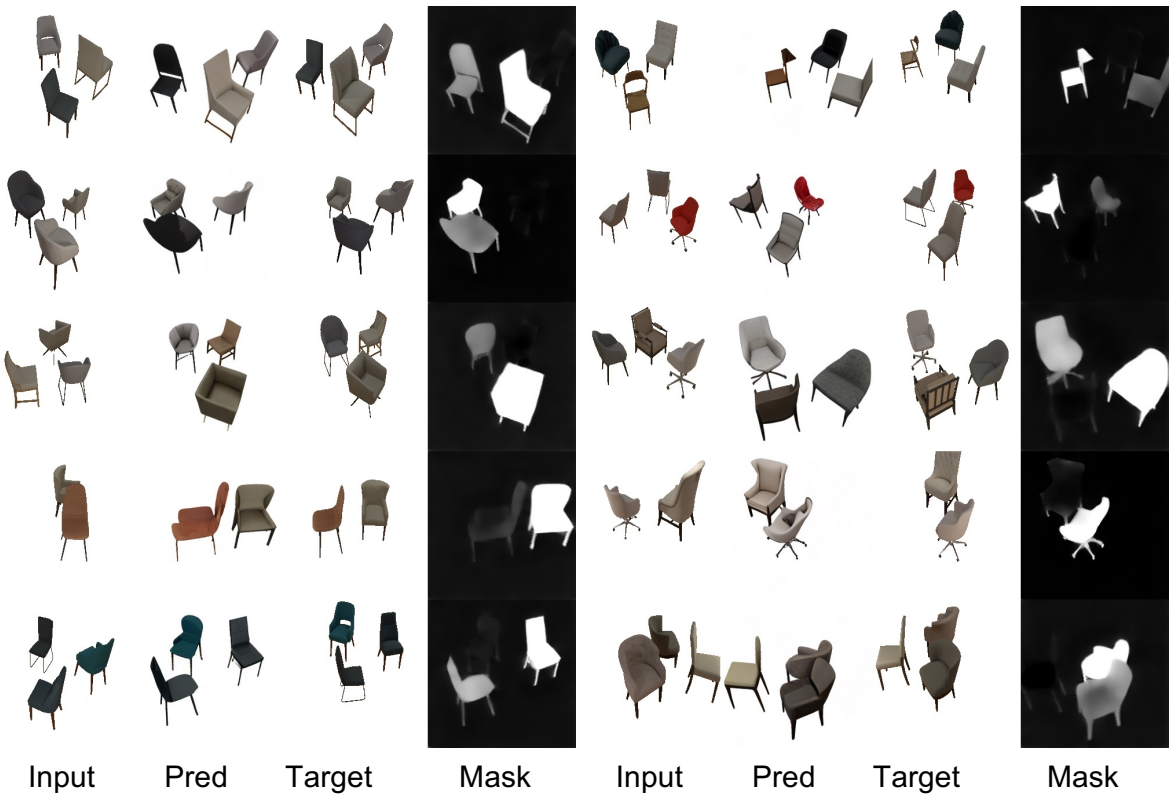

Figure S.13. **More visualized results on Room-Texture dataset.**

References

[1] Yixin Chen, Junfeng Ni, Nan Jiang, Yaowei Zhang, Yixin Zhu, and Siyuan Huang. Single-view 3d scene reconstruction with high-fidelity shape and texture. In *International Conference on 3D Vision (3DV)*, 2024. 2

[2] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 2

[3] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3, 4

[4] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *International Conference on Computer Vision (ICCV)*, 2021. 4

[5] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision (IJCV)*, 129:3313–3337, 2021. 2, 3, 5

[6] Ming Gui, Johannes S Fischer, Ulrich Prestel, Pingchuan Ma, Dmytro Kotovenko, Olga Grebenkova, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. Depthfm: Fast monocular depth estimation with flow matching. *arXiv preprint arXiv:2403.13788*, 2024. 2, 3

[7] Yifan Jiang, Hao Tang, Jen-Hao Rick Chang, Liangchen Song, Zhangyang Wang, and Liangliang Cao. Efficient-3dim: Learning a generalizable single-image novel-view synthesizer in one day. *arXiv preprint arXiv:2310.03015*, 2023. 1

[8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *International Conference on Computer Vision (ICCV)*, 2023. 2, 3

[9] Xin Kong, Shikun Liu, Xiaoyang Lyu, Marwan Taher, Xiaojuan Qi, and Andrew J Davison. Eschernet: A generative model for scalable view synthesis. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 7

[10] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision (ECCV)*, 2024. 3

[11] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *International Conference on Computer Vision (ICCV)*, 2023. 1

[12] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 7

[13] Rundong Luo, Hong-Xing Yu, and Jiajun Wu. Unsupervised discovery of object-centric neural fields. *arXiv preprint arXiv:2402.07376*, 2024. 4, 5

[14] 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 preprint arXiv:2304.07193*, 2023. 1

[15] Ege Ozguroglu, Ruoshi Liu, Dídac Surís, Dian Chen, Achal Dave, Pavel Tokmakov, and Carl Vondrick. pix2gestalt: Amodal segmentation by synthesizing wholes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 4

[16] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023. 7

[17] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv:2308.16512*, 2023. 7

[18] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 4, 5

[19] Basile Van Hoorick, Pavel Tokmakov, Simon Stent, Jie Li, and Carl Vondrick. Tracking through containers and occluders in the wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 4

[20] Peng Wang and Yichun Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023. 7

[21] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 4

[22] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[23] Katherine Xu, Lingzhi Zhang, and Jianbo Shi. Amodal completion via progressive mixed context diffusion. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9099–9109, 2024. 4

[24] Jiayu Yang, Ziang Cheng, Yunfei Duan, Pan Ji, and Hongdong Li. Consistnet: Enforcing 3d consistency for multi-view images diffusion. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 7

[25] Guanqi Zhan, Chuanxia Zheng, Weidi Xie, and Andrew Zisserman. Amodal ground truth and completion in the wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 4

[26] Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised scene deocclusion. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[27] Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4