

# Gen3DSR: Generalizable 3D Scene Reconstruction via Divide and Conquer from a Single View

## Supplementary Material

### Overview

This supplementary material includes additional details and results which provide insights into our novel approach for generalizable 3D scene reconstruction. In section 6, we present supporting information for the reproducibility of our method and the evaluation setup. Section 7 extends the ablation study in the main paper and shows the effect of replacing different modules in the pipeline (7.2), presents a qualitative comparison for the ablations (7.3), and discusses the application on outdoor scenes (7.4). Then, in section 8, we elaborate and illustrate the limitations of our method. Additional results and our code base can be found on the project page <sup>1</sup>.

### 6. Implementation details

#### 6.1. Amodal completion

Amodal completion differs from inpainting by ensuring that the model does not hallucinate new objects when recovering the missing parts, as shown in Figure 9.

The amodal completion step in our framework is fulfilled by an image-to-image Stable Diffusion [64] model fine-tuned on a custom dataset. We create this dataset starting from OmniObject3D [77] which contains 6000 high-quality real-scanned 3D objects. Though the model only needs 2D images for training, we found it beneficial to render 3D objects instead of using real images. This allows us to obtain well-segmented singular objects. The Blender-rendered images of the objects are used as the target images. To obtain the conditioning view, we mask-out parts of the object by randomly overlaying the silhouette of an arbitrary object sampled from the Objaverse dataset [13]. All the channels of the background and the occluded pixels are set to the same value of 127.5 (equivalent to zero after image normalization). The category labels provided in OmniObject3D are used as prompts to guide the diffusion process. Training samples from the dataset are provided in Figure 8. We train the model at a resolution of  $512 \times 512$  for 25000 iterations with a batch size of 16.

As our amodal completion model does not explicitly infer the amodal mask of the completed object, we obtain it by using a foreground segmentation model [61].

#### 6.2. Reprojection and view-space alignment

Utilizing the estimated depth map  $D$  alongside the camera calibration  $K_{img}$ , we perform pixel unprojection from the input image, resulting a point cloud  $P^{view}$  within the view space. We refer to  $P^{view}$  as our layout guide; this serves as a pivotal reference, ensuring the alignment of all individually reconstructed components within the view space, thus forming the complete scene. While the background modeling step directly fits a SDF to the corresponding background points  $P_{bg}^{view}$ , the instance processing step uses  $P_i^{view}$  for two purposes: reprojection and alignment.

The straightforward approach of simply cropping the instances out of the original image and feeding them to the single-shot object reconstruction method  $\mathcal{R}$  leads to deformed reconstructions. This happens because the view-conditioned diffusion model  $\mathcal{Z}$ , which is used for generating novel views of the object, assumes images are captured under a predefined setup [48]. Specifically, the object should be in the center of the image, captured by a camera with field of view of  $49.1^\circ$ , positioned at a distance between  $[1.0, 1.7]$  from the object which is normalized to fit into the unit cube. As these conditions are not satisfied by arbitrary crops within the image, we project  $P_i^{view}$  into a crop  $C_i$  with similar properties. The virtual camera employed for projection uses the  $\mathcal{Z}$ -compatible camera setup; *i.e.*, a camera positioned at a distance of 1.5 from the normalized  $P_i^{view}$  and oriented towards its center, with intrinsics  $K_{crop}$ . The reconstructed object might have a different scale compared to  $P_i^{view}$ . Therefore we estimate the scale factor  $s_i$  that aligns the reconstructed object with  $P_i^{view}$  using a RANSAC-based approach [18], accounting for the possible mismatches.

#### 6.3. Background modeling

The MLP used to reconstruct the background has 4 hidden layers with 128 neurons each and Softplus activations. The network is trained from scratch for each scene using points sampled along background camera rays. The SDF of the points is computed as the distance to the unprojected estimated depth. As we do not sample points in the regions originally occluded, the network implicitly interpolates the missing areas based on the visible surrounding regions.

#### 6.4. Integrated models

Our method ensembles several models, each tackling a different sub-tasks as follows:

- Camera calibration (estimation of field of view and principal point): Perspective Fields [33] trained on 360cities [1]

<sup>1</sup><https://andreedogaru.github.io/Gen3DSR>

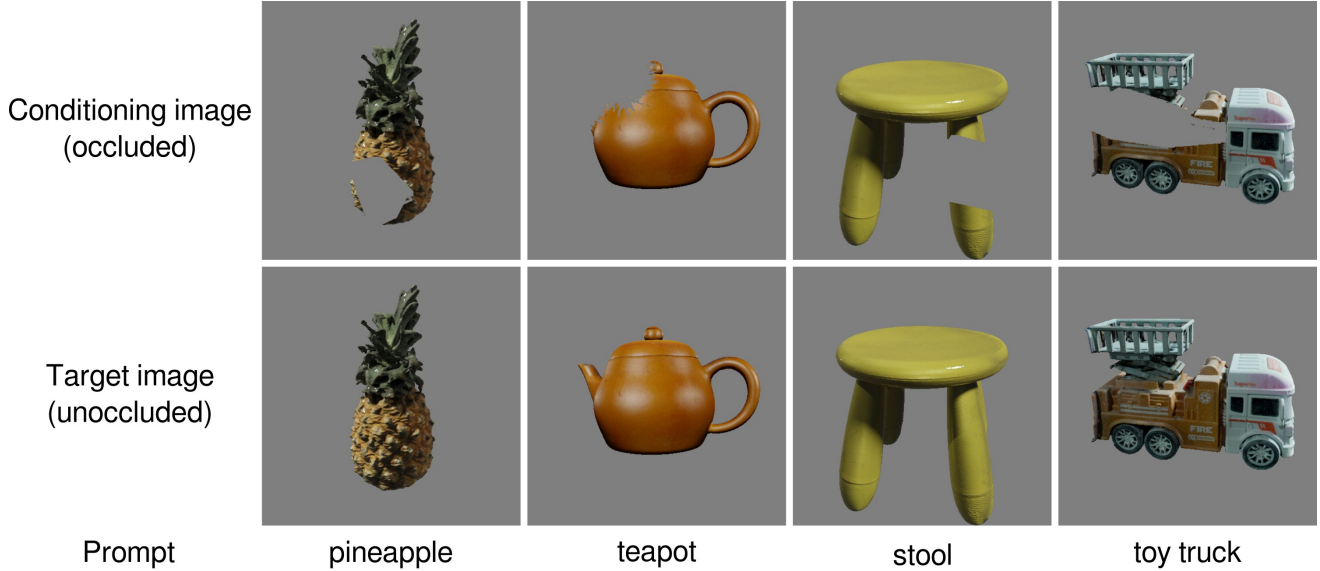


Figure 8. Training samples from the amodal completion synthetic dataset with objects from OmniObject3D [77].



Figure 9. In comparison to our amodal completion, inpainting models tend to hallucinate new objects when filling in the holes. Both models use *desk* as prompt.

and EDINA [15] datasets.

- Entity segmentation: CropFormer [60] with HorNet-L backbone trained on EntitySeg dataset [60].
- stuff-thing segmentation: OneFormer [31] with DiNAT-L backbone trained on ADE20K [86]. The model predicts 150 classes which are grouped in stuff and thing. We make some modifications to the original grouping; *thing*  $\rightarrow$  *stuff*: window, door, curtain, mirror, fence, rail, column, stairs, screen door, balustrade, step; *stuff*  $\rightarrow$  *thing*: plant, tent, crt screen, cradle, blanket.
- Monocular depth estimation: Depth Anything [80] fine-tuned on NYUv2 [54] (metric depth) and Marigold [34] with Stable Diffusion v2 backbone trained on two synthetic datasets (affine-invariant depth).
- Object recognition: OVSAM [82] which combines SAM [38] with CLIP[32] and is trained on COCO [44] and LVIS [23] datasets. For each object we sample 5 points inside the eroded instance mask to prompt the model.
- Amodal completion: our model based on Stable Diffusion v1.5 and trained on the synthetic dataset described in

section 6.1 of this supplementary material.

- Single-image object reconstruction: DreamGaussian [69] which reconstructs the object using 3D Gaussians [35] and employs Zero-1-to-3 XL [48] trained on ObjaverseXL [14] as the 2D diffusion prior.

We show variants of our pipeline with different modules for some of the processing steps (depth estimation, stuff-thing segmentation, and object reconstruction) in section 7.2.

## 6.5. Inference time

Using the default pipeline configuration on an NVIDIA RTX A5000, for an image with resolution  $1500 \times 1500$ , the scene analysis stage takes  $\approx 80$  sec and for each instance specific processing additional  $\approx 90$  sec are required. Similar to other compositional methods (*e.g.*, InstPIFu [45]), the inference time increases linearly with the complexity of the scene (number of objects). This limitation can be mitigated by parallellizing the reconstruction of the objects, provided the available hardware resources allow it. Still, the time per instance is quite high, mostly due to the inference time of DreamGaussian [69]. This particular method optimizes 3D Gaussians to views of the object and has an additional step for texture refinement, adding to the overall runtime. However, as we designed a modular pipeline, we can replace DreamGaussian [69] with new, improved methods for single-view 3D object reconstruction, such as LaRa [8]. This very recent method enables us to reduce the processing time per instance to  $\approx 50$  sec. Finally, the processing time could be further improved by streamlining the pipeline for a specific configuration of modules.

## 6.6. Evaluation details

For 3D-FRONT [19, 20] we report the metrics at the original scale of the geometry in the dataset using 1000000 points sampled from both the reconstructed meshes and the ground truth geometry. The F-Score is computed at a threshold of 0.1. As most of the modules in our pipeline operate at a high-resolution, the original resolution of the images ( $648 \times 484$ ) is insufficient. Therefore, we first increase their size to  $1296 \times 968$  using a super-resolution method [76].

For HOPE-Image [73], we compose the ground truth geometry using the provided pose annotations. Additionally, we found that a scale of 0.1 must first be applied to the original object meshes to match with the images. The metrics are reported at this scale based on 500000 sampled points and the threshold used for F-Score is 1.0. On this dataset, we only evaluate the reconstruction of the foreground instances, as there is no ground truth background geometry.

## 7. Additional experiments

### 7.1. HOPE-Image

We include in Figure 10 the qualitative results of the experiments on HOPE-Image dataset [73]. We compare our compositional approach with DreamGaussian [69], which reconstructs all the objects in the image at once. Our method can better handle complex scenes with many objects, as each instance is individually reconstructed. In contrast, both the appearance and the geometry of DreamGaussian’s reconstructions degrades when applied on scenes with multiple objects. This is mainly due to limitations of the Zero-1-to-3 XL method, which fails to generate realistic, multi-view-consistent views. The pitfall is expected, as the domain required to be modeled by the prior increases exponentially with the number of objects. We do not include InstPIFu [45] in the comparison on HOPE-Image dataset because the method fails to detect any of the considered objects in the scenes.

### 7.2. Alternative models

The proposed method is designed as a modular framework, allowing the straightforward replacement of the integrated models summarized in 6.4. We showcase this property of our method by exchanging the Marigold [34] depth estimation with Depth Anything [80], the stuff-thing detection based on OneFormer [31] with one based on CLIPSeg [50], and the single-view object reconstruction model, DreamGaussian [69], with One-2-3-45 [46]. We evaluate in Table 5 the performance of these modifications on full scene reconstruction (background and foreground objects) using the 3D-FRONT [19] dataset. For CLIPSeg, we define empirically a set of prompts for foreground (object, furniture) and background (background, floor, wall, curtain, window, ceiling), and consider foreground pixels the ones that have a lower

than 0.5 score for all the background prompts or higher than 0.1 for any of the foreground ones. The performance of our method decreases in this case because DreamGaussian poorly reconstructs the background regions that are misclassified as things. We observe this happens more frequently on indoor rooms when using CLIPSeg, and on tabletop scenes, such as the one in our method figure from the main paper, when using OneFormer.

The results show that our default configuration (as reported in the main paper) yields the best metrics on this dataset. Still, the alternative configurations are competitive, which proves that our pipeline is flexible and robust to different implementations of the modules.

### 7.3. Ablations

We provide in Figure 11 qualitative results corresponding to the two ablations considered in the main paper (see section 4.3). When ablating the amodal completion step, the reconstructed objects have holes or incorrect textures, corresponding to the regions that were occluded by other objects in the scenes. When skipping the reprojection step of the pipeline and directly using crops from the input image, the objects are reconstructed with deformed shapes to match the input view. By including both components, our full method reconstructs complete objects with a physically-correct geometry. We also consider these ablations together with LaRa [8], a very recent method for single view object reconstruction that operates in a feed-forward fashion. Compared to DreamGaussian which relies on Zero-1-to-3 XL to generate novel views conditioned on the input crop, LaRa uses a newer model, Zero123++, that can generate six fixed views that are more 3D consistent. As can be seen in Figure 12, the proposed amodal completion and reprojection are also beneficial for LaRa, boosting the reconstruction performance.

### 7.4. Outdoor scenes

Our method can reconstruct well a wide range of scenes, from tabletop setups to large rooms with many objects of varying sizes. Though we exemplify the performance of the proposed pipeline on indoor environments, by design the method is also applicable to outdoor scene reconstruction. In Figure 13, we provide qualitative results for such scenes. In these experiments, we use Marigold [34] as a depth estimator and compute the scale and shift based on Depth Anything [80] prediction of the model fine-tuned on KITTI [21] (suited for outdoor scenes).

We choose to focus more on indoor scene reconstruction because most outdoor scenes are predominantly composed of entities categorized as stuff, which we represent simply with a mesh approximating the estimated depth map. We believe a dedicated approach for outdoor background reconstruction would be better suited for this type of scenes.

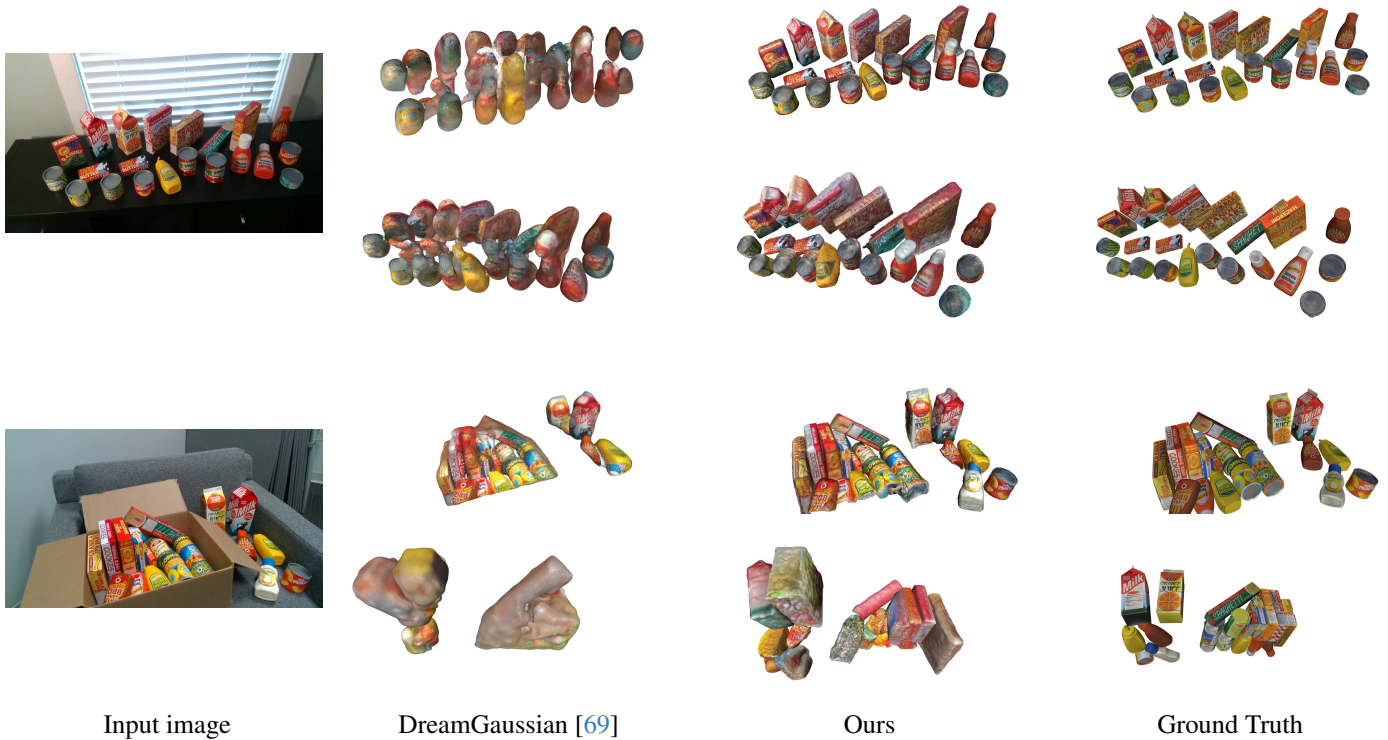


Figure 10. Qualitative results on HOPE-Image [73]. We show each reconstruction from two camera views, including the input one.

Depth estimation	stuff vs thing	Object reconstruction	Chamfer ↓	F-Score ↑
Marigold	OneFormer	DreamGaussian	<b>0.099</b>	<b>75.33</b>
Depth Anything	OneFormer	DreamGaussian	0.135	68.43
Marigold	CLIPSeg	DreamGaussian	0.166	65.56
Marigold	OneFormer	One-2-3-45	0.110	69.40

Table 5. Quantitative evaluation of our method using alternative models on 3D-FRONT [19] dataset.

Nonetheless, outdoor scenes with several objects are reconstructed reasonably well by our method, as can be observed in the qualitative results.

## 8. Limitations

We describe in this section some of the limitations our method inherits from the integrated modules.

As we rely on the unprojected depth for the layout of the scene, when the camera calibration or the scale and offset estimation fail to match the correct units, the reconstructed scene will have an erroneous structure. We provide an example of incorrect camera calibration in Figure 15. Furthermore, since entities are reconstructed independently, without relative constraints between the objects, the method does not address potential physical impossibilities such as intersecting objects.

In our experiments, we notice that DreamGaussian some-

times overestimates the size of the object along the  $z$ -dimension, when it cannot be observed in the input view. Such cases are prominent in Figure 10, *e.g.*, orange juice box in the last example. Additionally, DreamGaussian requires the estimation of the perceived camera elevation of the input crop with respect to the normalized object pose. Since this cannot be robustly inferred, it significantly affects the reconstruction performance of the model, as can be analysed in Figure 14.

Though our approach for modeling the background is superior to previous works, using a small MLP to model the ‘stuff’ of the scene (entities that are not objects) may sacrifice details in the texture. Also, the implicit interpolation for the occluded areas is sometimes insufficient to cover large missing regions, resulting in holes in the background surface.

As the proposed method follows a modular approach, we can improve on this limitations by upgrading the particular modules. For example, conditioning the depth estimation





Figure 11. Ablations visual comparison. In Ours<sub>DG</sub> we use DreamGaussian [69] for reconstructing individual objects with the proposed reprojection of the input pixels and amodal completion of the missing parts.



Figure 12. Ablations visual comparison. In Ours<sub>LaRa</sub> we use LaRa [8] for reconstructing individual objects with the proposed reprojection of the input pixels and amodal completion of the missing parts.

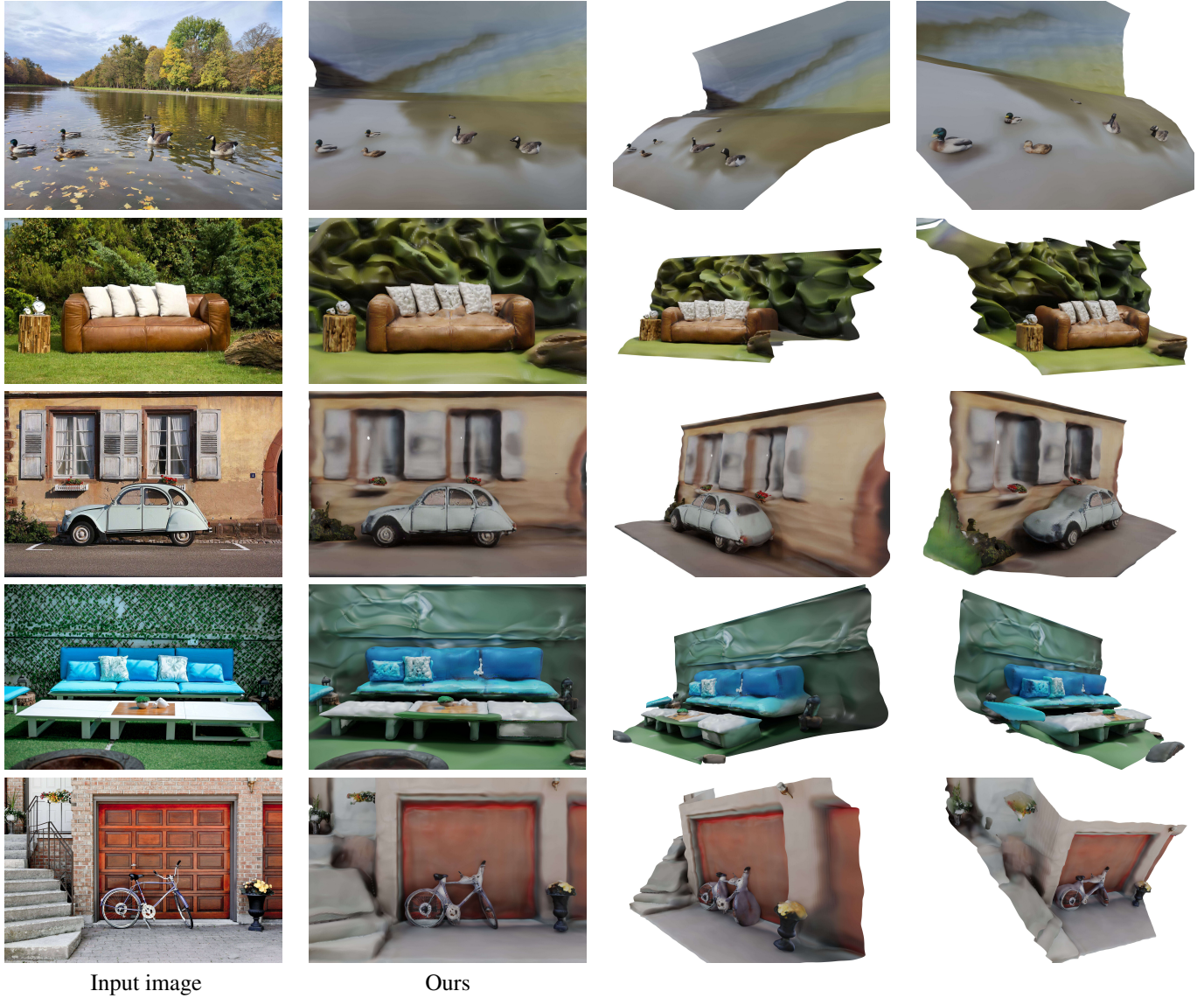


Figure 13. Qualitative results of our method on real-world outdoor scene reconstruction.

on the inferred camera calibration can help disambiguate depth scale [26, 66]. Also, the 3D object reconstruction module can benefit from having the estimated instance depth as an additional input [29, 74]. Lastly, relying more on the overall scene context during the amodal completion stage can improve the recovery of missing object parts due to occlusion, as it has been recently shown in [56, 78].

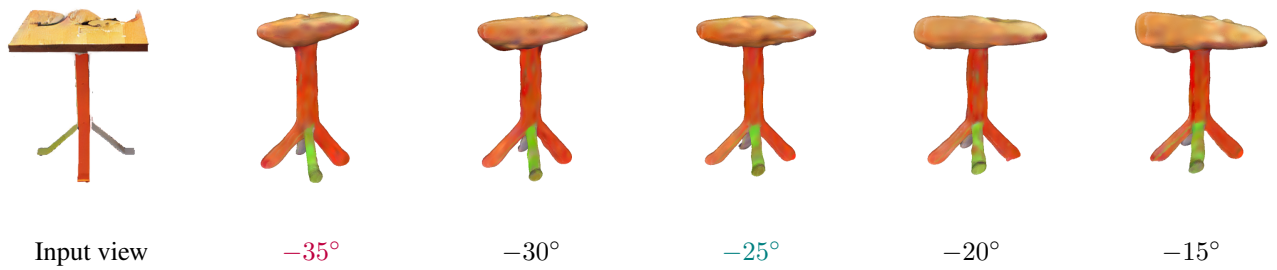


Figure 14. DreamGaussian [69] object reconstructions of the input view under different camera elevation angles. We illustrate the 3D reconstructions using a camera view from the side to highlight their differences. The elevation angle estimated by the method proposed in [46] is  $-35^\circ$ . However, the best reconstruction is achieved using an elevation angle of around  $-25^\circ$ .

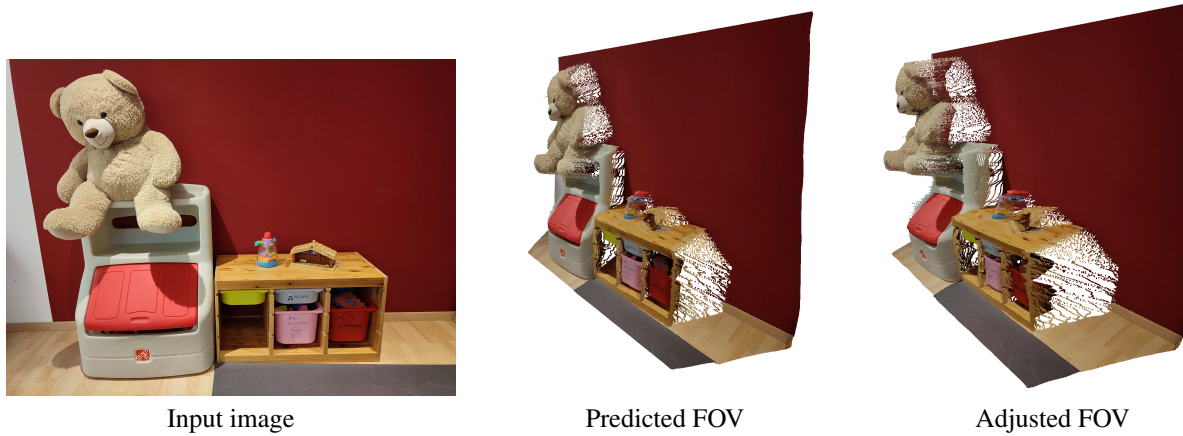


Figure 15. Comparison of unprojected depth under two camera settings. The field of view prediction of PerspectiveFields [33] is incorrect for this image, resulting in a flattened scene. The unprojected depth using a camera with a manually adjusted the field of view has more realistic proportions.