Reconstruction by Generation: 3D Multi-Object Scene Reconstruction from Sparse Observations

Andrii Zadaianchuk[1*], **Leonardo Barcellona**[1*], **Lennard Schuenemann**[1], **Christian Gumbsch**[1], **Zehao Wang**[2], **Muhammad Zubair Irshad**[4], **Fabien Despinoy**[3], **Rahaf Aljundi**[3], **Stratis Gavves**[1], **Sergey Zakharov**[4*†]

[1]University of Amsterdam, [2]KU Leuven, [3]Toyota Motor Europe, [4]Toyota Research Institute
[*]Core Contributor, [†]Project Lead

Accurately reconstructing complex full multi-object scenes from sparse observations remains a core challenge in computer vision and a key step toward scalable and reliable simulation for robotics. In this work, we introduce RecGen, a generative framework for probabilistic joint estimation of object and part shapes, as well as their pose under occlusion and partial visibility from one or multiple RGB-D images. By leveraging compositional synthetic scene generation and strong 3D shape priors, RecGen generalizes across diverse object types and real-world environments. RecGen achieves state-of-the-art performance on complex, heavily occluded datasets, robustly handling severe occlusions, symmetric objects, object parts, and intricate geometry and texture. Despite using nearly 80% fewer training meshes than the previous state of the art SAM3D, RecGen outperforms it by 30.1% in geometric shape quality, 9.1% in texture reconstruction, and 33.9% in pose estimation.

Project page: reconstruction-by-generation.github.io

Figure 1 *RecGen* generates full reconstructions of complex occluded scenes from single or multiple RGB-D images, enabling robust generation of digital twin replicas of real-world environments.Our model recovers occluded geometry of both objects and parts, is robust to imperfect sensor depth, and handles object symmetries — challenges that most recent baselines struggle to address.

1 Introduction

Simulation is increasingly used to train and evaluate embodied AI systems [1–4], however, its overall impact is limited by the substantial cost and complexity associated with constructing high-fidelity digital twins [5]. Constructing such twins typically requires detailed scanning and manual registration of objects within scenes, a labor-intensive process that is difficult to scale. A promising alternative is to recover structured multi-object scenes directly from sparse observations [6, 7]. However, accurately estimating object geometry and 6-DoF pose from limited RGB-D input in cluttered environments remains fundamentally challenging. Occlusions, object symmetries, complex geometry, and noisy depth observations make pose estimation under partial visibility brittle, posing challenges for scalable real-to-sim reconstruction[8].

Generative 3D models [9–11] have recently demonstrated strong potential for reconstructing real-world objects from sparse observations. In parallel, model-free pose estimation methods [12–15] leverage generated 3D shapes to perform pose registration against input images or depth maps. However, these approaches treat shape generation, completion, and pose estimation as separate stages, increasing complexity, compounding errors, and reducing robustness under occlusion. In contrast, we propose RecGen, a unified generative framework that jointly infers object geometry and 6-DoF pose from single or multiple RGB-D observations, enabling coherent reasoning about shape and pose under uncertainty. The novel design and training recipe of RecGen overcomes central limitations faced by existing 3D reconstruction methods, as qualitatively illustrated in Fig. 1.

The first limitation lies in pose registration where shapes predicted by image-conditioned generative models [10, 16] must be aligned post hoc to observed RGB or depth data, a process that is often brittle in cluttered scenes and for symmetric or weakly textured objects. In contrast, RecGen performs joint probabilistic estimation of shape and pose directly in the camera frame, enabling geometrically consistent object reconstruction without requiring separate registration stages.

Related to this problem is object completion under partial visibility where existing generative models either misinterpret visible regions as complete geometry or fail to infer the occluded or unobserved regions due to limited contextual cues. This issue largely arises from training on occlusion-free objects or masked images where the target object is fully visible, unlike real-world conditions. To address this issue, we introduce a large-scale synthetic dataset of occluded objects, which enables RecGen to learn priors that support robust shape completion under challenging occlusions. Importantly, unlike prior methods that take masked images as direct input, we encode masks as positional signals indicating which pixels belong to the object of interest, providing richer contextual information for reasoning about occlusions.

Symmetry further complicates pose estimation and texture reconstruction. The 6-DoF pose estimation of symmetric objects is inherently ambiguous. For objects such as bottles or boxes with semantic labels, textures must respect the object-to-camera orientation to correctly place view-dependent details. Without explicit pose conditioning, texturing networks often produce inconsistent or misaligned textures, as observed in recent generative methods such as SAM3D [7]. To overcome this challenge, we explicitly condition texture reconstruction in RecGen on the estimated object pose, enabling view-consistent and semantically aligned texturing even in the presence of geometric symmetries. Another limitation is that existing methods reconstruct objects as single monolithic meshes, without recovering their internal part structure [10, 16]. However, estimating such shapes and poses is essential to learn part-level control tasks such as articulated object manipulation [17, 18]. To address this, RecGen supports part-level shape and pose estimation by unifying scene decomposition into objects, and objects into parts within a single generalizable framework, by extending our object-level training with part-annotated assets.
Whereas depth images provide geometric cues that improve both shape and pose estimation, most generative models remain RGB-centric and use depth only in alignment stages [10, 16], resulting in error-prone multi-step pipelines. Although SAM3D [7] supports depth input, it is sensitive to commodity sensor noise, leading to pose misalignment and degraded reconstruction. We address this limitation by training on realistically estimated depth from FoundationStereo [19], rather than relying on perfect rendered depth, enabling RecGen to leverage 3D structural cues while remaining robust to imperfect measurements. Finally, an important capability largely unsupported by current generative models [7, 10, 16] is multi-view conditioning, despite its practical relevance in real-world setups where multiple cameras are often available. We train RecGen to explicitly support both single-view and multi-view conditioning within a unified framework. In the multi-view

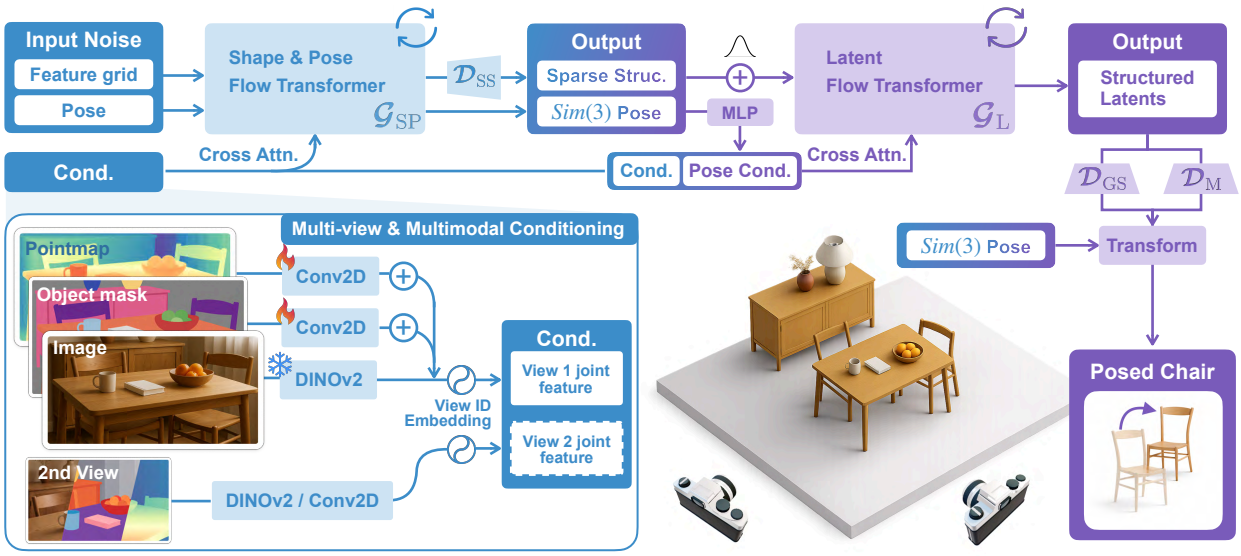


Figure 2 The RecGen architecture. Given one or more input observations consisting of RGB images, Point maps and Object masks, our framework (1) predicts a sparse object structure and its pose in a normalized camera frame, and (2) recovers textured meshes. Both stages employ flow transformer models conditioned on multimodal features, together with dedicated decoders to recover sparse structure, mesh, and texture.

setting, the model can integrate complementary observations across views, reduce reconstruction ambiguities, and improve both geometric consistency and pose accuracy. This enables RecGen to better exploit additional visual information when available.

To this end, RecGen is a novel framework and training recipe for joint shape completion and pose estimation that bridges the gap between generative modeling and real-world 3D reconstruction. Our main contributions are:

- We propose RecGen, a multi-hypothesis framework for jointly estimating object pose and complete 3D textured shape from one or a few images, without any prior knowledge of the object.
- We introduce a synthetic dataset of occluded objects and their parts, enabling robust RGB-D training under heavy occlusion and object symmetries.
- RecGen sets a new state of the art, surpassing SAM3D by 30.1% in geometric shape quality, 9.1% in texture reconstruction, and 33.9% in pose estimation, while trained on 80% less data.
- Extensive experiments show that RecGen generalizes robustly to part-level, occluded, symmetric, and uncommon objects.

2 Related Work

2.1 Pose and shape prediction

Joint modeling of object shape and pose in the camera frame has emerged as an important direction for scene-level 3D reconstruction, supported by rapid progress in the two problems independently. On the shape side, image-conditioned 3D generation has advanced with feed-forward and hybrid generative models such as CRM [20], LGM [21], InstantMesh [16], TRELLIS [10], and Hunyuan3D [22], which improve fidelity and spatial consistency via stronger geometric and latent priors. In parallel, methods for novel-object 6D pose estimation have also achieved strong performance. FoundationPose [15] unifies model-based and model-free 6D pose estimation and tracking for novel objects, while Any6D [12] focuses on model-free pose estimation from a single RGB-D anchor observation. In practice, shape and pose are jointly required and geometrically coupled, motivating recent efforts toward joint prediction. Approaches to joint shape–pose reconstruction fall into two categories: modular pipelines and unified feed-forward methods. Modular approaches (e.g., GigaPose [23], Pos3R [24], OmniShape [25], SceneComplete [26], Gen3DSR [27]) typically combine an image-to-3D reconstructor [16] with a separate pose alignment stage based on correspondences, depth, or registration.

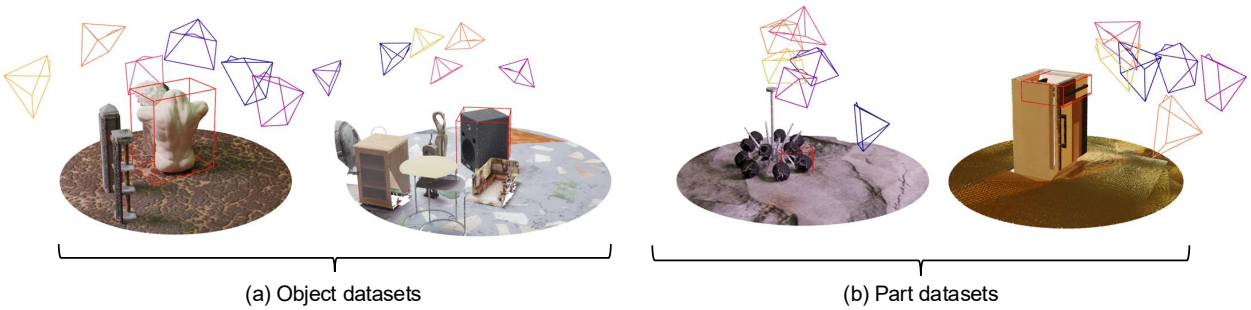


Figure 3 RecGen Training Dataset Samples. Representative examples of 3D assets from our training dataset, including compositional scenes with (a) objects from Objaverse-XL, ABO, HSSD, and (b) parts in object scenes the part-based datasets (PhysXNet, PartNext, PartNet-Mobility). Using such conditioning allows for scene-aware 3D generation of objects and their parts from partially occluded and posed objects, jointly with robust pose estimation of the corresponding assets.

While flexible, such pipelines decouple geometry and pose and may suffer from error propagation. In contrast, unified approaches predict both in a single forward pass. Methods such as CenterSnap [28] and ShAPO [29] jointly estimate complete 3D shape and 6D pose using camera-centered spatial representations. Recently, scene-level generative methods have started to jointly model instance geometry and spatial arrangement. MIDI [30] performs multi-instance diffusion to generate coherent 3D scenes from a single image. SAM3D [7] tackles generative monocular reconstruction, predicting object geometry together with scene layout in the camera frame. These approaches are largely formulated under a single monocular condition and do not naturally support richer multi-view conditioning. Recent works have also explored part-level 3D generation, synthesizing objects as collections of semantic components rather than monolithic meshes [31–34]. However, they typically focus on part decomposition or synthesis rather than pose reasoning from observations. Our approach enables joint shape and pose estimation across multiple conditions, supporting diverse inputs such as RGB-D and multi-view observations while inferring both object- and part-level representations.

2.2 Real-to-Sim in Robotics

Scene reconstruction and generation has gathered significant interest in robotics[35, 36]. Among emerging scene representations, 3D Gaussian Splatting [37] has gained attention due to its photorealistic rendering quality and explicit, point-based structure, which facilitates downstream robotics manipulation and navigation applications [38–47].

Building on these developments, recent robotics research underscores the importance of scene-level 3D generation in real-to-sim pipelines. Several approaches employ reconstructed or generated 3D scenes as intermediate representations for policy learning; for instance, X-Sim [48], DreMa [49], Real2Render2Real [50], and ZeroBot [51] depend on such simulation assets for robot learning. Complementary work focuses on scaling policy evaluation, as in Real2Sim-Eval [52], RobotArena [53], and PolaRiS [54], which transform real scenes into interactive simulation environments for reproducible benchmarking. More recently, a separate line of work has focused on making generated scenes physically usable through post-hoc refinement, e.g., via physics-consistent inter-object reasoning or physics-aware joint shape-pose optimization in cluttered environments [55–57]. Taken together, these works suggest that robotics increasingly demands scene models that are not only visually faithful, but also accurate at the scene level, physically plausible, and readily usable under multi-view observations. Our method targets this need by providing a high-fidelity scene-level generative base model with support for multi-view conditioning.

3 Method

Given one or two views of a real-world scene, we aim to reconstruct structured assets that serve as digital twins of the physical objects. Given v as the view index, we assume that each input viewpoint provides an RGB image $\mathbf{I}^{(v)} \in \mathbb{R}^{d\times d\times 3}$, a depth map $\mathbf{D}^{(v)} \in \mathbb{R}^{d\times d}$, camera intrinsics $\mathbf{K}^{(v)} \in \mathbb{R}^{3\times 3}$, and a set of segmented

regions $\mathbf{M}^{(v)}$ corresponding to the observed objects or object parts. Our objective is to estimate the *shape* $\mathbf{s}$, *pose* $\mathbf{T}^{(v)}$, and *appearance* $\mathbf{a}$ for each segmented region $\mathbf{M}^{(v)}$, where $\mathbf{T}^{(v)} \in \mathrm{Sim}(3)$ denotes a similarity transformation that maps object-centric coordinates to the normalized input frame.
Since an object's pose cannot be fully determined without knowledge of its shape, and vice versa, we jointly model these quantities rather than estimating them independently. Formally, we consider the joint conditional distribution

$$p\big(\mathbf{s}, \mathbf{a}, \{\mathbf{T}^{(v)}\}_v \mid \{\mathbf{I}^{(v)}, \mathbf{D}^{(v)}, \mathbf{K}^{(v)}\}_v\big),$$

which is highly complex and inherently multimodal.
To effectively model this distribution, we employ a generative framework based on rectified flow [58] that jointly produces high-quality 3D object shapes and their corresponding similarity transformations. Figure 2 provides an overview of the proposed framework.

3.1 Reconstruction by Generation

Our reconstruction framework (Fig. 2) consists of two main stages: (1) *object structure and pose generation*, and (2) *high-quality asset recovery*, both trained using rectified flow models to efficiently capture complex data distributions.

3.1.1 Object Structure and Pose Generation.

In the first stage, we jointly generate the object's sparse structure $\{\boldsymbol{p}_i\}_{i=1}^{L}$ together with its pose $\mathbf{T} \in \mathrm{Sim}(3)$, parameterized by rotation $\boldsymbol{R} \in \mathrm{SO}(3)$, translation $\boldsymbol{t} \in \mathbb{R}^3$, and isotropic scale $s \in \mathbb{R}^+$. The transformation $\mathbf{T}$ maps object-centric coordinates to the normalized input frame. To enable dense tensor processing, the sparse voxel coordinates are converted into a dense binary occupancy grid $\boldsymbol{O} \in \{0,1\}^{64\times64\times64}$, where active voxels are set to 1. The direct generation of $\boldsymbol{O}$ is computationally expensive. We therefore employ a 3D convolutional VAE to encode it into a lower-resolution continuous feature grid $\boldsymbol{S} \in \mathbb{R}^{16\times16\times16\times8}$, providing a smooth latent space suitable for rectified flow training with minimal information loss.
The pose $\mathbf{T}$ is jointly denoised alongside $\boldsymbol{S}$ by concatenating its parameters with the structure features as an additional token, enabling the model to exploit geometric consistency between shape and pose. At each timestep t, the generator $\boldsymbol{\mathcal{G}}_{\mathrm{SP}}$ predicts velocity fields for both $\boldsymbol{S}$ and $\mathbf{T}$, which are updated via Euler integration.
A transformer-based generator $\boldsymbol{\mathcal{G}}_{\mathrm{SP}}$ is trained to jointly produce $\boldsymbol{S}$ and $\mathbf{T}$ from noisy inputs. The serialized input grid is augmented with positional encodings and processed by a transformer with adaptive layer normalization (AdaLN) and gating mechanisms [59]. Conditioning is provided through cross-attention on our multimodal features formed by DINOv2 [60] image features, as well as point map and mask features extracted from respective inputs.
The denoised feature grid $\boldsymbol{S}$ is decoded into the discrete occupancy grid $\boldsymbol{O}$ using a decoder $\boldsymbol{\mathcal{D}}_{\mathrm{SS}}$, of the same VAE, and converted back into active voxels $\{\boldsymbol{p}_i\}_{i=1}^{L}$, representing the predicted sparse object structure. The denoised transformation $\mathbf{T}$ is applied to recover the object's rotation, translation, and scale.

Pose parameterization and normalization. Inspired by [61], we adopt pose parameterizations that avoid discontinuities, which could impair gradient-based optimization. In particular, we use the 6D continuous representation proposed in [62] as our main rotation representation for the structure generator $\boldsymbol{\mathcal{G}}_{\mathrm{SP}}$, which stores the first two columns of a rotation matrix and recovers the third via Gram–Schmidt orthogonalization. For the latent generator $\boldsymbol{\mathcal{G}}_{\mathrm{L}}$, we use the 9D pose parametrization as it was shown to perform best for model inputs. Both representations are extended with a translation vector $\boldsymbol{t} \in \mathbb{R}^3$ and an isotropic scale $s \in \mathbb{R}^+$, yielding the full pose $\mathbf{T} = \{\boldsymbol{R}, \boldsymbol{t}, s\}$.
In addition to choosing an appropriate rotation representation, we apply z-score normalization to all pose components computed over the entire training set:

$$\tilde{\mathbf{T}} = \{(\boldsymbol{\rho} - \boldsymbol{\mu}_\rho)/\boldsymbol{\sigma}_\rho,\ (\boldsymbol{t} - \boldsymbol{\mu}_t)/\boldsymbol{\sigma}_t,\ (s - \mu_s)/\sigma_s\}\,,$$

where $\boldsymbol{\rho}$ denotes the rotation parameters (quaternion or 6D), and $\boldsymbol{\mu}, \boldsymbol{\sigma}$ are component-wise means and standard deviations over the training dataset. This standardization ensures zero mean and unit variance for each

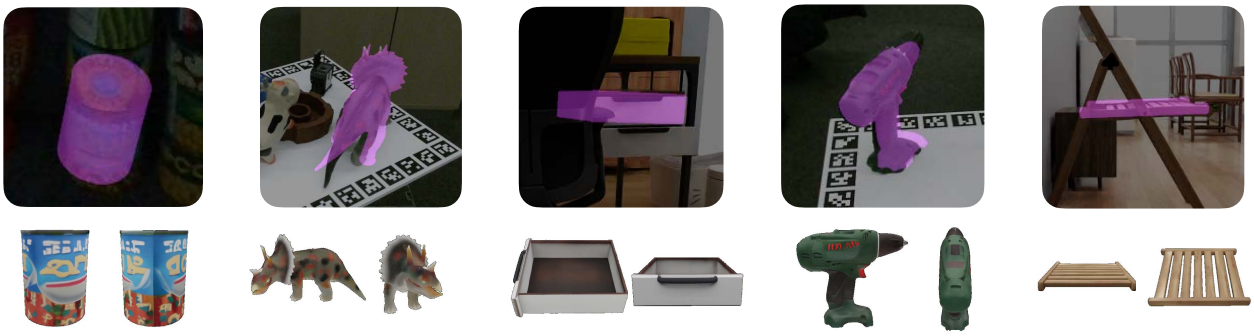


Figure 4 RecGen qualitative results. We demonstrate that our method is robust to occlusions, handles symmetric objects, and generalizes to real-world data despite being trained exclusively on synthetic data.

component, preventing any single quantity from dominating the flow matching objective. During inference, we denormalize via

$$\mathbf{T} = \{\tilde{\boldsymbol{\rho}} \cdot \boldsymbol{\sigma}_\rho + \boldsymbol{\mu}_\rho, \tilde{\boldsymbol{t}} \cdot \boldsymbol{\sigma}_t + \boldsymbol{\mu}_t, \tilde{s} \cdot \sigma_s + \mu_s\}.$$

Dynamic cropping and mask conditioning. To specify the target object, most object-centric approaches apply segmentation masks to the RGB image, retaining only foreground RGB pixels. However, this discards contextual environment information that can help infer occlusions and scene layout. At the same time, providing the full image is both expensive and unnecessary, since most of the important context information is contained in the object's vicinity. Instead, we dynamically crop the original image and corresponding binary object mask to the region around the object during training, allowing for anywhere from 20% to 100% padding around the object. We encode the obtained mask $\mathbf{M} \in \{0,1\}^{d\times d}$ using a learnable convolutional layer and inject the resulting feature map by adding it to the image features. This design allows the model to exploit both foreground and background cues when generating the object's structure and pose.

Pointmap conditioning. Many generative reconstruction methods rely on post-hoc pose optimization since their models do not directly leverage depth information. To overcome this limitation, we introduce *pointmap conditioning*, enabling the structure generator $\boldsymbol{\mathcal{G}}_{\text{SP}}$ to utilize depth cues directly. The pointmap $\mathbf{P} \in \mathbb{R}^{d\times d\times 3}$ is a convenient camera-invariant representation formed by recovering the missing spatial coordinates from the depth map $\mathbf{D} \in \mathbb{R}^{d\times d}$ using camera intrinsics $\mathbf{K} \in \mathbb{R}^{3\times 3}$. It is encoded through a learnable layer, and its feature map is added to the image features, providing explicit geometric grounding. This conditioning improves both pose accuracy and shape consistency without additional optimization. As depth can range drastically in the scene, we filter out all background pixels using provided object masks $\mathbf{M}$: $\mathbf{P}_{\text{obj}} = \mathbf{M} \cdot \mathbf{P}$. We further normalize the pointmap using its scale s_{obj} and its translation $\boldsymbol{t}_{\text{obj}}$ to unify the input scale for our network. For translation, we use a robust estimate of the object center (median pointmap value on each dimension). For scale, we use the distance between the 5-th and 95-th percentile of point norms from the median center. The final pointmaps are obtained with $\mathbf{P}^{\text{norm}}_{\text{obj}} = \frac{\mathbf{P}_{\text{obj}} - \boldsymbol{t}_{\text{obj}}}{s_{\text{obj}}}$, mapping the object into $[0,1]^3$. This way, background depth noise in the image does not affect the predictions, making the model more robust to real-world usage.

3.1.2 High-Fidelity Asset Recovery.

In the second stage, we generate the local latents $\{\boldsymbol{z}_i\}_{i=1}^{L}$ conditioned on the sparse structure and predicted pose using a sparse transformer $\boldsymbol{\mathcal{G}}_{\text{L}}$. To enhance efficiency, we pack the latents within 2^3 spatial neighborhoods using sparse convolutions [63] before serialization, as in DiT [59]. The packed sequence is processed through time-modulated transformer blocks, followed by a convolutional upsampling head with skip connections to preserve spatial detail. As in $\boldsymbol{\mathcal{G}}_{\text{SP}}$, timesteps are integrated using AdaLN, and multimodal conditioning is applied via cross-attention layers. Critically, the predicted pose $\mathbf{T}$ from Stage 1 is encoded through a learnable linear layer and concatenated with image, mask, and pointmap features. This pose conditioning is essential for symmetric objects with view-dependent appearance (e.g., cylindrical containers with labels), where only the pose provides the necessary grounding to generate $\boldsymbol{z}$ with appearance details consistent with the object's orientation. The resulting structured latents $\boldsymbol{z} = \{(\boldsymbol{z}_i, \boldsymbol{p}_i)\}_{i=1}^{L}$ are decoded by a mesh decoder $\boldsymbol{\mathcal{D}}_{\text{M}}$, which

extracts geometry via FlexiCubes [64], and a Gaussian Splatting (GS) decoder $\boldsymbol{\mathcal{D}}_{\mathrm{GS}}$, which produces a set of colored 3D Gaussians capturing appearance. To obtain a textured mesh, the GS representation is rendered from multiple viewpoints and the resulting images are baked onto the mesh.

Training and losses. Both $\boldsymbol{\mathcal{G}}_{\mathrm{SP}}$ and $\boldsymbol{\mathcal{G}}_{\mathrm{L}}$ are trained independently using the Conditional Flow Matching (CFM) objective from [58]. For $\boldsymbol{\mathcal{G}}_{\mathrm{SP}}$, we jointly optimize structure and pose using a weighted combination:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{CFM}}(\boldsymbol{S}) + \alpha \cdot \mathcal{L}_{\mathrm{CFM}}(\tilde{\mathbf{T}}),$$

where $\alpha = 0.01$ balances pose prediction with structure generation. We employ synthetic datasets with known ground-truth shapes and poses to supervise both spatial alignment and geometric reconstruction, ensuring consistency across the two generative stages.

Extension to Multiple Views. The vast majority of generative shape reconstruction methods resort to recovering shapes from a single image. This setup demonstrates exciting abilities of generative methods at recovering unobserved object parts using the learned object prior. However, practical real-world robotics and reconstruction setups commonly utilize multiple cameras allowing to alleviate uncertainty imposed by ambiguity in object symmetry and occlusion. To address this and increase the practical value of the method, we extend our training to the multi-view regime by adding an optional second image, pointmap, and mask tuple $\mathbf{I}^{(2)}$, $\mathbf{P}^{(2)}$, $\mathbf{M}^{(2)}$ with the corresponding pose $\mathbf{T}^{(2)}$. The per-view conditioning features are concatenated along the sequence dimension, and a learnable frame token embedding is added to each view's patches so the cross-attention layers can distinguish their origin. Similarly, since the model now predicts one pose per view, each pose output token receives a learnable view id embedding to disambiguate the two predictions. During training, we drop the second view and its pose with probability $p_{\mathrm{drop}} = 0.33$, allowing the network to leverage all available information while retaining single-view inference capability.

3.2 RecGen Dataset.

Our RecGen dataset leverages 198K high-quality 3D assets from 6 public objects and parts datasets. In particular, we use objects assets from: Objaverse-XL [65], ABO [66], and HSSD [67] and part assets from: PhysXNet [68], PartNext [69] and articulated parts from PartNet-Mobility [70]. For the object-based datasets, we create compositional scenes where other assets from the same dataset were randomly placed in the scene to create natural occlusions and non-trivial depth patterns. For the Part-based scenes, we used a single object, as their parts are often severely self-occluded. Each scene is rendered into 20 images with random camera poses, resulting in a diverse set of viewpoints and lighting conditions. The dataset contains a total of 3.2M synthetically generated RGB images, depth maps, segmentation masks, GT poses, and stereo depth maps of 198K scenes with and without occlusions. For the training of the appearance generation, we excluded PartNet-Mobility and PhysXNet subsets due to the lower quality of the provided texture. Some sample assets from the 6 datasets are shown in Fig. 3.

3.3 Implementation Details

RecGen adopts the rectified flow transformer architecture proposed in [10]. We use classifier-free guidance (CFG) with a drop rate of 0.1 and AdamW optimizer with a learning rate of 1e-4. The model is trained for 55K iterations with a batch size of 512 on 64 NVIDIA H100 GPUs. The training process takes approximately 48 hours. During inference, we use a CFG scale of 3.0 and perform 50 denoising steps. All experiments are performed with the `TRELLIS-image-large` [10] model as the base representation, starting from their pretrained network, which has around 1.2 billion parameters.

4 Experiments

Evaluation datasets. We evaluate our method and baselines on four object-based datasets (LM-O [71], HB [72], HOPE [73], and ReOcS [74]) and one part-based dataset (ArtVIP [75]) for shape and pose estimation. The selected object datasets are widely used for benchmarking 6DoF pose estimation and provide GT meshes [76]. Each dataset represents distinct challenges: LM-O and HB include diverse objects and highly

Table 1 Quantitative comparison on object and part datasets, RecGen achieves the best performance on all metrics across all datasets.

	Dataset	Model	CD_{norm} (↓)	ADD-SB (↓)	ADD-SB @0.1 (↑)	ADD-SB @0.05 (↑)	DRE @0.05 (↑)
Objects	HB	SceneComplete	0.234	0.258	65.2%	35.1%	0.0%
		Any6D (InstantMesh)	0.074	0.111	68.6%	36.4%	33.6%
		Any6D (Trellis)	0.106	0.157	47.8%	33.8%	26.8%
		SAM3D	0.033	0.062	92.4%	54.6%	34.6%
		RecGen (1-view)	**0.032**	**0.049**	**95.0%**	**73.8%**	**51.5%**
		RecGen (2-view)	0.029	0.048	95.4%	74.2%	50.9%
	ReOcS	SceneComplete	0.764	0.774	38.2%	26.1%	0.0%
		Any6D (InstantMesh)	0.055	0.066	89.5%	60.8%	60.5%
		Any6D (Trellis)	0.068	0.088	75.5%	47.4%	47.1%
		SAM3D	0.026	0.057	96.2%	43.6%	25.8%
		RecGen (1-view)	**0.019**	**0.032**	**100.0%**	**89.5%**	**60.8%**
		RecGen (2-view)	0.018	0.032	99.7%	91.1%	62.4%
	LMO	SceneComplete	0.186	0.222	50.0%	11.3%	0.0%
		Any6D (InstantMesh)	0.100	0.148	42.2%	11.3%	19.0%
		Any6D (Trellis)	0.116	0.196	29.6%	16.9%	15.5%
		SAM3D	0.057	0.110	64.1%	17.6%	34.5%
		RecGen (1-view)	**0.050**	**0.068**	**83.1%**	**50.0%**	**38.0%**
		RecGen (2-view)	0.056	0.075	83.1%	55.6%	37.3%
Parts	ArtVIP	SceneComplete	0.189	0.201	57.2%	34.0%	0.6%
		Any6D (InstantMesh)	0.089	0.100	61.3%	39.1%	16.2%
		Any6D (Trellis)	0.090	0.106	58.0%	37.7%	16.8%
		SAM3D	0.056	0.073	79.2%	45.8%	22.6%
		RecGen (1-view)	**0.026**	**0.034**	**96.4%**	**84.0%**	**24.4%**
		RecGen (2-view)	0.024	0.032	96.4%	86.4%	24.8%

occluded scenes, while HOPE and ReOcS contain multiple symmetric objects with complex textures. In addition, these datasets are captured with different depth sensors (structured light, time of flight, and stereo) allowing us to evaluate the robustness of the baselines across sensor types. Detailed descriptions of each evaluation dataset are provided in App. F.2.

Since most of the real world pose estimation benchmarks are object-based and datasets with part annotations are scarce, we introduce a part-based benchmark derived from ArtVIP [75], a collection of digital assets for high fidelity physical interaction in robot learning. The original dataset contains six static scenes; we extend it with six additional scenes featuring new objects. From these scenes, we select 284 object parts and render 924 high-quality RGB-D images. Further details on the benchmark construction are provided in App. F.3.

Evaluation metrics. To evaluate 6D pose estimation accuracy, we use the ADD-S metric. Since GT object meshes are generated rather than provided, we follow SAM3D [7] and adopt a bidirectional variant of ADD-S (denoted *ADD-SB*), which computes symmetric distances between the predicted and GT posed meshes. We report results at the standard 10% object diameter threshold and additionally at 5% ($ADD-SB@5\%$), as the former saturates on simpler datasets.

To assess robustness to occlusions, we introduce the Diameter Relative Error (DRE) metric. Although estimating object size is straightforward for fully visible objects with depth, it becomes significantly more challenging under heavy partial occlusions, where only a small portion is observed. We define DRE as $e_d = |d_{\text{pred}} - d_{\text{gt}}|/d_{\text{gt}}$, where d_{pred} and d_{gt} denote the predicted and GT diameters. We report $DRE@0.05$, the fraction of samples with $e_d < 5\%$.

To evaluate surface reconstruction quality, we compute Chamfer Distance (CD) after ICP alignment to GT mesh and normalize it by the GT diameter to ensure equal weighting across object sizes. The ICP step helps disentangle shape errors from pose inaccuracies.

Finally, to assess the visual fidelity of the reconstructions, we render the predicted shapes from predefined views and report standard perceptual metrics PSNR, SSIM, and LPIPS.

Figure 5 Qualitative comparison on symmetric objects. Our method generates textures consistent with the given pose, whereas SAM3D often produces incorrect textures because its appearance generation depends only on object shape, not pose.

Baselines. We compare RecGen against baselines for model-free pose estimation [12], scene completion [14], and 3D reconstruction from single images [7]. Any6D [12] is a model-free pose estimation method that generates a mesh using InstantMesh [16] and refines the scale and pose via a full-to-partial matching strategy. Although the authors propose an anchor-query approach, we treat the two views as equivalent in our experiments. Additionally, we evaluate a variant using TRELLIS [10] for mesh generation. For scene completion, we evaluate SceneComplete [14], which leverages inpainting to generate an occlusion-free mesh. The scale is estimated via feature matching, and the pose is refined using FoundationPose [15]. Finally, we evaluate SAM3D [7], the closest related approach, as it simultaneously estimates both mesh and pose for objects. A detailed description of each baseline and its usage is provided in Appendix E.

4.1 Pose and Shape Estimation for Objects and Parts

Table 1 reports shape quality (CD_{norm}) and pose estimation accuracy (ADD-SB) on both object-centric and part-level benchmarks.

Both RecGen and SAM3D outperform SceneComplete and Any6D, highlighting the importance of joint shape and pose training on large-scale datasets with occlusion. On object-centric benchmarks, RecGen (1-view) achieves an average CD_{norm} of 0.033 vs. 0.039 for SAM3D and 0.076 for Any6D. The 2-view variant further improves shape generation (CD_{norm} is 0.034 for objects, 0.024 for parts), enabling more accurate reconstruction in standard robotics setups where more than one RGB-D camera is available [77]. Inference-time pose-selection and a multi-sample selection strategies that further improve the two-view results are discussed in App. B.1 and App. B.2.

With respect to pose estimation, RecGen outperforms all baselines, including the state-of-the-art SAM3D. On object-centric benchmarks, RecGen (1-view) reaches 92.7% ADD-SB @0.1 and 71.1% @0.05 on average, compared to an average of 84.2% / 38.6% for SAM3D—nearly doubling accuracy at the stricter threshold. The 2-view variant enhances the average performance to 73.6% @0.05 (see also Table S1). RecGen's ability to accurately predict full object scale, even on occluded samples, can be further seen in the DRE metric with significant improvements over SAM3D, as qualitatively visible in Fig. 4. Robustness to occlusion severity is highlighted in Fig. 6: on object-based datasets the gap to SAM3D widens from 0.044 vs.

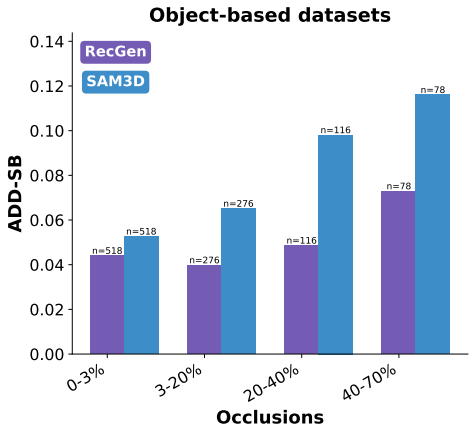


Figure 6 Robustness to occlusions. ADD-SB (lower is better) on object-based datasets (HB+LMO+ReOcS), binned by occlusion severity. RecGen's advantage widens as occlusion increases.

Table 2 Perception quality comparison before and after ICP.

Dataset	Model	Before ICP			After ICP		
		LPIPS (↓)	SSIM (↑)	PSNR (↑)	LPIPS (↓)	SSIM (↑)	PSNR (↑)
LMO+HB+HOPE	Any6D (InstantMesh)	0.225	**0.835**	15.46	0.230	0.825	15.20
	Any6D (Trellis)	0.263	0.829	14.56	0.257	0.820	14.48
	SAM3D	0.219	0.821	15.72	**0.161**	**0.841**	**17.42**
	RecGen (1-view)	**0.199**	0.825	**15.85**	0.170	0.834	16.54
	RecGen (2-view)	0.199	0.824	15.82	0.166	0.835	16.62
Symmetric	Any6D (InstantMesh)	0.193	**0.834**	16.31	0.201	0.822	15.96
	Any6D (Trellis)	0.190	0.834	**16.45**	0.187	**0.829**	16.50
	SAM3D	0.201	0.815	16.02	0.156	0.828	**17.21**
	RecGen (1-view)	**0.170**	0.816	15.63	**0.142**	0.827	16.12
	RecGen (2-view)	0.172	0.817	15.59	0.144	0.830	16.08

0.053 at 0–3% occlusion to 0.073 vs. 0.116 at 40–70% occlusion (a 37% relative improvement). A more complete analysis including Chamfer distance and the parts-based AV dataset is provided in App. A.1; a per-object breakdown on HB is reported in App. A.2, and additional qualitative comparisons are shown in App. G.1.
Reconstructing and localizing articulated object parts is a particularly challenging task that requires fine-grained geometric understanding. Although part geometries are often simpler than full objects, their recovery requires inferring the underlying shape under severe self-occlusion and benefits greatly from part-aware joint shape and pose prediction training. RecGen outperforms all baselines by a large margin on the ArtVIP part-level benchmark: for part-shape reconstruction, RecGen (1-view) reduces CD_{norm} by half compared to SAM3D (0.056 → 0.026); for part-pose estimation, it improves ADD-SB @0.05 by +38.2pp (45.8% → 84.0%), establishing state-of-the-art in both tasks. This capability makes RecGen ideal for extending Real-to-Sim-to-Real [49] beyond object rearrangement to articulated object manipulation. Additional qualitative comparisons on ArtVIP parts are shown in Appendix G.2.

4.2 Posed Appearance Generation

To evaluate the quality of our pose-aware appearance generation, we use the HB, LM-O, and HOPE datasets, which provide textures for the GT object meshes. For each predicted sample, we transform it using the predicted pose and render the generated appearance, then compare it with the GT object transformed with the GT pose and rendered from the same view. To disentangle the effects of pose estimation errors, we perform ICP alignment using GT shapes as references.
In Table 2, we observe that before additional ICP alignment RecGen outperforms the other baselines on average, demonstrating how the combined pose, shape, and appearance estimation leads to a much more faithful overall scene reconstruction. After ICP refinement, both RecGen and SAM3D perform significantly better than other baselines and perform comparably to each other.
We expect that a larger training dataset and additional usage of the depth during training of the encoder-decoder (Depth-VAE from SAM3D), as well as integrating multi-resolution training (TRELLIS 2), can further improve the quality of RecGen's pose-aware and multi-view appearance generation.
To additionally study the usage of the poses during appearance generation, we evaluated all methods on a subset of symmetric objects from HOPE and HB. We see a relative improvement in the perceptual similarity measured by the LPIPS metric (see Table 2). To verify the source of the improvement, we additionally perform a VLM-based classification of the orientation alignment based on two images using the GPT-5 model, comparing the GT-posed and rendered objects with the ICP-aligned prediction from the model. As depicted in Fig. 7, RecGen surpasses SAM3D in object texture orientation alignment by a large margin. We attribute this improvement to our pose-conditioned formulation, which enables more accurate texture recovery for symmetric objects by properly aligning textures with input views. Some qualitative results are shown in Fig. 5; a per-object breakdown of the VLM orientation evaluation and additional qualitative examples are provided in App. A.3.

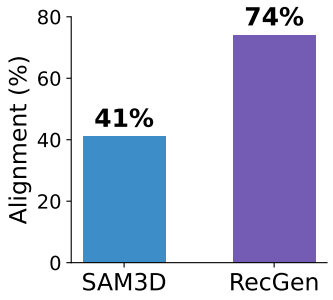


Figure 7 VLM-based evaluation of texture orientation alignment to the GT mesh on symmetric objects.

Table 3 Ablation study for joint shape and pose generation. Object-level results are reported on HB, LM-O, and ReOcS, and part-level results on ArtVIP. Values are shown as mean / median. Differences from the full model are highlighted in green (better), red (worse), and gray (similar).

Variant	Objects-centric		Part-centric	
	CD_{norm} (↓)	ADD-SB (↓)	CD_{norm} (↓)	ADD-SB (↓)
Full model	0.042 / 0.023	0.062 / 0.037	0.033 / 0.020	0.043 / 0.028
w/o stereo	0.048 / 0.030	0.078 / 0.050	0.030 / 0.018	0.039 / 0.027
w/o norm	0.042 / 0.026	0.074 / 0.048	0.038 / 0.025	0.056 / 0.041
w/o part-centric	0.040 / 0.023	0.060 / 0.037	0.073 / 0.037	0.086 / 0.048
w/o pretraining	0.044 / 0.031	0.067 / 0.046	0.044 / 0.028	0.056 / 0.036

4.3 Ablation study

The ablation study in Table 3 validates our design choices, showing that the full model achieves robustness to real-world conditions and generality across object- and part-level reasoning. Given the computational constraints, we train a base and ablated RecGen models for 150K iterations with a batch size of 64.

Stereo Noise. Removal of stereo noise augmentation substantially degrades object-centric metrics (CD 0.048 vs. 0.042, ADD-SB 0.078 vs. 0.062), as the model becomes less resilient to real-world sensor noise. The effect is negligible on synthetic ArtVIP data, where depth is noise-free. Disabling pose normalization has little impact on shape quality (CD remains at 0.042), but significantly hurts pose estimation across both settings (ADD-SB rises to 0.074 and 0.056 from 0.062 and 0.043, respectively), confirming that normalization simplifies the pose learning task and makes joint optimization more stable. Together, stereo augmentation and pose normalization provide complementary robustness — the former at the input level and the latter at the representation level.

Generality. Training without part-level data matches the full model on object-centric benchmarks (CD 0.040, ADD-SB 0.060) but degrades substantially on ArtVIP (CD 0.073 vs. 0.033, ADD-SB 0.086 vs. 0.043). Part supervision is necessary for articulated structures while not compromising object-level performance.

Pretraining and thereafter initializing weights improves all metrics, providing a strong geometric prior for shape reconstruction and joint pose estimation.
Finally, beyond accuracy, we find that RecGen is also substantially more efficient than SAM3D at inference time (1.8× faster, 1.6× less GPU memory); a detailed efficiency comparison is reported in Appendix C. Limitations and future directions are discussed in Appendix D.

5 Conclusion

In this paper, we propose RecGen, a generalist scene completion framework that recovers entirely multi-object shapes from partial input observations. RecGen addresses several long-standing challenges in computer vision: it is robust to occlusions, handles symmetric objects and relative object-parts, generalizes to real-world data despite being trained solely on synthetic data and works robustly across different real-world RGB-D sensors. Unlike many competing approaches, RecGen extends to multiple views, making it more suitable for real-world applications. We demonstrate that RecGen outperforms the competitive SAM3D by 30.1% in geometric shape quality, 9.1% in texture reconstruction, and 33.9% in pose estimation, while using 80% less training data meshes. We believe that RecGen can serve as an easy-to-deploy and easy-to-build-on framework for advancing real-to-sim reconstruction in robotics and other fields.

Acknowledgements

Andrii Zadaianchuk is funded by the European Union (ERC, EVA, 950086).

Contributions

Andrii Zadaianchuk was the main contributor and was responsible for conceptualization, methodology, training data generation, code development, model training, evaluations development, writing, and project direction. Sergey Zakharov provided main supervision and contributed to conceptualization, methodology, training data generation, code development, model training, and project direction. Leonardo Barcellona was a core contributor and was involved in conceptualization, evaluations development, part-based evaluation dataset, evaluation of the baselines, and project direction. Christian Gumbsch contributed to validation and formal analysis and provided important feedback during the project. Lennard Schuenemann contributed to validation and formal analysis. Zehao Wang contributed to visualization and writing. Muhammad Zubair Irshad, Fabien Despinoy, Rahaf Aljundi, and Stratis Gavves contributed to writing, review, editing and provided important feedback during the project.

References

[1] Li, C, Zhang, R, Wong, J, Gokmen, C, Srivastava, S, Martín-Martín, R, Wang, C, Levine, G, Lingelbach, M, Sun, J, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. CoRL. (2023)

[2] Savva, M, Kadian, A, Maksymets, O, Zhao, Y, Wijmans, E, Jain, B, Straub, J, Liu, J, Koltun, V, Malik, J, et al. Habitat: A platform for embodied ai research. ICCV. (2019)

[3] Mittal, M, Roth, P, Tigue, J, Richard, A, Zhang, O, Du, P, Serrano-Munoz, A, Yao, X, Zurbrügg, R, Rudin, N, et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. arXiv:2511.04831 (2025)

[4] Chen, T, Chen, Z, Chen, B, Cai, Z, Liu, Y, Li, Z, Liang, Q, Lin, X, Ge, Y, Gu, Z, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. arXiv:2506.18088 (2025)

[5] Tao, F, Zhang, H, and Zhang, C. Advancements and challenges of digital twins in industry. Nature Computational Science (2024)

[6] Zhu, L, Huang, S, Schindler, K, and Armeni, I. Living scenes: Multi-object relocalization and reconstruction in changing 3d environments. CVPR. (2024)

[7] Chen, X, Chu, FJ, Gleize, P, Liang, KJ, Sax, A, Tang, H, Wang, W, Guo, M, Hardin, T, Li, X, et al. Sam 3d: 3dfy anything in images. arXiv:2511.16624 (2025)

[8] Ikeda, T, Zakharov, S, Ko, T, Irshad, MZ, Lee, R, Liu, K, Ambrus, R, and Nishiwaki, K. Diffusionnocs: Managing symmetry and uncertainty in sim2real multi-modal category-level pose estimation. IROS. (2024)

[9] Liu, R, Wu, R, Van Hoorick, B, Tokmakov, P, Zakharov, S, and Vondrick, C. Zero-1-to-3: Zero-shot one image to 3d object. ICCV. (2023)

[10] Xiang, J, Lv, Z, Xu, S, Deng, Y, Wang, R, Zhang, B, Chen, D, Tong, X, and Yang, J. Structured 3d latents for scalable and versatile 3d generation. CVPR. (2025)

[11] Team, TH. *Hunyuan3D 1.0: A Unified Framework for Text-to-3D and Image-to-3D Generation.* (2024)

[12] Lee, T, Wen, B, Kang, M, Kang, G, Kweon, IS, and Yoon, KJ. Any6D: Model-free 6D Pose Estimation of Novel Objects. CVPR. (2025)

[13] Liu, Y, Wen, Y, Peng, S, Lin, C, Long, X, Komura, T, and Wang, W. Gen6d: Generalizable model-free 6-dof object pose estimation from rgb images. ECCV. (2022)

[14] Agarwal, A, Singh, G, Sen, B, Lozano-Pérez, T, and Kaelbling, LP. SceneComplete: Open-World 3D Scene Completion in Complex Real World Environments for Robot Manipulation. arXiv:2410.23643 (2024)

[15] Wen, B, Yang, W, Kautz, J, and Birchfield, S. Foundationpose: Unified 6d pose estimation and tracking of novel objects. CVPR. (2024)

[16] Xu, J, Cheng, W, Gao, Y, Wang, X, Gao, S, and Shan, Y. InstantMesh: Efficient 3D Mesh Generation from a Single Image with Sparse-view Large Reconstruction Models. arXiv:2404.07191 (2024)
[17] Yu, Q, Yuan, X, Jiang, Y, Chen, J, Zheng, D, Hao, C, You, Y, Chen, Y, Mu, Y, Liu, L, et al. Artgs: 3d gaussian splatting for interactive visual-physical modeling and manipulation of articulated objects. IROS. (2025)
[18] Jiang, T, Guan, Y, Ma, L, Xu, J, Meng, J, Chen, W, Zeng, Z, Li, L, Wu, D, and Chen, R. DexSim2Real^2: Building Explicit World Model for Precise Articulated Object Dexterous Manipulation. IEEE Transactions on Robotics (2025)
[19] Wen, B, Trepte, M, Aribido, J, Kautz, J, Gallo, O, and Birchfield, S. Foundationstereo: Zero-shot stereo matching. CVPR. (2025)
[20] Wang, Z, Wang, Y, Chen, Y, Xiang, C, Chen, S, Yu, D, Li, C, Su, H, and Zhu, J. *CRM: Single Image to 3D Textured Mesh with Convolutional Reconstruction Model.* (2024)
[21] Tang, J, Chen, Z, Chen, X, Wang, T, Zeng, G, and Liu, Z. *LGM: Large Multi-View Gaussian Model for High-Resolution 3D Content Creation.* (2024)
[22] Team, TH. *Hunyuan3D 2.0: Scaling Diffusion Models for High Resolution Textured 3D Assets Generation.* (2025)
[23] Nguyen, VN, Groueix, T, Salzmann, M, and Lepetit, V. Gigapose: Fast and robust novel object pose estimation via one correspondence. CVPR. (2024)
[24] Deng, W, Campbell, D, Sun, C, Zhang, J, Kanitkar, S, Shaffer, ME, and Gould, S. Pos3R: 6D Pose Estimation for Unseen Objects Made Easy. CVPR. (2025)
[25] Liu, K, Zakharov, S, Chen, D, Ikeda, T, Shakhnarovich, G, Gaidon, A, and Ambrus, R. *OmniShape: Zero-Shot Multi-Hypothesis Shape and Pose Estimation in the Real World.* (2025)
[26] Schonberger, JL and Frahm, JM. Structure-from-motion revisited. CVPR. (2016)
[27] Ardelean, A, Özer, M, and Egger, B. *Gen3DSR: Generalizable 3D Scene Reconstruction via Divide and Conquer from a Single View.* (2025)
[28] Irshad, MZ, Kollar, T, Laskey, M, Stone, K, and Kira, Z. CenterSnap: Single-Shot Multi-Object 3D Shape Reconstruction and Categorical 6D Pose and Size Estimation. ICRA. (2022)
[29] Irshad, MZ, Zakharov, S, Ambrus, R, Kollar, T, Kira, Z, and Gaidon, A. ShAPO: Implicit Representations for Multi-Object Shape Appearance and Pose Optimization. ECCV. (2022)
[30] Li, Y, Zhang, J, Chen, Z, Wang, Z, and Liu, Z. MIDI: Multi-Instance Diffusion for Single Image to 3D Scene Generation. CVPR. (2025)
[31] Chen, M, Shapovalov, R, Laina, I, Monnier, T, Wang, J, Novotny, D, and Vedaldi, A. PartGen: Part-level 3D Generation and Reconstruction with Multi-View Diffusion Models. CVPR. (2025)
[32] He, X, Wu, Y, Guo, X, Ye, C, Zhou, J, Hu, T, Han, X, and Du, D. UniPart: Part-Level 3D Generation with Unified 3D Geom–Seg Latents. arXiv:2512.09435 (2026)
[33] Zhang, L, Zhang, Q, Jiang, H, Bai, Y, Yang, W, Xu, L, and Yu, J. BANG: Dividing 3D Assets via Generative Exploded Dynamics. ACM TOG (2025)
[34] Lin, Y, Lin, C, Pan, P, Yan, H, Feng, Y, Mu, Y, and Fragkiadaki, K. *PartCrafter: Structured 3D Mesh Generation via Compositional Latent Diffusion Transformers.* (2025)
[35] Melnik, A, Alt, B, Nguyen, G, Wilkowski, A, Stefańczyk, M, Wu, Q, Harms, S, Rhodin, H, Savva, M, and Beetz, M. *Digital Twin Generation from Visual Data: A Survey.* (2026)
[36] Irshad, MZ, Comi, M, Lin, YC, Heppert, N, Valada, A, Ambrus, R, Kira, Z, and Tremblay, J. *Neural Fields in Robotics: A Survey.* (2024)
[37] Kerbl, B, Kopanas, G, Leimkühler, T, and Drettakis, G. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. ACM TOG (2023)
[38] Yu, J, Hari, K, El-Refai, K, Dalil, A, Kerr, J, Kim, CM, Cheng, R, Irshad, MZ, and Goldberg, K. Persistent Object Gaussian Splat (POGS) for Tracking Human and Robot Manipulation of Irregularly Shaped Objects. ICRA (2025)
[39] Qureshi, MN, Garg, S, Yandun, F, Held, D, Kantor, G, and Silwal, A. *SplatSim: Zero-Shot Sim2Real Transfer of RGB Manipulation Policies Using Gaussian Splatting.* (2024)
[40] Shorinwa, O, Tucker, J, Smith, A, Swann, A, Chen, T, Firoozi, R, Kennedy, MD, and Schwager, M. Splat-MOVER: Multi-Stage, Open-Vocabulary Robotic Manipulation via Editable Gaussian Splatting (2024)
[41] Abou-Chakra, J, Rana, K, Dayoub, F, and Sünderhauf, N. *Physically Embodied Gaussian Splatting: A Realtime Correctable World Model for Robotics.* (2024)

[42] Ji, M, Qiu, RZ, Zou, X, and Wang, X. GraspSplats: Efficient Manipulation with 3D Feature Splatting. arXiv:2409.02084 (2024)
[43] Chhablani, G, Ye, X, Irshad, MZ, and Kira, Z. *EmbodiedSplat: Personalized Real-to-Sim-to-Real Navigation with Gaussian Splats from a Mobile Device*. (2025)
[44] Escontrela, A, Kerr, J, Allshire, A, Frey, J, Duan, R, Sferrazza, C, and Abbeel, P. *GaussGym: An open-source real-to-sim framework for learning locomotion from pixels*. (2025)
[45] Shen, W, Yang, G, Yu, A, Wong, J, Kaelbling, LP, and Isola, P. Distilled feature fields enable few-shot language-guided manipulation. arXiv:2308.07931 (2023)
[46] Yang, S, Yu, W, Zeng, J, Lv, J, Ren, K, Lu, C, Lin, D, and Pang, J. *Novel Demonstration Generation with Gaussian Splatting Enables Robust One-Shot Manipulation*. (2025)
[47] Jiang, G, Chang, H, Qiu, RZ, Liang, Y, Ji, M, Zhu, J, Dong, Z, Zou, X, and Wang, X. GSWorld: Closed-Loop Photo-Realistic Simulation Suite for Robotic Manipulation. arXiv:2510.20813 (2025)
[48] Dan, P, Kedia, K, Chao, A, Duan, E, Pace, MA, Ma, WC, and Choudhury, S. X-Sim: Cross-Embodiment Learning via Real-to-Sim-to-Real. CoRL. (2025)
[49] Barcellona, L, Zadaianchuk, A, Allegro, D, Papa, S, Ghidoni, S, and Gavves, E. Dream to Manipulate: Compositional World Models Empowering Robot Imitation Learning with Imagination. ICLR. (2025)
[50] Yu, J, Fu, L, Huang, H, El-Refai, K, Ambrus, RA, Cheng, R, Irshad, MZ, and Goldberg, K. *Real2Render2Real: Scaling Robot Data Without Dynamics Simulation or Robot Hardware*. (2025)
[51] Kapelyukh, I, Zhang, X, James, S, Herlant, L, and Johns, E. ZeroBot: Learning From Scratch in Minutes With Generative Real2Sim. RA-L (2026)
[52] Zhang, K, Sha, S, Jiang, H, Loper, M, Song, H, Cai, G, Xu, Z, Hu, X, Zheng, C, and Li, Y. Real-to-Sim Robot Policy Evaluation with Gaussian Splatting Simulation of Soft-Body Interactions. ICRA. (2026)
[53] Jangir, Y, Zhang, Y, Lo, PC, Yamazaki, K, Zhang, C, Tu, KH, Ke, TW, Ke, L, Bisk, Y, and Fragkiadaki, K. *RobotArena ∞: Scalable Robot Benchmarking via Real-to-Sim Translation*. (2025)
[54] Jain, A, Zhang, M, Arora, K, Chen, W, Torne, M, Irshad, MZ, Zakharov, S, Wang, Y, Levine, S, Finn, C, Ma, WC, Shah, D, Gupta, A, and Pertsch, K. *PolaRiS: Scalable Real-to-Sim Evaluations for Generalist Robot Policies*. (2025)
[55] Yu, X, Talak, R, Shaikewitz, L, and Carlone, L. Picasso: Holistic Scene Reconstruction with Physics-Constrained Sampling. arXiv:2602.08058 (2026)
[56] Xiang, T, Cao, J, Guo, S, Zhao, G, Luo, AF, and Ma, J. *Real-to-Sim for Highly Cluttered Environments via Physics-Consistent Inter-Object Reasoning*. (2026)
[57] Huang, WC, Han, J, Ye, X, Pan, Z, and Hauser, K. *Simulation-Ready Cluttered Scene Estimation via Physics-aware Joint Shape and Pose Optimization*. (2026)
[58] Lipman, Y, Chen, RT, Ben-Hamu, H, Nickel, M, and Le, M. Flow Matching for Generative Modeling. ICLR. (2023)
[59] Peebles, W and Xie, S. Scalable diffusion models with transformers. ICCV. (2023)
[60] Oquab, M, Darcet, T, Moutakanni, T, Vo, HV, Szafraniec, M, Khalidov, V, Fernandez, P, Haziza, D, Massa, F, El-Nouby, A, Howes, R, Huang, PY, Xu, H, Sharma, V, Li, SW, Galuba, W, Rabbat, M, Assran, M, Ballas, N, Synnaeve, G, Misra, I, Jegou, H, Mairal, J, Labatut, P, Joulin, A, and Bojanowski, P. DINOv2: Learning Robust Visual Features without Supervision. arXiv:2304.07193 (2023)
[61] Geist, AR, Frey, J, Zhobro, M, Levina, A, and Martius, G. Learning with 3D rotations, a hitchhiker's guide to SO (3). arXiv:2404.11735 (2024)
[62] Zhou, Y, Barnes, C, Lu, J, Yang, J, and Li, H. On the continuity of rotation representations in neural networks. CVPR. (2019)
[63] Wang, PS, Liu, Y, Guo, YX, Sun, CY, and Tong, X. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. ACM TOG (2017)
[64] Shen, T, Munkberg, J, Hasselgren, J, Yin, K, Wang, Z, Chen, W, Gojcic, Z, Fidler, S, Sharp, N, and Gao, J. Flexible Isosurface Extraction for Gradient-Based Mesh Optimization. ACM TOG (2023)
[65] Deitke, M, Liu, R, Wallingford, M, Ngo, H, Michel, O, Kusupati, A, Fan, A, Laforte, C, Voleti, V, Gadre, SY, VanderBilt, E, Kembhavi, A, Vondrick, C, Gkioxari, G, Ehsani, K, Schmidt, L, and Farhadi, A. Objaverse-XL: A Universe of 10M+ 3D Objects. arXiv:2307.05663 (2023)
[66] Collins, J, Goel, S, Deng, K, Luthra, A, Xu, L, Gundogdu, E, Zhang, X, Yago Vicente, TF, Dideriksen, T, Arora, H, Guillaumin, M, and Malik, J. ABO: Dataset and Benchmarks for Real-World 3D Object Understanding. CVPR (2022)

[67] Khanna*, M, Mao*, Y, Jiang, H, Haresh, S, Shacklett, B, Batra, D, Clegg, A, Undersander, E, Chang, AX, and Savva, M. Habitat Synthetic Scenes Dataset (HSSD-200): An Analysis of 3D Scene Scale and Realism Tradeoffs for ObjectGoal Navigation. arXiv (2023)

[68] Cao, Z, Chen, Z, Pan, L, and Liu, Z. PhysX-3D: Physical-Grounded 3D Asset Generation. arXiv:2507.12465 (2025)

[69] Wang, P, He, Y, Lv, X, Zhou, Y, Xu, L, Yu, J, and Gu, J. Partnext: A next-generation dataset for fine-grained and hierarchical 3d part understanding. arXiv:2510.20155 (2025)

[70] Xiang, F, Qin, Y, Mo, K, Xia, Y, Zhu, H, Liu, F, Liu, M, Jiang, H, Yuan, Y, Wang, H, Yi, L, Chang, AX, Guibas, LJ, and Su, H. SAPIEN: A SimulAted Part-based Interactive ENvironment. CVPR. (2020)

[71] Brachmann, E, Krull, A, Michel, F, Gumhold, S, Shotton, J, and Rother, C. Learning 6d object pose estimation using 3d object coordinates. ECCV. (2014)

[72] Kaskman, R, Zakharov, S, Shugurov, I, and Ilic, S. Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects. ICCVW. (2019)

[73] Tyree, S, Tremblay, J, To, T, Cheng, J, Mosier, T, Smith, J, and Birchfield, S. 6-DoF Pose Estimation of Household Objects for Robotic Manipulation: An Accessible Dataset and Benchmark. IROS. (2022)

[74] Iwase, S, Irshad, MZ, Liu, K, Guizilini, V, Lee, R, Ikeda, T, Amma, A, Nishiwaki, K, Kitani, K, Ambrus, R, et al. ZeroGrasp: Zero-shot shape reconstruction enabled robotic grasping. CVPR. (2025)

[75] Jin, Z, Che, Z, Zhao, Z, Wu, K, Zhang, Y, Zhao, Y, Liu, Z, Zhang, Q, Ju, X, Tian, J, et al. Artvip: Articulated digital assets of visual realism, modular interaction, and physical fidelity for robot learning. arXiv:2506.04941 (2025)

[76] Hodan, T, Michel, F, Brachmann, E, Kehl, W, GlentBuch, A, Kraft, D, Drost, B, Vidal, J, Ihrke, S, Zabulis, X, et al. BOP: Benchmark for 6D object pose estimation. ECCV. (2018)

[77] Khazatsky, A, Pertsch, K, Nair, S, Balakrishna, A, Dasari, S, Karamcheti, S, Nasiriany, S, Srirama, MK, Chen, LY, Ellis, K, Fagan, PD, Hejna, J, Itkina, M, Lepert, M, Ma, YJ, Miller, PT, Wu, J, Belkhale, S, Dass, S, Ha, H, Jain, A, Lee, A, Lee, Y, Memmel, M, Park, S, Radosavovic, I, Wang, K, Zhan, A, Black, K, Chi, C, Hatch, KB, Lin, S, Lu, J, Mercat, J, Rehman, A, Sanketi, PR, Sharma, A, Simpson, C, Vuong, Q, Walke, HR, Wulfe, B, Xiao, T, Yang, JH, Yavary, A, Zhao, TZ, Agia, C, Baijal, R, Castro, MG, Chen, D, Chen, Q, Chung, T, Drake, J, Foster, EP, Gao, J, Guizilini, V, Herrera, DA, Heo, M, Hsu, K, Hu, J, Irshad, MZ, Jackson, D, Le, C, Li, Y, Lin, K, Lin, R, Ma, Z, Maddukuri, A, Mirchandani, S, Morton, D, Nguyen, T, O'Neill, A, Scalise, R, Seale, D, Son, V, Tian, S, Tran, E, Wang, AE, Wu, Y, Xie, A, Yang, J, Yin, P, Zhang, Y, Bastani, O, Berseth, G, Bohg, J, Goldberg, K, Gupta, A, Gupta, A, Jayaraman, D, Lim, JJ, Malik, J, Martín-Martín, R, Ramamoorthy, S, Sadigh, D, Song, S, Wu, J, Yip, MC, Zhu, Y, Kollar, T, et al. DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset (2024)

[78] Xiang, J, Chen, X, Xu, S, Wang, R, Lv, Z, Deng, Y, Zhu, H, Dong, Y, Zhao, H, Yuan, NJ, and Yang, J. Native and Compact Structured Latents for 3D Generation. Tech report (2025)

[79] Geng, Z, Wang, N, Xu, S, Ye, C, Li, B, Chen, Z, Peng, S, and Zhao, H. One View, Many Worlds: Single-Image to 3D Object Meets Generative Domain Randomization for One-Shot 6D Pose Estimation. arXiv:2509.07978 (2025)

[80] Denninger, M, Sundermeyer, M, Winkelbauer, D, Zidan, Y, Olefir, D, Elbadrawy, M, Lodhi, A, and Katam, H. Blenderproc. arXiv:1911.01911 (2019)

[81] Kerr, J, Kim, CM, Wu, M, Yi, B, Wang, Q, Goldberg, K, and Kanazawa, A. Robot See Robot Do: Imitating Articulated Object Manipulation with Monocular 4D Reconstruction. CoRL. (2025)

[82] Le, L, Xie, J, Liang, W, Wang, HJ, Yang, Y, Ma, YJ, Vedder, K, Krishna, A, Jayaraman, D, and Eaton, E. Articulate-Anything: Automatic Modeling of Articulated Objects via a Vision-Language Foundation Model. ICLR. ()

[83] Singh, R, Liu, JJ, Van Wyk, K, Chao, YW, Lafleche, JF, Shkurti, F, Ratliff, N, and Handa, A. Synthetica: Large Scale Synthetic Data Generation for Robot Perception. IROS. (2025)

[84] Han, X, Liu, M, Chen, Y, Yu, J, Lyu, X, Tian, Y, Wang, B, Zhang, W, and Pang, J. Re^3 Sim: Generating High-Fidelity Simulation Data via 3D-Photorealistic Real-to-Sim for Robotic Manipulation. arXiv:2502.08645 (2025)

[85] Dowdy, J and Vaz, JC. Isaac Sim-to-Real: Reinforcement Learning based Locomotion for Quadrupeds. CASE. (2025)

[86] Yu, Q, Moghani, M, Dharmarajan, K, Schorp, V, Panitch, WCH, Liu, J, Hari, K, Huang, H, Mittal, M, Goldberg, K, et al. Orbit-surgical: An open-simulation framework for learning surgical augmented dexterity. ICRA. (2024)
[87] Artykov, A, Boittiaux, C, and Lepetit, V. Articulated Object Understanding from a Single Video Sequence. ICCV. (2025)
[88] Liu, J, Mahdavi-Amiri, A, and Savva, M. Paris: Part-level reconstruction and motion analysis for articulated objects. ICCV. (2023)
[89] Guo, J, Xin, Y, Liu, G, Xu, K, Liu, L, and Hu, R. Articulatedgs: Self-supervised digital twin modeling of articulated objects using 3d gaussian splatting. CVPR. (2025)
[90] Lin, S, Fang, J, Irshad, MZ, Guizilini, VC, Ambrus, RA, Shakhnarovich, G, and Walter, MR. Splart: Articulation estimation and part-level reconstruction with 3d gaussian splatting. ICCV. (2025)
[91] Torne, M, Simeonov, A, Li, Z, Chan, A, Chen, T, Gupta, A, and Agrawal, P. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. arXiv:2403.03949 (2024)

Supplementary Material

In these supplementary materials, we present an additional detailed analysis of the RecGen performance. We show that:

1. RecGen is significantly more robust to partial and severe occlusions than prior work, with performance gaps widening as occlusion increases (App. A.1).
2. The performance gains are consistent across individual object instances, as demonstrated by a per-object breakdown on HB (App. A.2).
3. The pose-conditioned appearance generation of RecGen resolves symmetry ambiguities more reliably than pose-agnostic baselines (App. A.3).
4. Simple inference-time strategies, such as multi-view pose selection and multi-sample alignment-based selection, further improve reconstruction quality without retraining (App. B.1 and App. B.2).
5. RecGen achieves superior computational efficiency compared to SAM3D, requiring less memory and runtime while maintaining higher accuracy (App. C).

In addition, in Sec. F, we provide additional details about the training dataset construction, the proposed evaluation benchmark, and a comprehensive description of the baseline methods and their usage.
Finally, we discuss RecGen limitations and future work and show additional qualitative results on both object-based (Sec. G.1) and part-based datasets (Sec. G.2).

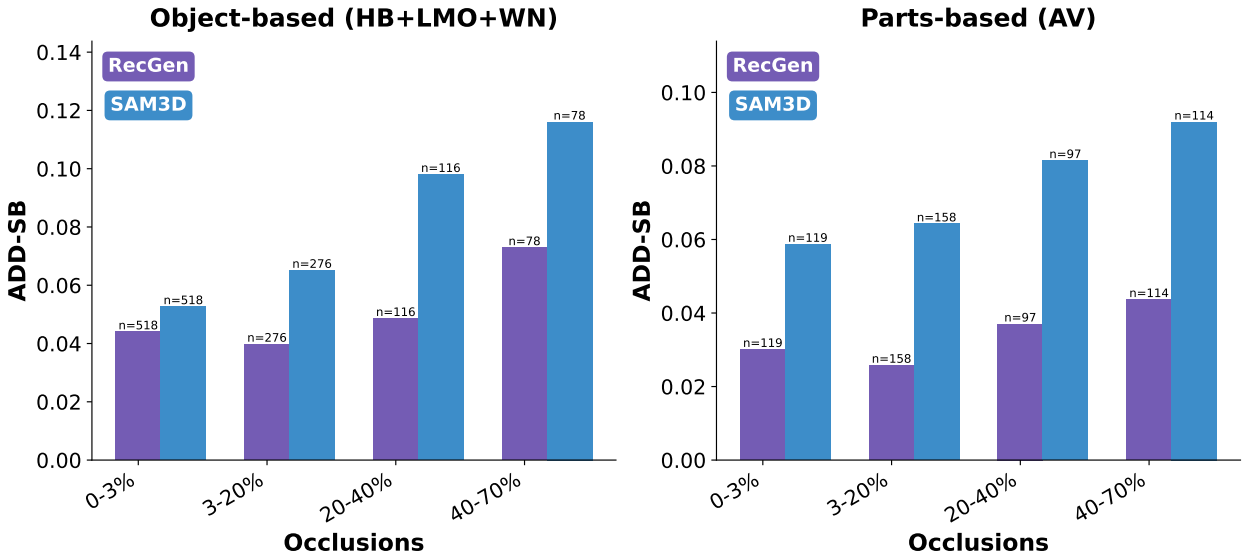
A Additional Analysis of the RecGen Performance

A.1 Robustness to Occlusions

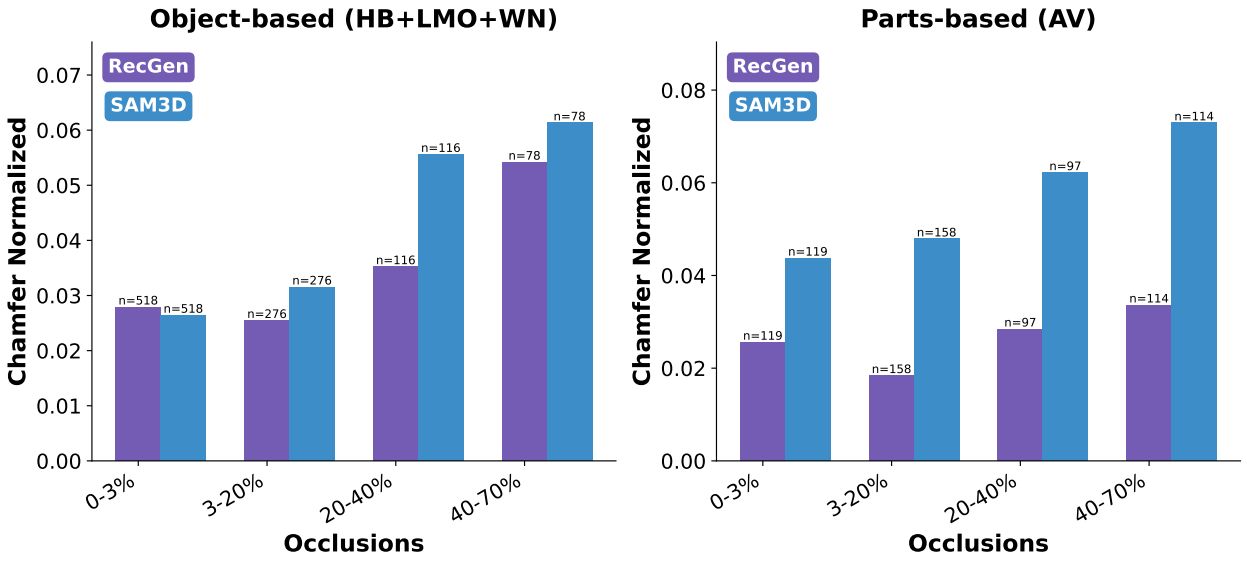
We analyze how pose and shape estimation degrade as the target object becomes increasingly occluded in the input view. For each test sample, we compute the *occlusion fraction* – the ratio of occluded object pixels to total object pixels in the input image, using the ground-truth segmentation masks provided by each dataset. We partition samples into four occlusion bins: 0–3% (nearly fully visible), 3–20%, 20–40%, and 40–70% (severely occluded). We then report the mean ADD-SB and normalized Chamfer scores for RecGen and the best performing baseline, SAM3D, within each bin. Results are aggregated into two groups: *object-based* datasets (HB, LMO, and ReOcS; 994 samples) and the *parts-based* dataset (ArtVIP; 500 samples).
Figure S1 shows reconstruction quality as a function of occlusion severity. On object-based datasets (Fig. S1a, left), RecGen consistently outperforms SAM3D in ADD-SB across all occlusion levels, and the gap widens as occlusion increases: at 0–3% occlusion the difference is modest (0.044 vs. 0.053), but at 40–70% occlusion RecGen achieves 0.073 compared to 0.116 for SAM3D—a 37% relative improvement. The normalized Chamfer distance (Fig. S1b) tells a similar story, with both methods performing comparably on fully visible objects but RecGen maintaining lower error under heavy occlusion.
On the parts-based AV dataset (Fig. S1a, right), the advantage of RecGen is even more pronounced: RecGen's ADD-SB degrades only mildly from 0–3% to 40–70% occlusion, changing by 0.014 (from 0.030 to 0.044), while SAM3D's error increases more drastically by 0.033 (from 0.059 to 0.092). The same trend holds for Chamfer, where RecGen outperforms SAM3D by roughly 2× across all occlusion bins.
These results suggest that RecGen's generative pose estimation combined with robust and object-centric scene normalization as well as usage of the real-world depth sensors during training are highly effective for improving real-world pose and shape estimation though occlusions.

A.2 Per-Object Analysis on HB

We present a detailed per-object breakdown of shape and pose metrics on the HB dataset, comparing RecGen to SAM3D across all 33 objects. We report Chamfer Distance (normalized by object diameter) as a shape quality metric and ADD-SB as a pose accuracy metric; both are lower-is-better. Three outlier samples (2 of SAM3D for `object 21` and 1 of RecGen for `object 16`) with ADD-SB ≥ 0.6 are excluded from both methods symmetrically.
Figure S2 visualizes the per-object comparison. RecGen achieves lower ADD-SB on **29 out of 33** objects and lower Chamfer distance on **26 out of 33** objects, demonstrating consistent improvement across the majority of object instances. The four objects where SAM3D achieves better ADD-SB (objects 12, 15, 16, 31) tend to be



(a) ADD-SB (lower is better) by occlusion severity.



(b) Normalized Chamfer distance (lower is better) by occlusion severity.

Figure S1 Reconstruction quality vs. occlusion severity. Samples are binned by the fraction of the target object visible in the input image. RecGen degrades gracefully as occlusion increases, while SAM3D's error grows substantially. Left: object-based datasets (HB+LMO+ReOcS). Right: parts-based dataset (AV).

cases where RecGen occasionally produces shape artifacts that affect pose alignment, while the underlying geometry predicted by SAM3D happens to be more stable for these specific instances. Notably, objects 15 and 16 are the only cases where SAM3D outperforms RecGen on *both* metrics simultaneously.

A.3 Symmetric Objects

Geometrically symmetric objects pose a unique challenge for joint shape and appearance reconstruction: because the object geometry is identical under some rotations, the predicted mesh can appear correct in terms of shape yet have its texture placed on the wrong side. Methods that generate appearance independently of pose, such as SAM3D [7], are particularly susceptible to this failure mode, as they have no mechanism to resolve which face of the object is visible in the input view. RecGen addresses this through its pose-conditioned

Figure S2 Per-object comparison on HB dataset. Chamfer Distance (top two rows) and ADD-SB (bottom two rows) for each of the 33 HB objects. Lower is better. RecGen (purple) outperforms SAM3D (blue) on 26/33 objects for shape and 29/33 for pose, while both have one outlier object on which they perform significantly worse than on other objects.

formulation, which conditions appearance generation on the estimated pose, enabling the model to assign textures consistently with the observed viewpoint.

To quantify how well the appearance generation network is using pose information, we perform a VLM-based orientation evaluation using GPT-5: for each symmetric object sample, we render the GT-posed object and the posed and additionally ICP-aligned prediction side by side and query the model whether the dominant visual regions (color blocks, labels, graphics) occupy the same spatial positions in both images. Figure S4 breaks down the per-object alignment rates across five symmetric objects from HOPE (objects 3, 8, 12, 25) and HB (object 29). RecGen outperforms SAM3D on every evaluated object, with the largest margin on HOPE object 3 (88% vs. 32%), where the texture is asymmetric along every axis, resulting in many plausible but incorrect orientations for pose-agnostic methods. The overall alignment rate is 74% for RecGen vs. 41% for SAM3D.

Figure S3 provides additional qualitative examples spanning three HOPE objects (3, 12, 25) and HB object 29. RecGen consistently reconstructs textures aligned with the ground-truth orientation, while SAM3D frequently produces flipped or misaligned textures.

B Inference Optimizations

B.1 Multi-view Pose Selection

In the two-view setting, RecGen predicts one 6DoF pose per input view, yielding two candidate poses $\mathbf{T}^{(1)}$ and $\mathbf{T}^{(2)}$ for the same reconstructed mesh $\mathbf{s}$. In the main paper, we report results using only the first pose $\mathbf{T}^{(1)}$. Here, we investigate whether an automatic selection strategy can consistently pick the better candidate at inference time, without access to GT meshes.

Figure S3 Appearance generation for objects with symmetric shapes. Top block: HOPE objects 3 and 12). Bottom block: HOPE object 25 and HB object 29. For each block: input image (top), RecGen reconstruction (middle), SAM3D reconstruction (bottom). RecGen produces textures consistent with the ground-truth orientation across diverse symmetric objects.

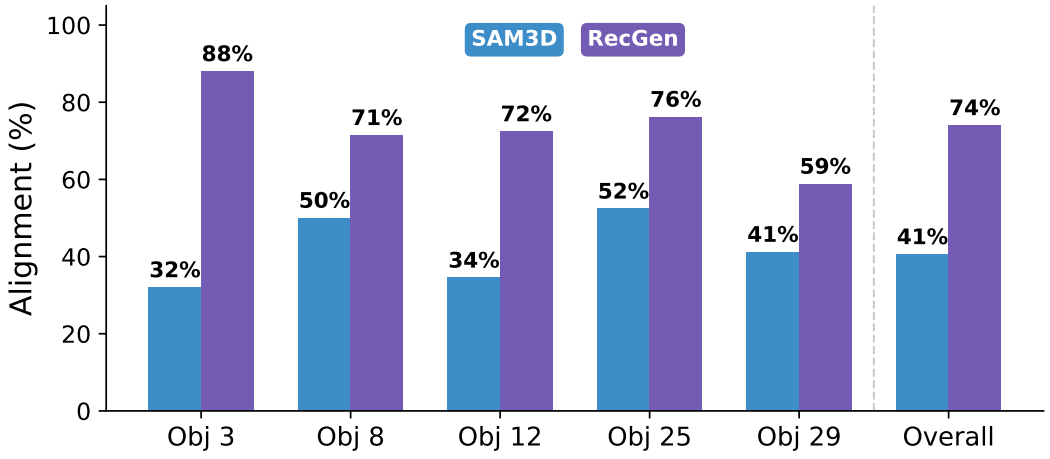


Figure S4 Per-object VLM orientation alignment. Alignment rates for each symmetric object, evaluated by GPT-5. RecGen consistently outperforms SAM3D across all objects. Overall alignment of RecGen on 5 objects (106 input images) is 74% vs. SAM3D 41%.

Table S1 Effect of multi-view pose selection. When two views are available, RecGen predicts two candidate poses. *Single-view alignment* scores each pose against its own view's pointmap in metric camera space. *Cross-view alignment* additionally uses GT relative camera poses to score each candidate against both views' pointmaps. Oracle selects the pose with lowest GT Chamfer distance.

	HB		LMO		ReOcS		ArtVIP		Avg.	
Method	CD_n	ADD-SB	CD_n	ADD-SB	CD_n	ADD-SB	CD_n	ADD-SB	CD_n	ADD-SB
RecGen (1-view)	0.032	0.049	0.051	0.068	0.019	**0.032**	0.026	0.034	0.032	0.046
RecGen (2-view)	0.029	0.048	0.056	0.075	**0.018**	**0.032**	0.024	0.032	0.032	0.047
RecGen (2-view, single-view)	**0.026**	0.043	**0.043**	0.059	0.019	0.033	0.023	0.030	0.028	0.041
RecGen (2-view, cross-view)	**0.026**	**0.042**	**0.043**	**0.057**	**0.018**	**0.032**	**0.022**	**0.029**	**0.027**	**0.040**
RecGen (2-view, oracle)	0.024	0.039	0.039	0.054	0.017	0.030	0.021	0.028	0.025	0.038

We propose a pointmap-based pose selection strategy that operates in metric camera space. For each candidate pose $\mathbf{T}^{(k)}$, we transform the predicted mesh into the camera coordinate frame of view k using the inverse of the pointmap normalization transform, then compute one-directional nearest-neighbor distances from each pointmap point to the mesh surface. A trimmed mean (removing the top 10% of distances) provides robustness to partial visibility. We consider two variants:

- *Single-view alignment*: each pose $\mathbf{T}^{(k)}$ is scored solely by its alignment to its own view's pointmap. The pose with the lower score is selected.
- *Cross-view alignment*: given the GT relative camera pose $\mathbf{T}_{\text{rel}} = \mathbf{T}_{\text{cam}}^{(i)} \circ (\mathbf{T}_{\text{cam}}^{(j)})^{-1}$, each candidate is additionally scored against the other view's pointmap by transforming the mesh into the other camera frame. The per-view scores are averaged and the pose with the lower combined score is selected.

Table S1 reports shape quality (CD_n) and pose accuracy (ADD-SB) for five configurations: single-view, two-view with the first pose, single-view alignment, cross-view alignment, and the oracle (GT-best). Both selection strategies substantially improve over the first-pose baseline on three of four datasets, with the largest gains on LMO (CD_n: 0.056 → 0.043, a 23% reduction) and HB (CD_n: 0.029 → 0.026, 10% reduction). Cross-view alignment, which leverages GT relative camera poses, achieves the best average performance (avg. CD_n: 0.027, ADD-SB: 0.040), outperforming single-view alignment (avg. CD_n: 0.028, ADD-SB: 0.041); both clearly improve over the first-pose baseline (avg. CD_n: 0.032, ADD-SB: 0.047). Notably, on LMO, where the two-view first-pose result is worse than single-view (0.056 vs. 0.051), both selection strategies recover and surpass it, demonstrating that the second pose provides complementary information that the selection mechanism can exploit. On ReOcS, where there are almost no occlusions, using the first pose is comparable or better than selecting between poses, as the original views already provide sufficient information for accurate pose prediction. Overall, we recommend using such pose selection in cases where severe occlusions are possible, as there it could be largest benefit from the additional view predictions.

While such simple strategy as using single-view alignment bridges the gap from single-view prediction to the optimal possible, the oracle row shows substantial remaining headroom (avg. CD_n: 0.025 vs CD_n: 0.028), suggesting that improved selection strategies, potentially leveraging learned scoring functions or multi-view consistency checks, could yield further gains.

Figure S5 provides qualitative examples from LMO illustrating how the second view improves reconstruction quality. In each row, we show the input image alongside novel-view overlays of the ground-truth mesh (grey) and the RecGen posed shape prediction (purple), as well as a Gaussian Splatting render. With only a single view, the predicted shape often deviates from the ground truth in unseen regions. Adding a second view consistently improves the alignment, producing tighter overlaps with the ground-truth geometry and more detailed appearance.

B.2 Multi-sample Generation Evaluation and Alignment-based Sample Selection

Since RecGen's reconstruction pipeline involves a stochastic denoising process, running multiple generations with different random seeds produces diverse shape and pose predictions for the same input. We investigate whether selecting among multiple generations can improve results, analogously to the multi-view pose selection above.

Figure S5 Single-view vs. two-view reconstruction on LMO. Each row shows one object: input image, three novel-view overlays (ground-truth in grey, prediction in purple) and a Gaussian Splatting render for the single-view (left group) and two-view (right group) settings. The second view reduces shape ambiguity, yielding reconstructions that more closely match the ground truth in scale (first row), appearance (second row) and rotations (third row).

Table S2 Effect of multi-sample generation selection on HB (538 samples, 5 seeds). *Single seed* reports the mean ± std across 5 independent generations. *Pointmap alignment* selects, for each instance, the seed whose mesh best aligns with the input view's pointmap in metric camera space. *Oracle* selects the seed with the lowest GT ADD-SB. Lower is better for both metrics.

Method	$CD_n \downarrow$	ADD-SB ↓
RecGen (single seed)	0.031 ± 0.001	0.048 ± 0.001
RecGen (pointmap alignment)	0.029	0.043
RecGen (oracle, best of 5)	0.023	0.037

We evaluate on HB using 5 independent seeds. For each seed, we obtain a full reconstruction with associated metrics. We consider two selection strategies: (1) *pointmap alignment*, which uses the same single-view metric-space alignment score as in the multi-view setting to pick the best generation and could be applied during inference, and (2) *oracle*, which selects the seed with the lowest GT CD_n (to show if one of many generation hypotheses from partial input information is close to GT).
Table S2 reports the results. The single-seed baseline averages $CD_n = 0.031 \pm 0.001$ and ADD-SB $= 0.048 \pm 0.001$ across seeds, showing low variance between runs. Pointmap alignment selection improves to $CD_n = 0.029$ and ADD-SB $= 0.043$, capturing a portion of the oracle gap ($CD_n = 0.023$, ADD-SB $= 0.037$). The oracle best-of-5 result represents a 26% improvement in CD_n and 23% in ADD-SB over a single seed, demonstrating substantial diversity across generations containing samples that are much closer to GT mesh. However, as there is a substantial gap to the optimal selection, more sophisticated selection mechanisms or using the 2-view RecGen are needed for effective selection between the generated samples.

C Inference Efficiency

A key advantage of RecGen's architecture design is that pointmaps and masks are fused additively with DINOv2 features of the inputs into a shared representation, rather than maintained as separate representations. This allows RecGen to be more efficient in terms of the memory and compute speed and allows for extensions to a multi-view version of the RecGen. To confirm this, we measure wall-clock time and peak GPU memory (total process usage for 10 objects from the HB dataset, excluding model loading and post-processing (mesh export, rendering) in comparison to the SAM3D baseline. Table S3 compares the inference cost of RecGen and SAM3D on a single NVIDIA A100-SXM4-80GB GPU. RecGen is 1.8× faster and requires 1.6× less GPU memory than SAM3D.

Table S3 Inference efficiency comparison. Measured on a single NVIDIA A100-SXM4-80GB, averaged over 10 HB samples. *Allocated* is peak PyTorch tensor memory; *Total* is full process GPU usage (`nvidia-smi`).

Method	Allocated Memory (GB) ↓	Total GPU Memory (GB) ↓	Inference Time (s) ↓
SAM3D [7]	17.8 ± 0.4	22.0 ± 1.4	13.0 ± 1.4
RecGen	10.4 ± 0.5	14.1 ± 1.6	7.3 ± 0.2

D Limitations and Future Work

While RecGen demonstrates strong performance across diverse benchmarks, several limitations remain. First, RecGen assumes access to accurate object segmentation masks. When masks are imprecise—for example, including background regions—background depth values can bleed into the object pointmap, corrupting the geometric conditioning signal and degrading both pose and shape estimation. Second, the quality of generated textures and shapes is inherently bounded by the capacity of the underlying TRELLIS VAE for representing assets [10]. While our pose-conditioned appearance generation ensures correct texture orientation, fine-grained surface and geometry details are sometimes lost during the latent encoding and Gaussian Splatting-based decoding pipeline. Incorporating higher-capacity decoders, such as [78], could further improve appearance fidelity. Third, RecGen's inference speed currently limits its applicability to real-time applications. With 50 denoising steps per stage across two generative stages, plus mesh extraction and texture baking, the full pipeline requires several seconds per object on a single GPU. While RecGen is already 1.8× faster than SAM3D (App. C), this remains far from the real-time requirements of interactive robotic manipulation or augmented reality applications.

Future work. Several promising directions emerge from the current work. A natural extension is to generate not only geometric and visual properties but also *physical parameters* such as mass, friction coefficients, collision geometries, and articulation joint types. Enriching the reconstructed assets with these properties would produce simulation-ready digital twins that can be directly imported into physics engines, significantly improving the utility of RecGen for real-to-sim transfer in robot learning pipelines. Another compelling direction is extending RecGen to the *dynamic* setting: given video observations, the model could jointly reconstruct objects and their motion trajectories, enabling scene understanding that captures temporal evolution rather than a single static snapshot. Finally, addressing the limitations outlined above represents important future work: developing end-to-end pipelines that jointly perform segmentation and reconstruction, adopting more expressive appearance decoders from recent advances in 3D generation [10], and exploring distillation or few-step denoising strategies to bring inference times closer to real-time operation.

E Detailed Baseline Description

The problem of simultaneously reconstructing objects and their parts is related to three research directions in recent literature: model-free pose estimation [12], scene completion [14], and single-image 3D reconstruction [7].

Model-free pose estimation. In model-free pose estimation, the goal is to estimate the pose of an object in an image (the query) given a reference view of the same object (the anchor). In several approaches, such as Any6D [12] and OneViewManyWords [79], the target view may coincide with the reference view without affecting the method. For this reason, we selected Any6D as a baseline for model-free pose estimation. Any6D generates a mesh using InstantMesh [16]. After an initial coarse alignment, it iteratively refines the pose and scale of the generated object to align it with the anchor image. At the end of the process, the mesh is scaled and the object pose is estimated. In the original model-free pose estimation setting, this mesh is then used to estimate the pose in the query image. Since our setup uses only a single image, we directly evaluate the mesh produced from the anchor view. Finally, we also experiment with replacing InstantMesh with TRELLIS [10].

Figure S6 Training dataset sample environment. Example indoor rendering setup used for data generation. A primary object is placed on a table-like support surface and surrounded by 3–10 distractor objects. Materials, textures, and lighting are randomized to increase visual diversity. Scenes are rendered in BlenderProc with stereo camera views sampled around the main object.

Scene completion. The objective of scene completion is to reconstruct complete object meshes or occupancy grids from a single-view RGB-D input [74]. When applied to open-set scenarios, these methods become closely related to the problem of simultaneous object reconstruction and pose estimation. Among them, we selected SceneComplete [14] as a baseline method. SceneComplete proposes a modular architecture. The method first inpaints occluded objects by conditioning on a prompt produced by a vision-language model (VLM) and an estimate of the occluded region. InstantMesh [16] is then used to reconstruct the object from the inpainted image, while the scale is estimated using DINO features extracted from the rendered mesh. Finally, FoundationPose [15] aligns the reconstructed mesh with the RGB-D observation. In the original work, the authors fine-tuned the inpainting module using LoRA to improve completion in the image plane. However, since the corresponding weights are not publicly available, we used the original pretrained weights for this module. To facilitate the inpainting process, we provided the ground-truth occlusion mask as input.

Single Image 3D reconstruction. The approach closest to RecGen is SAM3D [7], which simultaneously reconstructs objects and estimates their poses. SAM3D is proposed as a foundation model for 3D reconstruction, as it can generate aligned object meshes in an end-to-end manner. The method takes as input an RGB image, a segmentation mask of the target object, and optionally a point map of the scene. If the point map is not provided, it is estimated from the RGB image. The model is a two-stage diffusion architecture. In the first stage, the model predicts the object pose, scale, and structured latents [10]. In the second stage, it generates the object mesh and Gaussian representations conditioned on the structured latents. In our experiments, we provide the metric depth image to SAM3D to ensure a fair comparison.

F Detailed Datasets Description

F.1 Training Datasets

Each object gets a mini indoor scene constructed from our asset pool, which comprises 198K high-quality 3D assets collected from 6 public object and part datasets: Objaverse-XL, ABO, HSSD, PhysXNet, PartNext, and PartNet-Mobility. Object-centric datasets (Objaverse-XL, ABO, HSSD) are used to create compositional tabletop scenes where 3-10 randomly selected distractor objects from the same source dataset are placed around the main object to induce natural occlusions and complex depth relationships. For part-centric datasets (PhysXNet, PartNext, PartNet-Mobility), scenes contain a single object due to significant self-occlusion among articulated or fine-grained parts.

Scenes are rendered using BlenderProc [80], with randomized indoor layouts, textures, materials, and lighting configurations to enhance visual diversity and realism. A sample indoor setup used for rendering is shown in Fig. S6, illustrating the table-like support surface, surrounding distractors, and lighting arrangement. For each scene, twenty stereo camera views are sampled from varying azimuth, elevation, and distance around the primary object, producing a diverse set of viewpoints. In total, the dataset contains 198K scenes and 3.2M synthetically generated image pairs with and without occlusions.

For every rendered view, we provide RGB images, depth maps, stereo depth, semantic and instance segmentation masks, amodal masks, ground-truth 6D object poses, and full camera metadata. For appearance

generation training, subsets from PartNet-Mobility and PhysXNet are excluded due to lower texture quality. Example samples from the different datasets are shown in Fig. S7.

F.2 Object-based Evaluation Datasets

The experiments were conducted on four object-centric datasets: Linemod Occluded (LM-O) [71], NVIDIA Household Objects for Pose Estimation (HOPE) [73], ReOcS [74], and HomebrewedDB (HB) [72]. We selected these datasets to capture a variety of object types, cameras, and occlusion levels. For each dataset, we sampled a random subset of frames. We use standard 3×3 mask erosion to avoid misalignment between the depth map and the masks at the borders.

LM-O. This dataset contains 8 objects with significant occlusions, providing a standard benchmark for pose estimation under partial visibility. The RGB-D images were captured using a structured-light sensor from the Kinect v1 / PrimeSense family. From this dataset, we randomly sampled 142 frames.

HOPE. It includes 28 toy grocery objects captured in 50 scenes across 10 household and office environments, with up to five lighting variations and varying levels of occlusion. The RGB-D data were acquired using an Intel RealSense D415 camera, a stereo-based depth sensor delivering synchronized high-resolution color and depth streams. We selected this dataset for its lighting diversity and the presence of objects that are symmetric in shape but asymmetric in texture. From this dataset, we sampled 506 frames.

HB. It comprises 33 diverse objects (17 toy, 8 household, and 8 industry-relevant) recorded in 13 scenes with varying levels of complexity. We sampled 538 frames from the Kinect subset to include time-of-flight (ToF) camera data.

ReOcS. It provides 3D shape and pose annotations for 22 unseen objects, along with high-quality depth maps generated via learning-based stereo matching. The dataset is divided into three splits according to occlusion levels. We sampled 314 frames from the normal split, which contains balanced occlusions.

F.3 Part-based Evaluation Dataset

In the real-to-sim domain, the ability to decompose objects into components is fundamental for creating realistic simulations that are reliable and useful for robot learning [50, 81, 82]. Therefore, we believe that benchmarking the accuracy of methods that jointly perform reconstruction and pose estimation for object parts is crucial to understanding how reliable these approaches are for real-to-sim applications. A desirable dataset for part-based pose estimation should include RGB-D images captured from multiple viewpoints, camera parameters, part meshes, part poses, and realistic object arrangements. To the best of our knowledge, none of the existing datasets that provide ground-truth meshes possess all these characteristics. Since several works have demonstrated reliable sim-to-real transfer from scenes rendered with IsaacSim [15, 83–86], we decided to address this dataset shortage by leveraging this simulator. We started from the ArtVIP [75] articulation dataset, which contains six predefined scenes. We extended these scenes with additional articulated objects while preserving their realistic structure. From these scenes, we rendered views around the objects and organized the resulting data in the BoP format.
The following paragraphs provide further details on the motivations behind the dataset and its construction.

Object part estimation in real-to-sim. Estimating the shape and pose of object parts is fundamental for real-to-sim pipelines and robot learning. In particular, accurate part estimation is a key step in constructing models of articulated objects. [81, 82, 87–90]. When both the parts and their motions are correctly estimated, the resulting articulated models can be used to learn reliable manipulation policies. [81, 82] Recent advances in automatic real2sim and real2sim2real pipelines further highlight this need. [49, 50, 91]. For instance, RialTo [91] introduces a graphical interface for manual object part annotation. In DreMa [49] the robot parts are reconstructed starting from segmentation masks. Similarly, Real2Render2Real [50] uses segmentation to extract object parts, but additionally leverages videos of object motion to estimate their dynamics.

Dataset Generation. ArtVIP contains six scenes: children_room, dining_room, kitchen, kitchen_with_parlor, large_living_room, and small_living_room. For each scene, we created an additional replica containing more articulated objects defined in the ArtVIP dataset, resulting in a total of 12 scenes. For each object, we rendered 40 RGB-D images and visible segmentation masks using cameras uniformly sampled on a hemisphere around the object. We used a radius of 1.7 meters for all objects, except for those in the kitchen scene, where we used a radius of 1.2 meters because the objects are closer to each other. For objects that would otherwise fall outside the image frame, we increased the camera radius accordingly. From the 40 views, we discarded small masks or objects that were completely occluded. For each object, we extracted its constituent parts. We define parts as components of an object mesh whose motion depends on the object's structure. Parts can either be fixed elements (e.g., the legs of a chair) or movable components that change the object's state (e.g., articulated elements such as drawers). In addition to the rendered part masks, we also generated the full object mask. Finally, we extracted each part mesh in the world coordinate frame of the simulation, translated it to the origin, and computed its pose in each camera frame, organizing the resulting data in the BoP format.

Evaluation. From the generated dataset, we randomly sampled up to 4 views per part to construct the test set. At this stage, we initially selected 284 objects. We then manually filtered the objects to avoid oversampling parts that are widely represented in the dataset. After this process, the number of test objects was reduced to 262, corresponding to a total of 924 test frames. From this sample, we further randomly selected 500 frames for which a second view was available, enabling multi-view evaluation. The evaluation procedure follows the same protocol used for the other datasets.

G Additional Qualitative Comparison

G.1 Object-based Reconstruction

Figure S8 presents additional qualitative comparisons between RecGen and all baselines on scenes from HB, LMO, and ReOcS datasets, showing reconstructions from three viewpoints.

G.2 Part-based Reconstruction

Figure S9 presents a qualitative comparison between RecGen and SAM3D on part reconstruction from the ArtVIP dataset. Across diverse object parts categories (such as drawers, doors, lids, and appliance parts) RecGen generates reconstructions that more faithfully capture the part geometry, particularly for thin structures and parts under partial self-occlusion.

Figure S7 RecGen training dataset samples. Additional examples of 3D assets from our training dataset, including compositional scenes with objects from Objaverse-XL, ABO, HSSD, and parts in object scenes the part-based datasets from PhysXNet, PartNext, and PartNet-Mobility.

Figure S8 Qualitative comparison with baselines. We compare RecGen with SceneComplete, Any6D, and SAM3D on scenes from HB, LMO, and ReOcS datasets. Each row group shows the input image, reconstructions from each method, and the ground truth (GT) from three viewpoints.

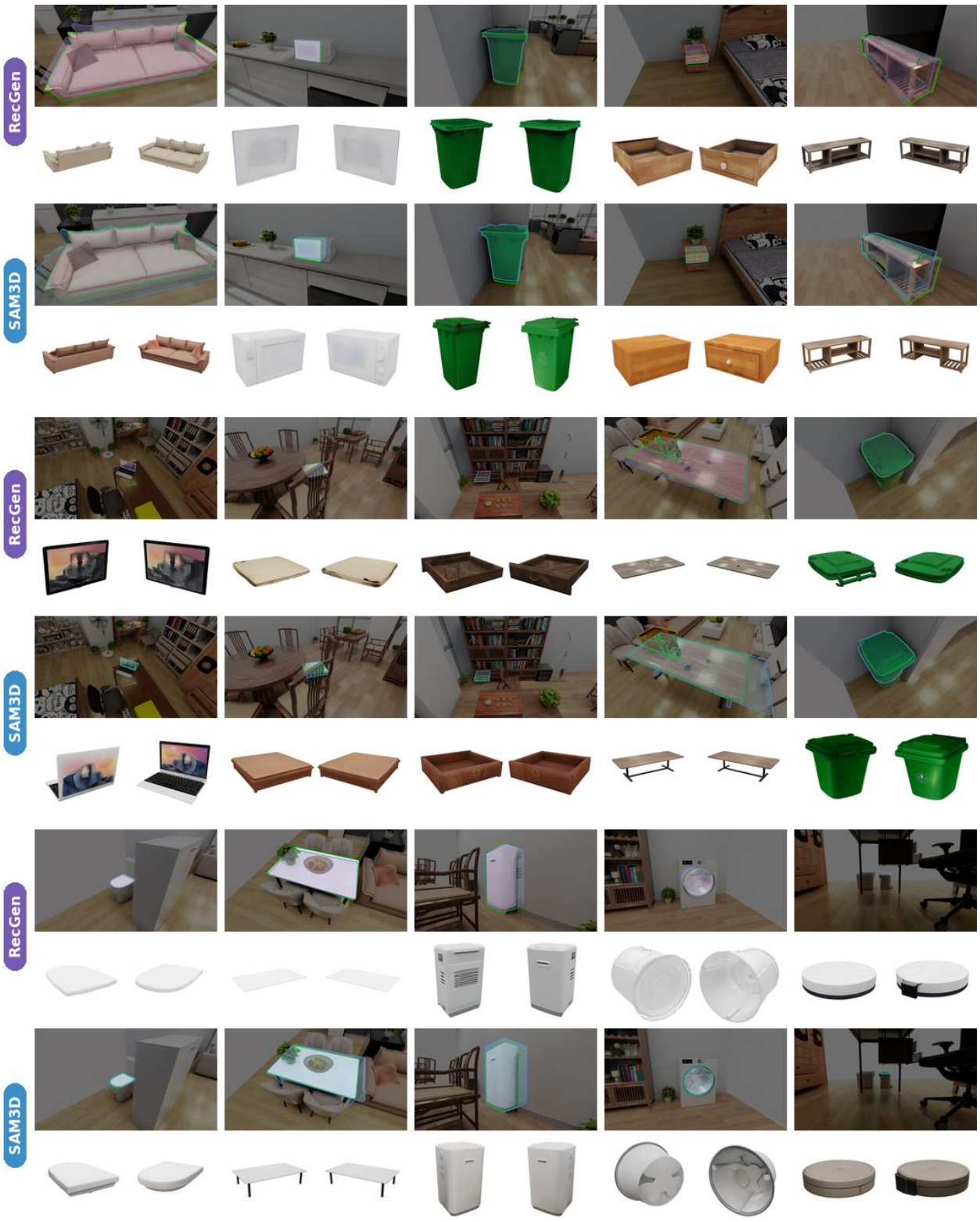


Figure S9 Part-based reconstruction: RecGen vs. SAM3D on ArtVIP. Each column shows one articulated part. For each method: scene overlay with GT mask contour (green) and predicted mesh (purple/blue), plus two novel-view GS renders. RecGen produces more accurate part geometry across diverse categories.