HIGH-FIDELITY SINGLE IMAGE TO 3D SCENE GENERATION WITH FOUNDATION MODEL ORCHESTRATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Single-image 3D scene generation supports immersive content creation and embodied-AI simulation. High-fidelity generation must provide complete object structure under occlusion, detailed geometry and appearance, and metrically consistent layout. Training one end-to-end model for all these goals would require samples that jointly encompass natural occlusion, high-quality geometry and texture, and metric-scale layout, making such data difficult to collect at scale. Existing methods therefore often trade off these goals: occlusion-aware generators may lack local detail, detail-rich object generators are sensitive to incomplete observations, and their visually plausible layouts can still have poor metric-scale accuracy. Our key observation is that structured latents naturally separate these roles: occlusion awareness is primarily reflected in the global structure of active voxel coordinates, while feature fields encode local detail and appearance. We exploit this data–capability decomposition through foundation-model orchestration: a heterogeneous latent infusion scheme conditions a detail-rich generator on active voxel coordinates supplied by an occlusion-aware model, to generate high-quality 3D assets under occlusion. To achieve accurate metric-scale layouts, we develop constraints on metric-related pose components based on visible-region boundaries and visible depth, which can be solved by an efficient iterative linear solver. Despite its simple design and straightforward implementation, extensive experiments on the large-scale 3D-FRONT and M3DLayout benchmarks show that our method significantly improves object fidelity and metric-scale layout accuracy, while retaining competitive computational efficiency. These results validate the effectiveness of foundation-model orchestration as a practical approach towards high-fidelity 3D scene generation.

1 INTRODUCTION

Single-image 3D scene generation supports immersive content creation and embodied-AI simulation (Wang et al., 2024; Dai et al., 2024; Ma et al., 2026; Lee et al., 2026). Advances in image-to-3D modeling have enabled the generation of individual assets with rich geometry and appearance (Liu et al., 2023a; Long et al., 2024; Hong et al., 2023; Tang et al., 2025; Li et al., 2025; Xiang et al., 2025a). Extending these capabilities to scenes requires three properties: complete object structure under occlusion, detailed geometry and appearance, and metrically consistent layout. Structural completeness and local detail determine how objects appear beyond the input viewpoint, while their physical scales and positions determine how they fit and interact within the scene.

Training an end-to-end model to satisfy all these demands would require prohibitively costly data collection. Specifically, a sample must jointly encompass natural occlusion, high-quality geometry and texture, including material information for physically based rendering (PBR), and metric-scale pose, placement, and camera information. Samples with all these annotations are difficult to obtain at the scale needed for generalization. SAM3D (SAM 3D Team et al., 2025) illustrates this cost: its data engine combines synthetic and render-paste data with model-in-the-loop annotation and substantial artist intervention. Its model refines geometry and synthesizes textures, yet our comparisons reveal room for improvement in local surface detail under occlusion (Fig. 2). Complementary generators offer finer geometry and PBR materials (Xiang et al., 2025a), motivating a framework that combines these capabilities without collecting a new jointly annotated dataset.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Constrained by data, existing approaches tend to trade off the goals, prioritizing occlusion-aware structure, detailed geometry and appearance, or scene layout (SAM 3D Team et al., 2025; Xiang et al., 2025a; Huang et al., 2025b; Yao et al., 2025). Our key insight is that the requirements of occlusion-aware structural completion and local detail synthesis align with the structure–feature decomposition of structured 3D latents (Xiang et al., 2025b; 2026). Occlusion awareness is primarily reflected in the global structure of active voxel coordinates, while feature fields encode local detail and appearance. This decomposition allows the two capabilities to be drawn from complementary pretrained models, without requiring them to be learned together from a jointly annotated dataset.

We introduce FOMO, a training-free foundation-model orchestration framework that realizes this idea through a heterogeneous latent infusion scheme. It supplies active voxel coordinates produced by occlusion-aware models to detail-rich generators, to produce a detail-rich asset under occlusion while avoiding the need for cross-model feature alignment. To achieve metric-scale consistency of scene layouts, we observe that, of the seven degrees of freedom in each object's pose, only scale and translation depend on metric scale. Thus, during the calibration process, we retain object rotation from the generator prior and solve only for scale and translation. Using constraints on visible-region boundaries and depth, we formulate these variables as a small linear system, which can be solved efficiently, yielding an efficient metric-scale calibration stage.

Extensive experiments on the large-scale 3D-FRONT (Fu et al., 2021) and M3DLayout (Zhang et al., 2025) benchmarks show that FOMO outperforms the baselines in object fidelity and layout accuracy. Moreover, it demonstrates robustness to occlusion, different scaling of scenes, and depth noise, with competitive computational efficiency. The results validate the value of foundation-model orchestration as a viable approach for achieving high-quality 3D scene reconstruction.

Our contributions can be summarized as follows:

- A heterogeneous latent infusion scheme that exploits the structure–feature decomposition, designed for high-fidelity asset generation under occlusion, requiring no extra training.
- An efficient iterative linear calibration algorithm that recovers metric scale and translation.
- Experiments show that FOMO improves object fidelity and layout accuracy, while remaining robust to occlusion, scene scaling, and depth noise with computational efficiency.

2 Related Works

Scene Reconstruction and Compositional Generation. Total3D and InstPIFu infer scene geometry and layout from image features (Nie et al., 2020; Liu et al., 2022a). Compositional pipelines complete and assemble individual assets (Zhou et al., 2024; Ardelean et al., 2024), while MIDI, CAST, and CUPID couple generation with pose or component alignment (Huang et al., 2025b; Yao et al., 2025; Huang et al., 2025a). 3D-Fixer completes partial geometry in place, and SceneReGen generates objects in observed orientations before estimating translation and scale (Yin et al., 2026; Tian et al., 2026). HIVE-3D uses learned voxel super-resolution to refine scene components hierarchically (Zang et al., 2026). Feed-forward, multi-view, and physics-aware approaches are discussed in Appendix Sec. A.10.

Structured Representations and Instance Refinement. Multi-view methods synthesize consistent views for reconstruction (Liu et al., 2023b; Long et al., 2024), whereas 3D-native models learn asset representations directly (Zhang et al., 2024; Li et al., 2025). TRELLIS associates local features with sparse voxel support, and TRELLIS.2 models detailed geometry and materials (Xiang et al., 2025b;a). DVD learns discrete voxel occupancy while retaining TRELLIS's feature-generation stage (Xiang et al., 2026). These models rely on data that emphasizes detailed geometry, texture, appearance, or material consistency, which need not contain the natural occlusion and metric-scale pose annotations required by scene generation.

Occlusion-Aware Reconstruction and 3D Data Engines. Occlusion-aware reconstruction uses different conditioning strategies: Amodal3R uses mask-weighted attention (Wu et al., 2025), SAM3D learns from visually grounded object data (SAM 3D Team et al., 2025), and Axolotl3D combines posed images, visibility masks, and partial point clouds through multimodal training (Hu & Shugrina, 2026). RecGen (Zadaianchuk et al., 2026) also targets robust single-image reconstruction from partial observations, but its reconstruction pipeline remains specialized to its own

learned representation and supervision regime, whereas FOMO treats occlusion-aware structure as a transferable interface for complementary pretrained generators. Our method uses such pretrained capabilities as modular sources rather than requiring their supervision regimes to be merged into a new end-to-end scene dataset.

Metric Layout and Pose Calibration. Gen6D, MegaPose, SAM-6D, and FoundationPose recover unseen-object poses from reference observations, learned matching, or rendering (Liu et al., 2022b; Labbé et al., 2023; Lin et al., 2024; Wen et al., 2024). ICP (Besl & McKay, 1992) aligns point sets; the Open3D variant used in our evaluation optimizes rotation, translation, and uniform scale in $\mathrm{Sim}(3)$ (Zhou et al., 2018). For generated geometry, Zhang et al. (2026) combine Sim(3) registration with scale-locked rigid refinement against partial observations. SnapPhysics uses registration and object-wise similarity optimization to align SAM3D assets with estimated metric depth (Kang et al., 2026). These methods update rotation during alignment.

Distinguished from existing work, FOMO focuses on latent compatibility between complementary pretrained priors for high-fidelity scene generation. It develops a training-free framework that exploits this compatibility to combine capabilities learned from different data regimes, without requiring joint end-to-end training.

3 METHODOLOGY

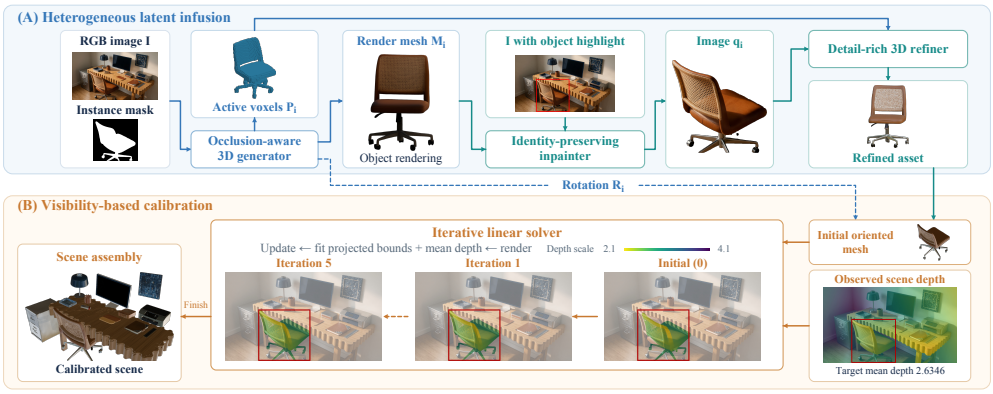

Figure 1: Illustration of the FOMO framework. Given an image and instance masks, the heterogeneous latent infusion scheme first produces high-fidelity assets under occlusion, and then visibility-based calibration refines the object pose in metric scale.

As discussed in Sec. 1, to avoid the prohibitively costly joint data collection, FOMO develops a training-free framework that orchestrates complementary pretrained priors for scene generation. Given an image $I \in \mathbb{R}^{H\times W\times 3}$ and instance masks $\{m_i\}_{i=1}^{N}$, an occlusion-aware generator first produces a coarse mesh M_i^{src} and active voxel coordinates $\mathcal{P}_i$ for each object. We render the coarse mesh and mark the target bounding box b_i in the original scene image, then provide these two complementary views to an identity-preserving inpainter to obtain a complete, detail-rich object image q_i. A detail-rich generator uses $(\mathcal{P}_i, q_i)$ to synthesize a refined mesh (Sec. 3.1). Finally, given camera intrinsics K and metric depth D, we estimate object-wise scale and translation from projected boundaries and visible depth and assemble the calibrated objects into a scene (Sec. 3.2). Fig. 1 illustrates the pipeline.

3.1 INSTANCE GENERATION

Active voxel coordinates as a shared interface. Structured latents (SLAT) (Xiang et al., 2025b;a) represent an object as $\mathcal{Z} = \{(\mathbf{p}_j, \boldsymbol{\ell}_j)\}_{j=1}^{V}$, pairing coordinates of active voxels $\mathcal{P} = \{\mathbf{p}_j\}$ with local geometry and appearance features $\mathcal{L} = \{\boldsymbol{\ell}_j\}$. For a SLAT-based generator r conditioned on an

object image q, the generation process factorizes as

$$p_r(\mathcal{P}, \mathcal{L} \mid q) = \underbrace{p_r(\mathcal{P} \mid q)}_{\text{structure}} \underbrace{p_r(\mathcal{L} \mid \mathcal{P}, q)}_{\text{feature synthesis}}. \tag{1}$$

The structure–feature decomposition reveals a transferable interface between complementary pre-trained priors: occlusion-aware completion primarily completes the global structure represented by active voxel coordinates, while detail-rich generation produces higher-quality feature fields encoding finer geometry and appearance. This correspondence lets us conduct high-fidelity asset generation under occlusion without joint end-to-end training.

Heterogeneous latent infusion. Given I_i, the image crop of object i, an occlusion-aware mesh generator g first produces a coarse mesh M_i^{src} and active voxel coordinates $\mathcal{P}_i$:

$$(M_i^{\text{src}}, \mathcal{P}_i) \sim G_g(I_i). \tag{2}$$

We render the complete coarse mesh from a view in which the object is fully visible. This rendering exposes completed shape and object identity without scene occluders, but it does not provide detailed texture or local geometry. As such, we mark object i in I and provide both images to an identity-preserving inpainter:

$$q_i = G_{\text{VLM}}(\text{Render}(M_i^{\text{src}}),\ \text{Mark}(I, b_i))\,, \tag{3}$$

where b_i is the bounding box of i in the input image and q_i is the resulting complete, detail-rich object image. Note that b_i can be obtained from the input instance mask m_i. The mesh rendering supplies complete object structure, while the target-marked scene image provides instance-specific appearance and context. The VLM combines them to preserve complete object shape while enhancing the visible evidence to be detail-rich. The detail-rich mesh generator takes as inputs the active coordinates and q_i, then synthesizes local features and decodes them into a refined asset:

$$\mathcal{L}_i \sim p_r(\mathcal{L} \mid \mathcal{P}_i, q_i), \qquad M_i^r = D_r(\mathcal{P}_i, \mathcal{L}_i), \tag{4}$$

where D_r is the refiner's asset decoder. Thus, the upstream generator provides occlusion-aware structure and the VLM supplies an object-specific visual condition for the detail-rich generator. Our experiments evaluate this transfer across three source generators and two refiners, using explicit coordinate conventions and feature sampling at supplied active indices (Appendix Sec. A.2).

3.2 Metric-Scale Calibration from Visible Geometry

Visual plausibility alone does not guarantee metric-scale layout consistency due to scale-depth ambiguity. For each object, denote its pose as $(s_i, R_i, \mathbf{t}_i)$. We notice that rotation R_i is independent of metric scale, whereas object scaling s_i and translation $\mathbf{t}_i$ are not. We therefore retain the rotation produced by the upstream generator and optimize only the metric-dependent scale and translation. Given camera intrinsics K, an object mask, and metric depth D, we derive the constraints they impose on object poses. The two constraints have complementary roles: projected boundaries constrain the image-space footprint, while visible depth anchors the object in metric camera space.

For one object, let $\boldsymbol{\mu}$ be the arithmetic mean of all mesh vertices. Transforming it with a uniform scale s and translation $\mathbf{t} = (t_x, t_y, t_z)^\top$ will update its vertex at $\mathbf{x}$ to

$$\mathbf{x}' = \boldsymbol{\mu} + s(\mathbf{x} - \boldsymbol{\mu}) + \mathbf{t}. \tag{5}$$

Projected boundary matching. For one object, let (u_L, u_R, v_T, v_B) be its 2D bounding box. From the current isolated rendering, we obtain surface points corresponding to the four projected extrema and assume that they remain the corresponding extrema within each iteration of translation and scaling. A point $\mathbf{x}_b = (X_b, Y_b, Z_b)^\top$ associated with a horizontal boundary u_b satisfies

$$f_x \frac{\mu_x + t_x + s(X_b - \mu_x)}{\mu_z + t_z + s(Z_b - \mu_z)} + c_x = u_b. \tag{6}$$

Here f_x, c_x are the horizontal focal length and principal point from K. This gives a linear equation in (t_x, t_y, t_z, s). The two horizontal boundaries and two vertical boundaries, using f_y, c_y, provide four equations that constrain the projected extent. Boundary matching alone, however, leaves scale–depth ambiguity unresolved.

Table 1: Comparing general-purpose metric-scale calibrators, with CD-S↓ / FScore-S↑ / IoU-B↑. ICP denotes Open3D $\mathrm{Sim}(3)$ registration with scale estimation enabled.

Base Model	None	ICP	SAM3D-post	Ours
		3D-FRONT		
CUPID	3.535/0.029/0.010	3.888/0.106/0.025	2.616/0.200/0.103	**0.169/0.780/0.435**
MIDI	6.123/0.000/0.000	0.243/0.656/0.322	2.984/0.260/0.146	**0.060/0.936/0.686**
SceneGen	6.230/0.000/0.000	0.366/0.459/0.124	3.338/0.237/0.127	**0.113/0.841/0.593**
SAM3D	0.106/0.857/0.454	0.095/0.878/0.449	0.116/0.835/0.440	**0.062/0.934/0.649**
RecGen	0.108/0.858/0.477	0.107/0.853/0.449	0.126/0.817/0.427	**0.100/0.860/0.555**
Gen3DSR	0.160/0.758/0.420	0.142/0.773/0.384	0.173/0.738/0.386	**0.114/0.852/0.552**
DepR	6.066/0.000/0.000	0.354/0.415/0.066	3.578/0.192/0.108	**0.119/0.827/0.498**
PartCrafter	6.178/0.000/0.000	0.265/0.620/0.221	3.751/0.170/0.092	**0.209/0.688/0.291**
		M3DLayout		
CUPID	10.436/0.004/0.000	9.379/0.029/0.004	7.800/0.127/0.056	**0.581/0.445/0.230**
MIDI	10.906/0.000/0.000	2.293/0.122/0.016	4.693/0.266/0.159	**0.181/0.784/0.498**
SceneGen	10.964/0.000/0.000	2.261/0.206/0.029	5.963/0.193/0.103	**0.288/0.641/0.393**
SAM3D	0.198/0.725/0.403	0.167/0.768/0.415	0.192/0.728/0.405	**0.136/0.849/0.549**
RecGen	0.226/0.697/0.367	0.230/0.666/0.333	0.215/0.693/0.366	**0.214/0.708/0.463**
Gen3DSR	0.303/0.566/0.334	0.239/0.609/0.334	0.257/0.643/0.370	**0.211/0.721/0.471**
DepR	10.936/0.000/0.000	2.164/0.261/0.042	6.614/0.135/0.066	**0.261/0.625/0.361**
PartCrafter	10.993/0.000/0.000	2.264/0.121/0.019	9.286/0.067/0.028	**1.448/0.206/0.074**

Visible-depth matching. To resolve scale–depth ambiguity, we use the observed depth D to constrain the mean camera depth of visible surfaces. Let Ω_i be the set of visible pixels (u, v) within the instance mask m_i with finite positive depth $D(u, v)$. Their mean depth is

$$\widetilde{d}_i = \frac{1}{|\Omega_i|} \sum_{(u,v)\in\Omega_i} D(u, v). \tag{7}$$

Without a clear correspondence between the observed depth and the generated surface, we cannot directly match pointwise depth. Therefore we instead constrain the mean depth of the generated surface to match the observed mean depth. Let $\mathcal{V}_i$ contain the visible surface samples corresponding to foreground pixels with finite positive depth in an isolated rendering of the generated mesh, with mean depth $\overline{Z}_{\mathcal{V}_i}$. Aligning the mean depth of the generated surface with the observed mean depth gives:

$$\mu_z + t_z + s(\overline{Z}_{\mathcal{V}_i} - \mu_z) = \widetilde{d}_i. \tag{8}$$

Joint solution and iterative refinement. Note that the four boundary constraints and one depth constraint together form a linear system in (t_x, t_y, t_z, s). Thus, they can be summarized as $A_i\boldsymbol{\theta}_i = \mathbf{b}_i$, with $A_i \in \mathbb{R}^{5\times4}$ and $\boldsymbol{\theta}_i = (t_x, t_y, t_z, s)^\top$. Thus each update solves only five linear constraints in four unknowns:

$$\begin{aligned} \boldsymbol{\theta}_i^\star &= \arg\min_{\boldsymbol{\theta}_i} \|\Lambda(A_i\boldsymbol{\theta}_i - \mathbf{b}_i)\|_2^2, \\ \Lambda &= \mathrm{diag}(2/W, 2/W, 2/H, 2/H, 1). \end{aligned} \tag{9}$$

One could refer to Appendix Secs. A.1 and A.8 for the full derivation.

4 Experiments

4.1 Experimental Setup

We evaluate how foundation-model orchestration combines complete object structure, local geometry and appearance, and metric-scale scene layout. The quantitative evaluation uses synthetic rooms from 3D-FRONT (Fu et al., 2021) and M3DLayout (Zhang et al., 2025); additional AI-generated scenes provide qualitative examples. For quantitative comparisons, we follow the MIDI (Huang et al., 2025b) evaluation protocol, render each scene, and supply the rendered RGB images, GT

Table 2: Comparing FOMO variants and baselines on object fidelity and metric-scale calibration accuracy. Each entry stands for the performance on 3D-FRONT / M3DLayout respectively. Details in Sec. 4.1.

Model	CD-O↓	FScore-O↑	CD-S↓	FScore-S↑	IoU-B↑
3D-FRONT / M3DLayout					
CUPID	0.144/0.159	0.824/0.832	3.535/10.436	0.029/0.004	0.010/0.000
MIDI	0.112/0.120	0.894/0.901	6.123/10.906	0.000/0.000	0.000/0.000
SceneGen	0.135/0.143	0.885/0.849	6.230/10.964	0.000/0.000	0.000/0.000
SAM3D	0.105/0.101	0.903/0.865	0.106/0.198	0.857/0.725	0.454/0.403
RecGen	0.126/0.128	0.823/0.847	0.108/0.226	0.858/0.697	0.477/0.367
Gen3DSR	0.173/0.181	0.649/0.627	0.160/0.303	0.758/0.566	0.420/0.334
DepR	0.195/0.203	0.595/0.574	6.066/10.936	0.000/0.000	0.000/0.000
PartCrafter	0.149/0.156	0.805/0.784	6.178/10.993	0.000/0.000	0.000/0.000
CUPID+FOMO.T	0.138/0.140	0.893/0.885	0.169/0.581	0.784/0.441	0.429/0.227
CUPID+FOMO.T.2	0.132/0.122	0.891/0.882	0.166/0.580	0.780/0.446	0.434/0.224
RecGen+FOMO.T	0.118/0.123	0.903/0.899	0.099/0.221	0.861/0.702	0.553/0.463
RecGen+FOMO.T.2	0.109/0.113	0.910/**0.916**	0.102/0.211	0.870/0.705	0.556/0.466
SAM3D+FOMO.T	0.102/0.099	0.925/0.904	0.066/0.130	**0.933**/0.847	**0.654/0.552**
SAM3D+FOMO.T.2	**0.082/0.098**	**0.938**/0.912	**0.058/0.129**	0.930/**0.853**	0.649/**0.552**

camera intrinsics, GT masks, and GT depth or pointmaps as required by each model. Model configurations are detailed in Appendix Sec. A.2.

We report unsquared bidirectional Chamfer distance and F-Score at the object level (CD-O and FScore-O) and scene level (CD-S and FScore-S), together with 3D bounding-box overlap (IoU-B). CD-O and FScore-O assess center-aligned generated meshes before metric-scale calibration; CD-S, FScore-S, and IoU-B assess the calibrated scene layout. For the calibration analysis under occlusion, CD-O-raw evaluates objects in the scene frame after each calibration setting, without center alignment. Distances are in metres, and F-Score uses a distance threshold of $0.1\,\mathrm{m}$. Lower CD and higher F-Score/IoU are better. Appendix Secs. A.2 and A.3 describe more details on implementation and evaluation.

4.2 Effectiveness of Proposed Mechanisms

Metric-scale calibration. As Tab. 1 shows, the proposed metric-scale calibrator obtains the best reported performance on all the scene-level metrics, in every generator-calibrator-dataset tuple. For SAM3D on 3D-FRONT, Ours reduces CD-S from ICP's 0.095 to 0.062 and increases IoU-B from 0.449 to 0.649. These gains over the full $\mathrm{Sim}(3)$ baseline support the effectiveness of our visible-region constraints while retaining the generator's rotation and solving only for scale and translation. The gains extend from generators with substantial initial errors, such as CUPID and MIDI, to those with better metric-scale accuracy, like SAM3D and RecGen. This validates the effectiveness and compatibility of our visible region-based metric-scale calibrator as a general-purpose calibrator. This comparison focuses on scene-level alignment; each generator shares its pre-calibration CD-O and FScore-O with the corresponding base-generator row in Tab. 2.

We also inspect the rationale behind our calibrator. As Fig. 13 illustrates, our calibrator exhibits improved object placement performance compared to ICP and SAM3D-post. We hypothesize that, by exploiting the global visibility information in pose calibration, our calibrator avoids the ill-posed problem of estimating pointwise correspondences between an incomplete pointmap and a generated surface, thus exhibiting greater robustness to object occlusion. The occlusion sensitivity analysis on 3D-FRONT in Fig. 3 and Tab. 5 supports this hypothesis: our calibrator improves placement over the compared alternatives on most occlusion intervals for each mesh generator. Also, as illustrated in Fig. 7, partial visibility can make metric-scale calibration under-constrained, with multiple scale–translation solutions consistent with the visible observation. Our solver therefore favors solutions more directly supported by visible evidence: it constrains the object's projected extent to match the visible-region bounding box and anchors its mean visible depth to the observed depth. In the illustrated example, these constraints select Solution 1, whose projected extent matches the observed bounds, over the enlarged and translated Solution 2.

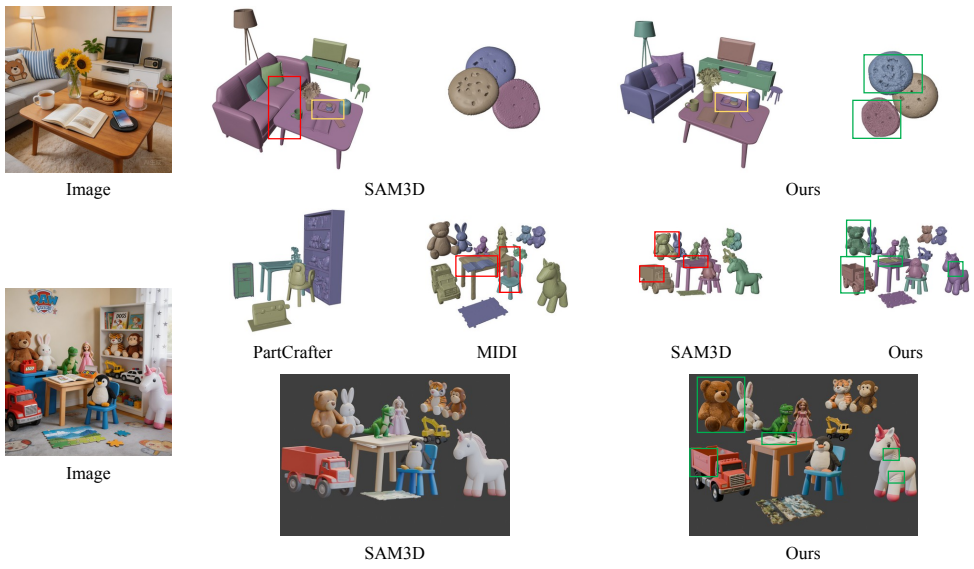

Figure 2: Qualitative comparison of fidelity to the input, local detail, and scene layout. In the upper example, FOMO more faithfully reproduces the cookies' porous surface structure visible in the input image (zoom in for better visualization). In the lower example, our results preserve the folds on the unicorn's legs and neck and the bear's plush surface texture, while correctly placing the book on the tabletop. Further detail comparisons appear in Fig. 8 and Fig. 9. Fig. 5 and Fig. 6 provide additional examples, while Fig. 14–32 extend the comparison across a broader collection of scenes.

Heterogeneous latent infusion. We equip CUPID, RecGen, and SAM3D with our visible region-based calibrator, then refine their meshes with TRELLIS or TRELLIS.2 to obtain FOMO.T and FOMO.T.2. To assess the added refinement, compare the lower block of Tab. 2 with the corresponding Ours entries in Tab. 1. The calibration-only references share the object metrics of their base generators in the upper block of Tab. 2, since CD-O and FScore-O are evaluated before calibration. Both refiners improve these object metrics for all three generators on both datasets, while the reported scene metrics differ from the calibration-only references by at most 0.010. The paired visual comparisons below show the local detail responsible for the object-fidelity improvement.

Comparing the lower and upper blocks of Tab. 2 evaluates the complete framework against the base generators. Both FOMO variants improve all five reported metrics over their corresponding base generators on both datasets. For example, SAM3D+FOMO.T.2 reduces CD-O from 0.105 to 0.082, raises FScore-O from 0.903 to 0.938, and reduces CD-S from 0.106 to 0.058 on 3D-FRONT. These comparisons support complementary roles for refinement and calibration in improving object fidelity and metric-scale scene alignment within a training-free framework.

Fig. 2 highlights the joint improvement in local surface detail and scene placement through the cookie, unicorn, and tabletop-book examples. Fig. 8 and 9 provide direct visual comparisons of the local detail produced by FOMO and SAM3D. Even with substantial portions of the toy car and audio mixer missing from the input view, FOMO generates a detailed cargo bed and distinct knobs and sliders, whereas SAM3D produces overly smooth surfaces that blur these geometric features. The solid-shaded comparisons in Fig. 9 make these differences visible independently of texture, illustrating how heterogeneous latent infusion enriches local geometry while retaining the completed object structure. Fig. 6 further shows fine surface details and material-dependent reflections, demonstrating benefits for both geometry and appearance. The broader comparisons in Fig. 14–32 extend this visual assessment across source generators, refiners, and scene types, including bedrooms, kitchens, recording studios, and workshops. Together, these examples illustrate the effectiveness of our latent infusion scheme. Tab. 6 provides more evidence on occlusion robustness of our latent infusion scheme. Both FOMO variants show lower CD-O than their respective base generators in every occlusion interval, with FOMO.T.2 giving the lowest values throughout. The human preference study

in Tab. 9 further supports improvements over SAM3D in both faithfulness to the input and detail fineness.

Ablation on detail-rich input-image inpainting. The 2×2 ablation in Tab. 8 evaluates the individual and combined contributions of inpainting and heterogeneous latent infusion to improved object-level geometric fidelity on 3D-FRONT. We compare the inpainter-conditioned results in Fig. 14–32 with direct mesh-render and segmentation-crop conditioning in Fig. 33–51. The two collections use the same 19 input scenes in the same order: compare Fig. 14 with Fig. 33, Fig. 15 with Fig. 34, and so on, matching the RecGen or SAM3D row and the TRELLIS or TRELLIS.2 column. A complete coarse-mesh rendering supplies structure but limited appearance detail; a segmentation crop retains observed appearance but can be incomplete under occlusion. In the bedroom pair, the direct conditions produce darker or altered furniture surfaces, whereas the inpainter-conditioned outputs better retain the input's appearance. These paired qualitative comparisons support combining complete structure and observed appearance through the identity-preserving inpainter.

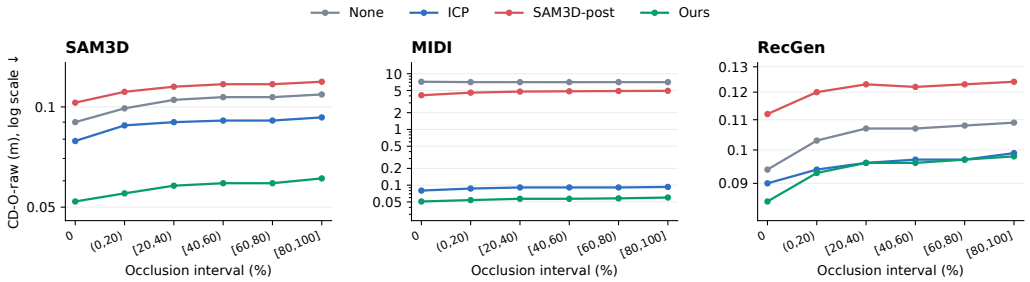

Figure 3: Comparing the occlusion robustness of general-purpose calibrators on 3D-FRONT. Tab. 5 reports the detailed numerical values and additional comparisons on other base generators.

4.3 Robustness to Occlusion, Depth Noise, and Scene Scaling

Occlusion. We examine how occlusion affects both metric-scale placement and object fidelity. For calibration, Fig. 3 and Tab. 5 show lower CD-O-raw for Ours than for None and SAM3D-post throughout the reported comparisons; it also matches or improves on the displayed ICP values except in the first two SceneGen intervals. These comparisons demonstrate the benefit of visible-region constraints across diverse source generators, while also identifying intervals where ICP remains competitive. For heterogeneous latent infusion, Tab. 6 shows that both FOMO variants perform favorably against their base generators across all occlusion intervals. Appendix Sec. A.3 details the design of these experiments.

We additionally examine calibration sensitivity to bounding-box truncation by measuring the proportional reductions in object width and height from isolated-rendering to scene-visible GT bounding boxes on 3D-FRONT. Tab. 7 compares Ours and ICP on the same SAM3D meshes across 16 joint horizontal–vertical truncation bins using CD-O-raw; Appendix Sec. A.4 details the protocol.

Table 3: Robustness to depth noise on 3D-FRONT with SAM3D. Noise is added before generation, and the full pipeline is rerun with fixed seeds. Entries give noisy scores and changes from each configuration's clean reference; Ours uses SAM3D+FOMO.T.2.

Method	CD-S↓	FScore-S (%)↑	IoU-B↑
None	0.1970 (+0.0910)	61.19 (-24.51)	0.247 (-0.207)
ICP	0.1593 (+0.0643)	68.63 (-19.17)	0.273 (-0.176)
SAM3D's post processing	0.1915 (+0.0755)	61.39 (-22.11)	0.251 (-0.189)
Ours	**0.062 (+0.004)**	**91.58 (-1.42)**	**0.623 (-0.026)**

Depth noise. We add Gaussian noise $\epsilon \sim \mathcal{N}(0, 0.1^2)$ to depth before generation and rerun the full pipeline with fixed seeds on 3D-FRONT (Tab. 3). We verify that SAM3D's generated mesh is unchanged: its mesh-generation branch does not depend on depth or pointmaps, so noise affects pose estimation and subsequent calibration. Relative to its clean reference, SAM3D+FOMO.T.2 has

a CD-S increase of 0.004, an FScore-S decrease of 1.42 percentage points, and an IoU-B decrease of 0.026. All three alternatives deteriorate more in each metric, supporting the robustness of our placement procedure under noisy depth.

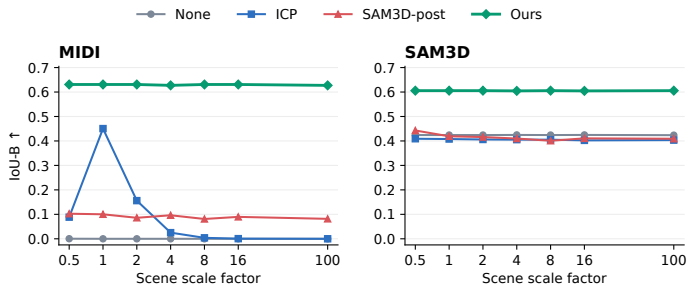

Figure 4: Comparing the calibrators' robustness against scene scaling, on 3D-FRONT, with MIDI and SAM3D, measured by IoU-B. Tab. 10 and Tab. 11 report all the numerical values in detail.

Scene scaling. We test the generality of calibrators on different scales of scenes. We test MIDI and SAM3D on a subset of 100 scenes randomly chosen from 3D-FRONT, at scales from $0.5\times$ to $100\times$, rerunning generation and calibration at each scale. As Fig. 4 shows, Ours achieves the highest IoU-B at every tested scale for both generators, maintaining approximately 0.627–0.631 for MIDI and 0.605–0.606 for SAM3D. This stability indicates that the visible-region and depth constraints preserve relative object placement as the metric scale changes. Ours also obtains the lowest CD-S throughout both comparisons; the complete CD-S, FScore-S, and IoU-B results at every scale are provided in Tab. 10 and Tab. 11. Appendix Sec. A.7 gives the full protocol and metric definitions.

Table 4: Reported runtime per object, in seconds. FOMO uses SAM3D with TRELLIS.2 refinement.

Method	CUPID	MIDI	SceneGen	SAM3D	RecGen	Gen3DSR	DepR	PartCrafter	FOMO
Time (s)	22	11	14	10	13	168	12	18	16

4.4 Efficiency

Tab. 4 reports an average of 16 seconds per object for the SAM3D+TRELLIS.2 configuration, compared with 10 seconds for SAM3D alone. The additional components comprise coarse-mesh rendering (less than 10 ms per object), detail-rich feature synthesis (about 2 s per object), and metric-scale calibration (about 100 ms per object). For identity-preserving inpainting, a VLM call takes approximately 33 s; processing the objects of a 10–20-object scene concurrently amortizes this cost to approximately 1.6–3.3 s per object. The reported figure therefore reflects amortized scene processing rather than the latency of a single isolated VLM request. Together with the small linear calibration solve, this orchestration retains a practical runtime while adding geometric detail and material appearance.

5 Conclusion

We presented FOMO, a training-free framework for high-fidelity single-image 3D scene generation through foundation-model orchestration. Heterogeneous latent infusion uses active voxel coordinates as a shared interface between occlusion-aware structure and detail-rich feature synthesis, with identity-preserving inpainting providing a complete object condition. Metric-scale calibration complements this refinement by solving for scale and translation from projected boundaries and visible depth. Across eight upstream generators on 3D-FRONT and M3DLayout, the reported calibration comparison improves scene geometry and layout; the full FOMO configurations also improve object fidelity for CUPID, RecGen, and SAM3D. Together, these results support active voxel coordinates and visible geometry as practical interfaces for combining complementary pretrained capabilities without joint end-to-end training. Future work can extend this approach to uncertain masks and depth, and the physical constraints required for simulation.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic scene images for evaluation, implementing methods, providing feedback on experimental design, translating text, and interpreting results. We have not used generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for mathematical proofs or assist in writing proofs, propose or refine hypotheses, clean or reformat datasets, or conduct qualitative or thematic data analysis.

We have reviewed all AI-assisted work, including testing the code, cross-checking the results, and reviewing the manuscript. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Andreea Ardelean, Mert Özer, and Bernhard Egger. Gen3DSR: Generalizable 3D scene reconstruction via divide and conquer from a single view. *arXiv preprint arXiv:2404.03421*, 2024. URL https://arxiv.org/abs/2404.03421.

P. J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. doi: 10.1109/34.121791. URL https://doi.org/10.1109/34.121791.

ByteDance Seed. Seedream 5.0 Pro. Official model page, July 2026. URL https://seed.bytedance.com/en/seedream5_0_pro. Released July 8, 2026; accessed September 25, 2026.

Tianyuan Dai, Josiah Wong, Yunfan Jiang, Chen Wang, Cem Gokmen, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Automated creation of digital cousins for robust policy learning, 2024. URL https://arxiv.org/abs/2410.07408.

Martin A. Fischler and Robert C. Bolles. Random sample consensus. *COMMUN ACM*, 1981.

Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10933–10942, 2021.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023.

Anita Hu and Maria Shugrina. Axolotl3D: a Unified Framework for Faithful 3D Shape Completion, 2026. URL https://arxiv.org/abs/2607.20660.

Binbin Huang, Haobin Duan, Yiqun Zhao, Zibo Zhao, Yi Ma, and Shenghua Gao. Cupid: Generative 3d reconstruction via joint object and pose modeling, 2025a. URL https://arxiv.org/abs/2510.20776.

Zehuan Huang, Yuan-Chen Guo, Xingqiao An, Yunhan Yang, Yangguang Li, Zi-Xin Zou, Ding Liang, Xihui Liu, Yan-Pei Cao, and Lu Sheng. Midi: Multi-instance diffusion for single image to 3d scene generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 23646–23657, 2025b.

Suji Kang, Seok-Young Kim, Young Bin Kim, Taewook Ha, Dieter Schmalstieg, Shohei Mori, and Woontack Woo. SnapPhysics: A Physics-Aware Scene Graph from a Single View for Interactive Mixed Reality Scenes. *arXiv preprint arXiv:2609.19815*, 2026. URL https://arxiv.org/abs/2609.19815.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. FlowEdit: Inversion-free text-based editing using pre-trained flow models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19721–19730, October 2025.

URL https://openaccess.thecvf.com/content/ICCV2025/html/Kulikov_FlowEdit_Inversion-Free_Text-Based_Editing_Using_Pre-Trained_Flow_Models_ICCV_2025_paper.html.

Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. MegaPose: 6D pose estimation of novel objects via render & compare. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 715–725. PMLR, 2023. URL https://proceedings.mlr.press/v205/labbe23a.html.

Inhee Lee, Sangwon Baik, Sungjoo Kim, Hyeonwoo Kim, Hyunsoo Cha, and Hanbyul Joo. SimuScene: Simulation-Ready Compositional 3D Scene Reconstruction from a Single Image, 2026. URL https://arxiv.org/abs/2606.03994.

Weiyu Li, Jiarui Liu, Hongyu Yan, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. CraftsMan3D: High-fidelity mesh generation with 3D native diffusion and interactive geometry refiner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5307–5317, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Li_CraftsMan3D_High-fidelity_Mesh_Generation_with_3D_Native_Diffusion_and_Interactive_CVPR_2025_paper.html.

Jiehong Lin, Lihua Liu, Dekun Lu, and Kui Jia. SAM-6D: Segment anything model meets zero-shot 6D object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27906–27916, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Lin_SAM-6D_Segment_Anything_Model_Meets_Zero-Shot_6D_Object_Pose_Estimation_CVPR_2024_paper.html.

Siyou Lin, Zhou Xue, Hongwen Zhang, Liang An, Dongping Li, Shaohui Jiao, and Yebin Liu. Mix3r: Mixing feed-forward reconstruction and generative 3d priors for joint multi-view aligned 3d reconstruction and pose estimation, 2026. URL https://arxiv.org/abs/2605.03359.

Yuchen Lin, Chenguo Lin, Panwang Pan, Honglei Yan, Yiqiang Feng, Yadong Mu, and Katerina Fragkiadaki. Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers. *arXiv preprint arXiv:2506.05573*, 2025.

Haolin Liu, Yujian Zheng, Guanying Chen, Shuguang Cui, and Xiaoguang Han. Towards high-fidelity single-view holistic reconstruction of indoor scenes. In *European Conference on Computer Vision*, pp. 429–446. Springer, 2022a. doi: 10.1007/978-3-031-19769-7_25. URL https://link.springer.com/chapter/10.1007/978-3-031-19769-7_25.

Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3D mesh in 45 seconds without per-shape optimization. In *Advances in Neural Information Processing Systems*, volume 36, pp. 22226–22246. Curran Associates, Inc., 2023a. doi: 10.52202/075280-0976. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/4683beb6bab325650db13afd05d1a14a-Abstract-Conference.html.

Yuan Liu, Yilin Wen, Sida Peng, Cheng Lin, Xiaoxiao Long, Taku Komura, and Wenping Wang. Gen6D: Generalizable model-free 6-DoF object pose estimation from RGB images. In *Computer Vision – ECCV 2022*, pp. 298–315. Springer, 2022b. doi: 10.1007/978-3-031-19824-3_18. URL https://link.springer.com/chapter/10.1007/978-3-031-19824-3_18.

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

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9970–9980, 2024.

Xiaoxuan Ma, Jiashun Wang, Nicolas Ugrinovic, Yehonathan Litman, and Kris Kitani. REST3D: Reconstructing Physically Stable 3D Scenes from a Single Image, 2026. URL https://arxiv.org/abs/2605.30338.

Yanxu Meng, Haoning Wu, Ya Zhang, and Weidi Xie. Scenegen: Single-image 3d scene generation in one feedforward pass, 2025. URL https://arxiv.org/abs/2508.15769.

Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3DUnderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 55–64, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Nie_Total3DUnderstanding_Joint_Layout_Object_Pose_and_Mesh_Reconstruction_for_Indoor_CVPR_2020_paper.html.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

SAM 3D Team, Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, Aohan Lin, Jiawei Liu, Ziqi Ma, Anushka Sagar, Bowen Song, Xiaodong Wang, Jianing Yang, Bowen Zhang, Piotr Dollár, Georgia Gkioxari, Matt Feiszli, and Jitendra Malik. Sam 3d: 3dfy anything in images, 2025. URL https://arxiv.org/abs/2511.16624.

Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pp. 1–18. Springer, 2025.

Zefan Tian, Yuteng Ye, Yiheng Zhang, Yuhang Yang, Xueqiang Lv, Shizhou Zhang, Le Liu, and Di Xu. Sceneregen: Generative reconstruction of 3d scenes from a single image, 2026. URL https://arxiv.org/abs/2608.23930.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details, 2025. URL https://arxiv.org/abs/2507.02546.

Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation, 2024. URL https://arxiv.org/abs/2311.01455.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. FoundationPose: Unified 6D pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17868–17879, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wen_FoundationPose_Unified_6D_Pose_Estimation_and_Tracking_of_Novel_Objects_CVPR_2024_paper.html.

Tianhao Wu, Chuanxia Zheng, Frank Guan, Andrea Vedaldi, and Tat-Jen Cham. Amodal3R: Amodal 3D Reconstruction from Occluded 2D Images, 2025. URL https://arxiv.org/abs/2503.13439.

Hongchi Xia, Tianhang Cheng, Wei-Chiu Ma, and Shenlong Wang. FIRE3D: Feed-forward Interactive 3D Scene Reconstruction Within A Minute. *arXiv preprint arXiv:2609.08848*, 2026. URL https://arxiv.org/abs/2609.08848.

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3d generation, 2025a. URL https://arxiv.org/abs/2512.14692.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3D latents for scalable and versatile 3D generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480, June 2025b. URL https://openaccess.thecvf.com/content/CVPR2025/html/Xiang_Structured_3D_Latents_for_Scalable_and_Versatile_3D_Generation_CVPR_2025_paper.html.

Zhengrui Xiang, Jiaqi Wu, Fupeng Sun, Heliang Zheng, and Yingzhen Li. DVD: Discrete Voxel Diffusion for 3D Generation and Editing, 2026. URL https://arxiv.org/abs/2605.07971.

Kaixin Yao, Longwen Zhang, Xinhao Yan, Yan Zeng, Qixuan Zhang, Wei Yang, Lan Xu, Jiayuan Gu, and Jingyi Yu. Cast: Component-aligned 3d scene reconstruction from an rgb image, 2025. URL https://arxiv.org/abs/2502.12894.

Junliang Ye, Shenghao Xie, Ruowen Zhao, Zhengyi Wang, Hongyu Yan, Wenqiang Zu, Lei Ma, and Jun Zhu. Nano3d: A training-free approach for efficient 3d editing without masks, 2025. URL https://arxiv.org/abs/2510.15019.

Ze-Xin Yin, Liu Liu, Xinjie Wang, Wei Sui, Zhizhong Su, Jian Yang, and Jin Xie. 3d-fixer: Coarse-to-fine in-place completion for 3d scenes from a single image, 2026. URL https://arxiv.org/abs/2604.04406.

Andrii Zadaianchuk, Leonardo Barcellona, Lennard Schuenemann, Christian Gumbsch, Zehao Wang, Muhammad Zubair Irshad, Fabien Despinoy, Rahaf Aljundi, Stratis Gavves, and Sergey Zakharov. Reconstruction by generation: 3d multi-object scene reconstruction from sparse observations, 2026. URL https://arxiv.org/abs/2604.27106.

Bin Zang, Wenting Zheng, Xiaoliang Luo, Zhiyuan Fang, Shi Li, Lvchun Wang, Wei Yu, Yi Zhao, Tian Xie, Yuchi Huo, and Rengan Xie. HIVE-3D: Hierarchical Voxel Enhancement for High-Quality 3D Scene Generation. *arXiv preprint arXiv:2607.13468*, 2026. URL https://arxiv.org/abs/2607.13468.

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)*, 43(4):1–20, 2024.

Yiheng Zhang, Zhuojiang Cai, Mingdao Wang, Meitong Guo, Tianxiao Li, Li Lin, and Yuwang Wang. M3DLayout: A multi-source dataset of 3D indoor layouts and structured descriptions for 3D generation, 2025. URL https://arxiv.org/abs/2509.23728.

Yuchen Zhang, Luanyuan Dai, Yiwei Wang, Xiwei Xu, Jianing Zhang, Johnny. r. zhang, Xianhui Meng, Yanbiao Ma, Jiayi Ma, and Xiaoshuai Hao. Robust 3D Alignment of Generative Reconstructions via Partial Monocular Observations, 2026. URL https://arxiv.org/abs/2607.00498.

Qingcheng Zhao, Xiang Zhang, Haiyang Xu, Zeyuan Chen, Jianwen Xie, Yuan Gao, and Zhuowen Tu. DepR: Depth guided single-view scene reconstruction with instance-level diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5722–5733, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zhao_DepR_Depth_Guided_Single-view_Scene_Reconstruction_with_Instance-level_Diffusion_ICCV_2025_paper.html.

Junsheng Zhou, Yu-Shen Liu, and Zhizhong Han. Zero-shot scene reconstruction from single images with deep prior assembly. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv preprint arXiv:1801.09847*, 2018. URL https://arxiv.org/abs/1801.09847.

A SUPPLEMENTARY MATERIAL

This appendix connects the mechanisms in the main paper to their implementation and evaluation. We derive the visible region-based metric-scale calibrator, specify how heterogeneous latent infusion and image conditioning are implemented, and detail the comparisons of calibration, object fidelity, occlusion, depth noise, and scene scaling. Additional shaded and textured renderings show which differences from the baselines concern geometry, material appearance, or object placement.

A.1 DERIVATION OF METRIC-SCALE CALIBRATION

Variables and projection model. The calibrator retains the upstream object's rotation and estimates a uniform scale s and translation $t := (\Delta X, \Delta Y, \Delta Z)$. This separates metric-dependent pose components from the generator's orientation prior. For the derivation below, we use a shared focal length f and coordinates centered at the principal point, so a camera-space point satisfies $x = fX/Z$. Here X is its horizontal coordinate and Z its forward depth in the right/down/forward camera frame. Sec. 3.2 gives the corresponding formulation with separate focal lengths and explicit principal-point offsets.

Projected boundary constraint. The first constraint aligns the projected extent of the generated mesh M_i with the visible-region bounding box b_i. After applying the prescribed rotation and camera-forward initialization, we render the mesh in isolation and extract its upper, lower, left, and right projected extrema (x_*, y_*) and corresponding surface points (X_*, Y_*, Z_*). Isolated rendering exposes the generated object's extent even where another scene object would occlude it. Scaling about the vertex mean and translating the mesh transforms a surface point as follows:

$$\begin{aligned} &\mathcal{T}_p((X,Y,Z),(\Delta X,\Delta Y,\Delta Z,s)) = \\ &(X_{\text{avg}} + \Delta X + s(X - X_{\text{avg}}), \\ &Y_{\text{avg}} + \Delta Y + s(Y - Y_{\text{avg}}), \\ &Z_{\text{avg}} + \Delta Z + s(Z - Z_{\text{avg}})), \end{aligned} \tag{10}$$

The quantities X_{avg}, Y_{avg}, and Z_{avg} denote the mean vertex coordinates; $\mathcal{T}_p$ is the resulting pointwise affine transformation. Within one update, we hold the extracted extrema fixed as approximations to the transformed projected extrema. Substituting the transformed leftmost point into the pinhole model and matching the left edge of b_i gives Eq. 11. Algorithm 1 collects this equation and the remaining constraints into the weighted linear system used for one update.

$$\frac{f(X_{\text{avg}} + \Delta X + s(X_{\text{left}} - X_{\text{avg}}))}{Z_{\text{avg}} + \Delta Z + s(Z_{\text{left}} - Z_{\text{avg}})} = b_{\text{left}}^{(x)} \tag{11}$$

The instance index is omitted, and $b_{\text{left}}^{(x)}$ is the horizontal coordinate of the observed left boundary. The right, upper, and lower boundaries give analogous equations. Multiplying by the projection denominator makes each boundary equation linear in s and t. This construction uses the observed image extent directly, without estimating pointwise correspondences between a generated surface and a partial pointmap.

Visible-depth constraint. Projected extent alone cannot determine metric size and distance: jointly scaling an object and its distance from the camera can preserve its projection. The second constraint matches the rendered mean visible depth to the mean finite positive depth within the observed instance mask, denoted by $\tilde{d}_i$. This adds a metric anchor to the four image-space constraints:

$$\sum_{p \in \mathcal{V}} \left(Z_{\text{avg}} + \Delta Z + s(Z_p - Z_{\text{avg}})\right) / |\mathcal{V}| = \tilde{d}_i \tag{12}$$

The set $\mathcal{V}$ contains visible surface samples corresponding to foreground pixels with finite positive depth in the current isolated rendering, with equal weight per valid pixel. It is distinct from the



Figure 5: Additional comparisons with SAM3D across household and industrial scenes. The shaded geometry and textured renderings show how FOMO combines structural completion with sharper local detail, including the fan grille, animal fur, and forklift frame. In the forklift–box example (bottom row), FOMO more faithfully preserves the plastic wrapping around the boxes visible in the input image, including its crinkled surface appearance. Corresponding views separate changes in geometry from changes in appearance; highlighted regions indicate details that are smoothed or distorted in the baseline.

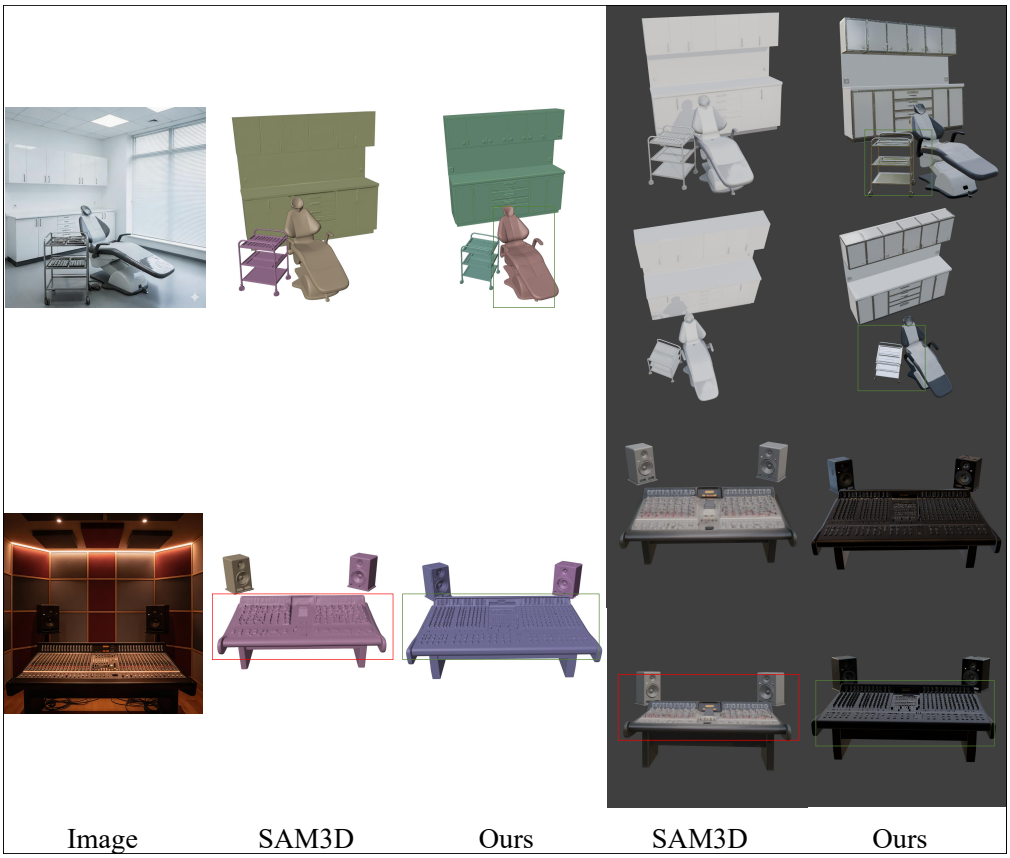

Figure 6: Geometry and appearance fidelity compared with SAM3D in a medical room and recording studio. In the recording-studio example, FOMO more faithfully preserves the dense arrangement of individual buttons, knobs, and sliders visible on the input audio mixer. The shaded views resolve these controls as distinct geometric structures, while the textured views show their surface appearance; zoom in for better visualization. The medical-room example additionally shows finer chair upholstery and reflections on the metal cart.

Algorithm 1 One weighted least-squares update of the visible region-based metric-scale calibrator. Four projected boundary equations constrain the image-space extent, and one visible-depth equation anchors scale and translation; the upstream rotation is retained. The projected extrema and visible samples are refreshed between updates. Instance indices are omitted.

Require: X_{avg}, Y_{avg}, Z_{avg}, $\{X_*\}$,$\{Y_*\}$,$\{Z_*\}$,$\{b_*^{(x)}\}$, $\{b_*^{(y)}\}$, $\forall * \in \{\text{left}, \text{right}, \text{upper}, \text{lower}\}$, $\tilde{d}$, f, Z, W, H ▷ Z is the isolated object depth render; W, H are original image dimensions

$dX_*, dY_*, dZ_* \leftarrow X_* - X_{\text{avg}}, Y_* - Y_{\text{avg}}, Z_* - Z_{\text{avg}}$

$\mathcal{V} \leftarrow \{p \mid Z_p > 0 \text{ and } Z_p \text{ is finite}\}$

$$A \leftarrow \begin{bmatrix} f & 0 & -b_{\text{left}}^{(x)} & f \cdot dX_{\text{left}} - b_{\text{left}}^{(x)} \cdot dZ_{\text{left}} \\ f & 0 & -b_{\text{right}}^{(x)} & f \cdot dX_{\text{right}} - b_{\text{right}}^{(x)} \cdot dZ_{\text{right}} \\ 0 & f & -b_{\text{upper}}^{(y)} & f \cdot dY_{\text{upper}} - b_{\text{upper}}^{(y)} \cdot dZ_{\text{upper}} \\ 0 & f & -b_{\text{lower}}^{(y)} & f \cdot dY_{\text{lower}} - b_{\text{lower}}^{(y)} \cdot dZ_{\text{lower}} \\ 0 & 0 & 1 & \frac{1}{|\mathcal{V}|} \sum_{p \in \mathcal{V}} (Z_p - Z_{\text{avg}}) \end{bmatrix}$$

$$B \leftarrow \begin{bmatrix} b_{\text{left}}^{(x)} \cdot Z_{\text{avg}} - f \cdot X_{\text{avg}} \\ b_{\text{right}}^{(x)} \cdot Z_{\text{avg}} - f \cdot X_{\text{avg}} \\ b_{\text{upper}}^{(y)} \cdot Z_{\text{avg}} - f \cdot Y_{\text{avg}} \\ b_{\text{lower}}^{(y)} \cdot Z_{\text{avg}} - f \cdot Y_{\text{avg}} \\ \tilde{d} - Z_{\text{avg}} \end{bmatrix}$$

$\Lambda \leftarrow \text{diag}(2/W, 2/W, 2/H, 2/H, 1)$

Solve $\min_X \|\Lambda(AX - B)\|_2^2$, where $X = \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \\ s \end{bmatrix}$

observed masked pixels used to compute $\tilde{d}_i$; no pointwise correspondence between these sets is required. Its mean depth also differs from Z_{avg}, which averages all mesh vertices and defines the center of scaling. We hold the rendered samples fixed within an update and recompute them after transforming the mesh. Matching mean depth is the mechanism by which local depth fluctuations can be averaged, while the projected boundaries retain the image-space extent constraint.

Iterative solution. The four boundary equations and the depth equation form five constraints on four unknowns. We fit them jointly with the weighted least-squares formulation in Eq. 9, then rerender the mesh to update its vertex mean, extrema, and visible samples. Iteration addresses the approximation that the extremal points remain fixed during a single solve. We use a fixed budget of five scale–translation updates per object, rerendering after each update; we do not stop early based on a residual-convergence threshold. If an iterate has fewer than four valid foreground samples after the visibility safeguards below, we stop and restore the best previously valid iterate according to normalized bounding-box error plus relative mean-depth error. A nonfinite solution, nonpositive scale, or absence of a valid visible iterate is treated as a failure. Each step is a small linear solve, accounting for the low calibration overhead alongside instance generation and image conditioning.

A.2 Implementation Details

For metric-scale calibration, we evaluate eight upstream generators: CUPID (Huang et al., 2025a), MIDI (Huang et al., 2025b), SceneGen (Meng et al., 2025), SAM3D (SAM 3D Team et al., 2025), RecGen (Zadaianchuk et al., 2026), Gen3DSR (Ardelean et al., 2024), DepR (Zhao et al., 2025), and PartCrafter (Lin et al., 2025). We pair each generator with four calibrators: None, ICP (Besl & McKay, 1992), SAM3D-post (SAM 3D Team et al., 2025), and Ours. Throughout the paper, ICP uses Open3D's point-to-point estimator with `with_scaling=True` (Zhou et al., 2018), jointly optimizing rotation, translation, and uniform scale for each object. This provides a strong seven-degree-of-freedom $\text{Sim}(3)$ baseline that can correct both pose and scale errors. For heterogeneous latent infusion, we use three models with strong occlusion-aware mesh generation ability, namely CUPID, RecGen, and SAM3D, as sources of active voxel coordinates and two SLAT-based mesh generators, namely TRELLIS (Xiang et al., 2025b) and TRELLIS.2 (Xiang et al.,

2025a), as detail-rich refiners. We denote the resulting FOMO configurations, namely our metric-scale calibration plus extra refinement by TRELLIS and TRELLIS.2, by FOMO.T and FOMO.T.2, respectively.

Hardware and randomization. The cross-generator calibration and evaluation runs use 8 RTX 3090 GPUs. Evaluation uses launcher seed 0, while the individual methods retain their specified randomization settings: object-wise ICP uses seed 42, and DepR uses base seed 0 on 3D-FRONT and 0 on M3DLayout, with stable per-room seeds passed to NumPy and PyTorch. Gen3DSR retains its native RANSAC (Fischler & Bolles, 1981) defaults: its regressors do not set `random_state`, and the wrapper seed does not enter the helper's global NumPy state. Those estimates can therefore vary between runs. Runtime figures describe the configurations and amortization stated in the main paper.

Structural support and detail synthesis. Each source generator receives the RGB image, visible instance masks, and any GT geometric inputs required by its native interface. We retain its mesh, pose metadata, and the active voxel coordinates used by its feature sampler. Using the native sparse coordinates preserves the structural support on which its occlusion-aware completion is expressed. For refinement, these coordinates condition the feature sampler of TRELLIS or TRELLIS.2; each refiner then uses its own decoder and material exporter. The no-refinement branch retains the source mesh. The generators exchange structural support rather than learned feature vectors, implementing the structure–feature decomposition in Sec. 3.1.

Coordinate conventions and transfer. The sparse tensor stores integer coordinates in (b, x, y, z) order, with $b = 0$ for a single object. For source resolution n_g, the center of voxel $\mathbf{c} \in \{0, \dots, n_g - 1\}^3$ is

$$\mathbf{p}_g = \frac{\mathbf{c} + \frac{1}{2}\mathbf{1}}{n_g} - \tfrac{1}{2}\mathbf{1}, \qquad \mathbf{c}_r = \left\lfloor n_r \left(Q_{g\to r}\mathbf{p}_g + \tfrac{1}{2}\mathbf{1} \right) \right\rfloor .$$

Here $Q_{g\to r}$ is a fixed coordinate-axis conversion between generator and refiner grids, distinct from the object's scene orientation R_i. The evaluated CUPID, RecGen, and SAM3D sources expose 64^3 grids, and both refiners consume active coordinates at $n_r = 64$; their six pairwise conversions use $Q_{g\to r} = I_3$. Thus transfer preserves the original active indices and their ordering, without mesh voxelization, interpolation, or occupancy resampling. Coordinates outside the declared grid, duplicate coordinates, and mismatched axis declarations are rejected. The native export convention maps normalized voxel coordinates through $(x, y, z) \mapsto (x, z, -y)$; the source object's fixed orientation is then applied once to place the decoded mesh in the camera frame before scale–translation calibration. TRELLIS.2's 1024^3 geometry decoding resolution is distinct from the 64^3 latent coordinate grid. This explicit grid and frame agreement, together with the refiner's ability to sample features at supplied active coordinates, determines compatibility.

Identity-preserving image conditioning. The complete coarse-mesh rendering supplies object shape, while the target-marked scene image supplies instance-specific appearance and context. We use Seedream 5.0 Pro (ByteDance Seed, 2026), a multimodal image generation and editing model, as the identity-preserving inpainter to combine these observations into a complete, detail-rich object image for the refiner. The conditioning package includes an RGB crop, a binary full-object mask, and an RGBA crop. For refiner input, we crop around the foreground alpha bounds, pad to a square with a side-length factor of 1.2 for TRELLIS and 1.0 for TRELLIS.2, and resize with Lanczos interpolation to 518×518 and 1024×1024 pixels, respectively. The alpha crop is composited over black before extracting the refiner image condition. Direct mesh-render and segmentation-crop inputs bypass this step in the visual ablation: the former carries coarse appearance, while the latter can be incomplete under occlusion. The comparisons in Fig. 33–51 examine these alternatives using the same source-generator and refiner combinations.

Geometric observations. For the controlled benchmark comparison, visible pointmaps are back-projected from GT camera-Z depth using the original intrinsics. A valid pixel (u, v) with depth z maps to $z((u - c_x)/f_x, (v - c_y)/f_y, 1)$ before the prescribed frame conversion. Instance masks select the visible surface used for calibration; complete GT mesh samples are reserved for geometric evaluation. Input interfaces remain method-specific: CUPID and MIDI use image and mask

conditioning; RecGen, SAM3D, and DepR also use depth or pointmap information; Gen3DSR uses the camera and pointmap to construct reprojected crops. All these inputs are obtained from mesh rendering and thus are ground-truth. This controlled setting isolates generation and calibration from errors in upstream mask or depth estimation and ensures fair comparison. For qualitative scenes without GT depth, we use MoGe (Wang et al., 2025).

Calibration baselines. Our ICP baseline (Besl & McKay, 1992) uses Open3D's point-to-point registration (Zhou et al., 2018), with `TransformationEstimationPointToPoint` configured with `with_scaling=True`. It independently aligns samples from each generated mesh to the visible pointmap associated with its GT object by estimating a similarity transform $\mathbf{x}' = sR\mathbf{x}+\mathbf{t}$, where $s > 0$ is a uniform scale, $R \in \mathrm{SO}(3)$ is a rotation, and $\mathbf{t} \in \mathbb{R}^3$ is a translation. All seven degrees of freedom are optimized, making ICP a strong $\mathrm{Sim}(3)$ baseline with explicit scale estimation. Object-to-mask associations are established before optimization; point correspondences are updated during ICP, and each object receives its own transform. The target is the observed partial surface, rather than the complete GT mesh or a merged scene cloud. SAM3D-post (SAM 3D Team et al., 2025) uses masked-pointmap initialization and silhouette pose refinement, retaining its configured update-rejection rules with shape-ICP disabled. Ours retains the supplied orientation and solves for four variables—uniform scale and translation—using projected extent and mean visible depth. The comparison therefore tests how point correspondences and visible-region constraints use partial geometric evidence when both calibrators can adjust scale.

Camera-forward initialization. Our calibration initialization uses one shared translation $(0, 0, d_z)$, with $d_z \geq 0$, to make the generated objects renderable in the fixed source camera. We choose the smallest displacement satisfying the near-plane and image-frustum inequalities for all vertices, with a one-pixel image margin, and extend the far plane if required. No lateral translation or scale change is introduced at this step, so the initial relative layout is preserved. All vertices must lie in front of the camera.

Rasterization and linear updates. We use PyRender with an EGL backend and first render at the original image resolution. If fewer than four foreground pixels are obtained, we sample a padded region within the original image bounds more densely, starting with 32 samples per crop dimension and doubling up to 1024 samples per framebuffer dimension. The source camera, scene pose, and image domain remain fixed. For crop origin (u_0, v_0) and sampling factors (a_x, a_y), the intrinsics become $(a_x f_x, a_y f_y, a_x(c_x - u_0), a_y(c_y - v_0))$, and rendered coordinates map back through $u = u_0 + u'/a_x$, $v = v_0 + v'/a_y$. Clipping planes may tighten around the complete transformed object.

Finite positive-depth samples determine the projected extrema and mean visible depth. The solver uses the original camera intrinsics and image dimensions, with the row weights in Eq. 9. Each object is updated independently through a float64 `numpy.linalg.lstsq` solve of five equations in four variables. The source mesh is retained, and calibrated outputs record the transforms, initialization, rasterization settings, and solver status.

Prompt template. The following prompt is provided to Seedream 5.0 Pro: a rendering of the complete coarse mesh is supplied as the first image, followed by the original scene image with the target bounding box.

```
Generate one detailed image of the complete, unoccluded target object from the two reference images.

Image 1 is a rendering of the target object's complete coarse mesh. Use it as a guide to the object's overall shape, proportions, constituent parts, and viewing direction.
Image 2 is the original scene image, with a bounding box highlighting the same target object. Use its visible regions to preserve the object's identity, colors, materials, texture, and fine details. The box identifies the target; other objects or background regions inside the box are not part of it.
```

```
Imagine how this specific object would look without anything blocking it. Complete the hidden regions consistently with the mesh structure and the visible evidence in the scene. Preserve observed details while replacing coarse rendering artifacts with coherent, realistic surface detail. Keep the object's design and part configuration consistent with the references.

Show only the target object, fully within the image, with its complete silhouette and constituent parts visible from the mesh-rendering viewpoint. Remove foreground occluders and surrounding scene content. Do not attach, merge, duplicate, or invent components that do not belong to the target. Include a previously hidden part only when it is consistent with the target's structure and identity.

Place the isolated object on a plain white background with clear, even lighting and a small margin on every side. Do not crop any part of the object. Do not include the bounding box, annotations, text, additional objects, or multiple views. Output only the completed object image.
```

Depth-noise comparison. For Tab. 3, we add Gaussian noise $\epsilon \sim \mathcal{N}(0, 0.1^2)$ to input depth before generation and rerun the complete pipeline, recomputing depth-derived inputs and keeping the seeds and configuration fixed between each clean/noisy pair. We verify that SAM3D produces the same mesh geometry in these paired runs: depth and pointmaps affect its pose-generation branch, while its mesh-generation branch is independent of these inputs. Thus, the changes in scene-level metrics reflect sensitivity in the pose-estimation and calibration pathway, including SAM3D's native pose prediction for None. Each row retains the same refinement choice in its clean and noisy runs. The clean references are SAM3D with the corresponding calibrator in Tab. 1 for None, ICP, and SAM3D-post, and SAM3D+FOMO.T.2 in Tab. 2 for Ours. Ours has smaller changes in all three metrics: CD-S increases by 0.004, compared with 0.0643–0.0910; FScore-S falls by 1.42 percentage points, compared with 19.17–24.51; and IoU-B falls by 0.026, compared with 0.176–0.207. FScore-S is expressed in percent in this table, so its differences are percentage points.

Calibration and full-framework comparisons. Tab. 1 applies each calibrator to a fixed source prediction for a given generator and dataset, isolating the alignment procedure. Its scene metrics assess whether corrected scale and translation improve overall geometry and object layout. Tab. 2 additionally introduces heterogeneous latent infusion and therefore evaluates complete FOMO configurations against the base generators. CD-O and FScore-O are evaluated on generated meshes before metric-scale calibration, with object centers aligned. Calibration-only configurations therefore share the object-level scores of their source generator; refinement is assessed by evaluating the refined mesh under the same pre-calibration protocol. The shaded detail views and image-conditioning ablation provide complementary evidence about the contribution of refinement.

A.3 Evaluation Protocol

Datasets and comparison scope. The benchmark evaluations cover all test rooms from MIDI's 3D-FRONT split and 1000 scenes from M3DLayout. The calibration matrix crosses eight source generators—CUPID, MIDI, SceneGen, SAM3D, RecGen, Gen3DSR, DepR, and PartCrafter—with None, ICP, SAM3D-post, and Ours, yielding 64 generator–calibrator–dataset configurations. Each comparison reuses the same generated meshes across calibration settings. The latent-infusion comparison uses CUPID, RecGen, and SAM3D with the TRELLIS and TRELLIS.2 refiners, denoted by FOMO.T and FOMO.T.2. AI-generated scene images broaden the qualitative assessment of clutter, partial visibility, fine detail, and material appearance; they are not additional metric-GT benchmarks.

Single-image task definition and auxiliary inputs. Our use of "single-image-based scene generation" follows established usage in this literature: one RGB image supplies the visual observation, while pretrained perception models can extract auxiliary instance masks and geometric cues from that same image. For example, MIDI (Huang et al., 2025b), explicitly presented as *Multi-Instance Diffusion for Single Image to 3D Scene Generation*, describes an inference pipeline that first obtains

object masks with Grounded-SAM (its Supplementary Sec. 2). Grounded-SAM combines Grounding DINO for detection with SAM for segmentation (Ren et al., 2024). Thus, using pretrained detection and segmentation is part of an established single-image generation pipeline.

Similarly, the SAM 3D abstract describes its task as "predicting geometry, texture, and layout from a single image" (SAM 3D Team et al., 2025). Its experimental settings use a pointmap-conditioned geometry model, estimate pointmaps with MoGe when they are unavailable, and specify ground-truth depth/pointmaps as the reference for quantitative layout evaluation (SAM 3D Team et al., 2025, Sec. 4). These settings distinguish the source image, auxiliary geometric observations, and the reference used for quantitative evaluation.

Following this convention, FOMO uses a single scene image and can obtain the required auxiliary observations through pretrained detection, segmentation, and monocular geometry estimation. Our quantitative experiments supply GT camera intrinsics, instance masks, and depth or pointmaps as required by each method, providing controlled observations for evaluating generation and calibration (Sec. 4.1). In practical image-based use, pretrained models can supply these auxiliary inputs from the source image; our qualitative examples use MoGe for geometric observations, as described in Sec. A.2. We therefore distinguish the single-image task designation from the source of auxiliary inputs in each evaluation setting, consistently with the above prior work.

Metric definitions and weighting. CD is the unsquared bidirectional nearest-neighbor distance in metres, and FScore uses a $0.1\,\mathrm{m}$ distance threshold. CD-S and FScore-S compare scene point sets; IoU-B averages the axis-aligned bounding-box overlap of matched objects within a scene. The scene metrics are then averaged with equal weight per room. CD-O and FScore-O compare individual generated meshes with GT before applying metric-scale calibration, after aligning their AABB centers and without scale normalization; successful pairs receive equal weight. For a refined configuration, we evaluate the refined mesh at this same stage, so calibration transforms do not enter either object-level metric. CD-O-raw instead evaluates each object in the scene frame after applying the specified calibration setting, without center alignment, to retain placement error. FScore and IoU lie in $[0, 1]$ unless a table explicitly uses percentages. These complementary measures distinguish local object fidelity from metric-scale scene alignment, the two objectives of our framework.

Occlusion measurement. We render each GT object both alone and with the objects present in its source RGB image, using a common camera and rasterization convention. For a positive isolated projection area, the occlusion ratio is $\mathrm{clip}(1 - n_{\mathrm{visible}}/n_{\mathrm{isolated}}, 0, 1)$. The zero group includes equal visible and isolated counts and negative raw ratios clipped to zero when rasterization produces more visible than isolated samples. Eligible objects must be projectable, present in the source render, and associated with an input GT mask identifier; eligibility is defined independently of prediction success. Tab. 12 summarizes the resulting evaluation population. The interval-specific comparisons expose behavior on the less frequent occluded objects that can be obscured by a single overall mean.

Two questions under occlusion. The calibration and latent-infusion analyses ask complementary questions: whether the generated objects are placed accurately, and whether their individual geometry retains fidelity under partial visibility. Both use six disjoint intervals, interpreted as 0%, $(0, 20)\%$, $[20, 40)\%$, $[40, 60)\%$, $[60, 80)\%$, and $[80, 100]\%$. Every object belongs to one interval; these labels denote ranges of occlusion rather than cumulative thresholds.

Why calibration uses CD-O-raw. CD-O-raw evaluates each object in the scene coordinate frame with its predicted scale, rotation, and translation retained. Center correction would remove translation errors that the calibrator is intended to fix, while a single CD-S value mixes objects with different visibility. The raw object metric therefore supports occlusion-specific comparisons of metric-scale placement. We compare None, ICP, SAM3D-post, and Ours on all the selected rooms in 3D-FRONT and M3DLayout, and report mean error for each generator and interval in Tab. 5. Fig. 3 shows SAM3D, MIDI, and RecGen from left to right; Tab. 5 contains all seven generator comparisons. The reported values in Tab. 5 show lower CD-O-raw for Ours than for None and SAM3D-post in every generator–interval comparison. Ours also matches or improves on the displayed ICP values throughout the CUPID, MIDI, SAM3D, RecGen, Gen3DSR, and DepR comparisons. For example, SAM3D has CD-O-raw 0.061 with Ours in the highest-occlusion interval, compared with 0.109 for None and 0.093 for ICP. ICP has lower error for SceneGen in the first two intervals, while Ours

Table 5: Metric-scale placement under input-image occlusion on 3D-FRONT. Each cell lists CD-O-raw in metres for the disjoint intervals 0%, $(0, 20)\%$, $[20, 40)\%$, $[40, 60)\%$, $[60, 80)\%$, and $[80, 100]\%$, from top to bottom; lower is better. None applies no additional calibration; Ours uses FOMO's visible-boundary and visible-depth constraints. Bold marks the lowest displayed value within each generator and interval, including ties.

Stage 1 model	None	ICP	SAM3D-post	Ours
CUPID	6.173	0.161	4.466	**0.155**
	5.638	**0.167**	4.227	**0.167**
	5.685	0.170	4.390	**0.169**
	5.683	0.172	4.425	**0.169**
	6.143	0.173	4.906	**0.169**
	6.169	0.174	4.958	**0.170**
MIDI	7.167	0.080	4.102	**0.051**
	7.066	0.087	4.573	**0.054**
	7.056	0.091	4.777	**0.057**
	7.049	0.091	4.834	**0.057**
	7.060	0.091	4.888	**0.058**
	7.052	0.093	4.926	**0.060**
SAM3D	0.090	0.079	0.103	**0.052**
	0.099	0.088	0.111	**0.055**
	0.105	0.090	0.115	**0.058**
	0.107	0.091	0.117	**0.059**
	0.107	0.091	0.117	**0.059**
	0.109	0.093	0.119	**0.061**
RecGen	0.094	0.090	0.112	**0.085**
	0.103	0.094	0.120	**0.093**
	0.107	**0.096**	0.123	**0.096**
	0.107	0.097	0.122	**0.096**
	0.108	**0.097**	0.123	**0.097**
	0.109	0.099	0.124	**0.098**
Gen3DSR	0.131	0.092	0.141	**0.088**
	0.153	0.107	0.163	**0.103**
	0.164	0.111	0.174	**0.108**
	0.165	0.112	0.175	**0.109**
	0.165	0.112	0.174	**0.109**
	0.165	0.114	0.174	**0.111**
DepR	6.602	0.469	4.479	**0.110**
	6.718	0.447	4.944	**0.116**
	6.816	0.447	5.194	**0.117**
	6.834	0.451	5.258	**0.116**
	6.858	0.455	5.313	**0.116**
	6.869	0.461	5.356	**0.117**
SceneGen	6.727	**0.098**	3.786	0.103
	7.135	**0.104**	4.072	0.107
	6.572	0.112	4.177	**0.110**
	6.683	0.120	4.219	**0.110**
	6.672	0.122	4.391	**0.114**
	6.784	0.130	4.528	**0.115**

gives lower error in the remaining four. These comparisons support visible-region calibration across the evaluated generators while retaining the cases in which another calibrator performs better.

A.4 Calibration under Bounding-Box Truncation

This experiment examines how reductions in an object's visible bounding-box width and height affect calibration on 3D-FRONT. For each object in the evaluated scenes, we obtain its GT mask from an isolated rendering and its visible GT instance mask from the full scene, using the same camera, object pose, and rasterization settings. Let $(w_{\text{iso}}, h_{\text{iso}})$ and $(w_{\text{vis}}, h_{\text{vis}})$ denote the widths and heights of their axis-aligned 2D bounding boxes. We define horizontal and vertical truncation rates as

$$r_x = 1 - \frac{w_{\text{vis}}}{w_{\text{iso}}}, \qquad r_y = 1 - \frac{h_{\text{vis}}}{h_{\text{iso}}}. \tag{13}$$

For example, reducing the visible width from 1 to 0.8 gives $r_x = 20\%$. These rates measure the loss of projected extent along each axis. The $(r_x, r_y) = (0, 0)$ group retains the full bounding-box width and height, but can still contain occlusion within that box.

We independently bin each axis into 0%, $(0, 25]\%$, $(25, 50]\%$, and $(50, 75]\%$, yielding the 16 joint conditions in Tab. 7. The grid covers objects with a nonempty visible mask and both truncation rates

Table 6: Performance of different FOMO variants under occlusion. CD-O values are reported here in different occlusion intervals. The occlusion intervals are the same as those in Tab. 5.

Base model	None	FOMO.T	FOMO.T.2
CUPID	0.140	0.137	**0.130**
	0.148	0.137	**0.135**
	0.154	0.140	**0.133**
	0.154	0.143	**0.137**
	0.152	0.146	**0.138**
	0.153	0.147	**0.138**
RecGen	0.126	0.117	**0.109**
	0.124	0.117	**0.107**
	0.125	0.119	**0.110**
	0.126	0.122	**0.111**
	0.130	0.127	**0.111**
	0.136	0.129	**0.115**
SAM3D	0.104	0.101	**0.082**
	0.104	0.102	**0.080**
	0.108	0.103	**0.082**
	0.111	0.106	**0.083**
	0.112	0.104	**0.085**
	0.112	0.111	**0.090**

at most 75%. Bin membership is determined from GT masks and is shared by both calibrators. We apply Ours and ICP to the same SAM3D-generated object meshes and compare the same matched objects within each cell. Each entry gives the equally weighted mean CD-O-raw for Ours followed by that for ICP, in metres. As defined in Sec. A.3, CD-O-raw evaluates calibrated objects in the scene coordinate frame without center alignment, retaining scale, rotation, and translation errors.

Table 7: Calibration sensitivity to horizontal and vertical bounding-box truncation on 3D-FRONT with SAM3D. Columns group horizontal truncation r_x (width reduction); rows group vertical truncation r_y (height reduction), as defined in Eq. 13. Each cell has the format Ours/ICP and gives mean CD-O-raw in metres on the same matched objects; lower is better. Zero truncation along an axis means that the bounding-box extent is preserved along that axis.

Vertical truncation r_y	Horizontal truncation r_x			
	0%	$(0, 25]\%$	$(25, 50]\%$	$(50, 75]\%$
0%	0.049 / 0.084	0.052 / 0.091	0.057 / 0.099	0.064 / 0.097
$(0, 25]\%$	0.053 / 0.088	0.056 / 0.095	0.060 / 0.101	0.068 / 0.110
$(25, 50]\%$	0.055 / 0.098	0.058 / 0.105	0.064 / 0.105	0.070 / 0.112
$(50, 75]\%$	0.062 / 0.099	0.066 / 0.112	0.071 / 0.114	0.078 / 0.120

Calibration accuracy under boundary truncation. Ours achieves lower CD-O-raw than ICP in all 16 groups in Tab. 7, with relative reductions of 34.0%–44.8%. When both bounding-box dimensions are preserved, the errors are 0.049 for Ours and 0.084 for ICP. With both axes truncated by $(50, 75]\%$, Ours retains a 35.0% reduction, with 0.078 compared with ICP's 0.120. For Ours, error increases monotonically with truncation along either axis when the other axis's bin is held fixed. This trend identifies boundary loss as a relevant source of calibration difficulty while showing a consistent advantage over ICP throughout the tested conditions.

Boundary visibility as a diagnostic criterion. Grouping by boundary truncation reveals a wider range of calibration errors than grouping by occluded area alone in the reported comparisons. For the same SAM3D source model, Ours spans 0.049–0.078 in Tab. 7, compared with 0.052–0.061 across the area-based occlusion intervals in Tab. 5. The range of bin means therefore expands from 0.009 to 0.029 metres, approximately $3.2\times$, exposing both a lower-error group and a higher-error group. ICP likewise spans 0.084–0.120 under boundary grouping, compared with 0.079–0.093 under area grouping. Occluded-area ratio measures the amount of missing foreground, while r_x and r_y measure loss of the projected extents used by our calibration constraints. The wider separation of bin means and the ordered dependence on both axes support boundary visibility as a more discrimi-

native criterion for analyzing our calibrator's sensitivity on this benchmark. The range comparison is descriptive, since the two partitions differ in their numbers of bins and coverage.

A.5 Image Conditioning and Heterogeneous Latent Infusion

Tab. 8 reports a 2×2 ablation on 3D-FRONT that separates image conditioning from heterogeneous latent infusion (HLI). Segmentation uses the observed RGB object crop extracted with its instance mask; inpainting uses the complete object image produced by the identity-preserving inpainter in Eq. 3. Direct TRELLIS.2 predicts its own active voxel coordinates from the image condition, whereas HLI supplies active voxel coordinates from SAM3D to TRELLIS.2's feature sampler. All four configurations use TRELLIS.2 for feature synthesis and mesh decoding. The two inpainting configurations share the same SAM3D-guided inpainter output for each object, and the two HLI configurations share the same SAM3D active voxel coordinates.

Comparing direct TRELLIS.2 with HLI under the same image condition tests the contribution of transferred structural support. Comparing segmentation with inpainting under the same generation setting tests the contribution of the completed image condition. Inpainting + HLI is the SAM3D+TRELLIS.2 instance-generation configuration of FOMO. We follow Sec. A.3: CD-O and FScore-O evaluate generated meshes before metric-scale calibration, after aligning AABB centers and without scale normalization.

Table 8: Ablation of image conditioning and HLI on 3D-FRONT. The coordinate source distinguishes direct TRELLIS.2 generation from HLI with SAM3D structure. Combining inpainting with HLI achieves the best result on both metrics, marked in bold. CD-O is measured in metres; FScore-O lies in $[0, 1]$ and uses a $0.1\,\mathrm{m}$ distance threshold.

Configuration	Coordinate source	CD-O↓	FScore-O↑
Segmentation + TRELLIS.2	TRELLIS.2	0.201	0.572
Segmentation + HLI	SAM3D	0.097	0.881
Inpainting + TRELLIS.2	TRELLIS.2	0.115	0.902
Inpainting + HLI (FOMO)	SAM3D	**0.082**	**0.938**

Results and interpretation. The combined Inpainting + HLI configuration achieves the lowest CD-O (0.082) and highest FScore-O (0.938), reducing CD-O by 59.2% relative to Segmentation + TRELLIS.2. HLI improves both metrics under each image condition: with segmentation, CD-O decreases from 0.201 to 0.097 and FScore-O increases from 0.572 to 0.881; with inpainting, CD-O decreases from 0.115 to 0.082 and FScore-O increases from 0.902 to 0.938. The latter 28.7% reduction in CD-O shows the additional value of supplying SAM3D active voxel coordinates even when TRELLIS.2 already receives the completed image.

Inpainting likewise improves both metrics with either coordinate source. For direct TRELLIS.2, it reduces CD-O from 0.201 to 0.115 and raises FScore-O from 0.572 to 0.902; with HLI, the corresponding changes are 0.097 to 0.082 and 0.881 to 0.938. Segmentation + HLI and Inpainting + TRELLIS.2 trade off the two metrics, whereas their combination performs best on both. These results support complementary contributions from transferred structural support and completed image conditioning to object geometry on 3D-FRONT.

A.6 Human Preference Study of Input Faithfulness and Detail Fineness

We design a paired preference study with 20 participants to assess faithfulness to the input and detail fineness separately. The study uses two sets of objects: 100 randomly sampled from 3D-FRONT and 100 randomly sampled from objects reconstructed in AI-generated scene images. Each object is reconstructed by SAM3D and by FOMO, using the SAM3D+TRELLIS.2 configuration. All participants assess the same objects, yielding 2,000 judgments per dataset for each evaluation criterion.

For each comparison, participants see the source scene image with the target object indicated, an object crop, and matched renderings of the two reconstructions. The candidates use the same viewing

conditions, lighting, background, and image resolution, with zooming available for close inspection. Method names are hidden, the left–right order of candidates is randomized, and object order is randomized for each participant. Participants select one of the two candidates independently for each criterion. *Faithfulness to input* asks: "Which reconstruction more faithfully matches the target object in the input image?" Participants assess agreement in overall shape and proportions, the presence and arrangement of visible parts, colors, and surface appearance. Missing or invented parts and altered arrangements reduce faithfulness, even when they are rendered sharply. *Detail fineness* asks: "Which reconstruction has finer, more clearly resolved geometric and surface features?" Participants assess the resolution and clarity of small features, including edges, surface relief, and texture. For example, a reconstruction with crisp buttons can score well for detail fineness but poorly for faithfulness if their number or arrangement differs from the input.

Tab. 9 organizes the preference rates by object set and evaluation criterion. For each dataset and criterion, a method's preference rate is the percentage of the 2,000 judgments that select it; the two methods' rates therefore sum to 100%. Faithfulness judgments are grounded in the object's visible properties in the input, while the 3D-FRONT geometry metrics provide a separate comparison against complete GT objects.

Table 9: Human preference for faithfulness to the input and detail fineness. Twenty participants compare SAM3D with FOMO on 100 randomly sampled objects from each data source. Faithfulness measures agreement with the observed object; fineness measures the resolution and clarity of small features. Entries are preference rates (%) for each criterion; higher is better.

	3D-FRONT		AI-generated images	
Method	Faithfulness to input↑	Detail fineness↑	Faithfulness to input↑	Detail fineness↑
SAM3D	37.4	21.2	26.7	18.9
FOMO	62.6	78.8	73.3	81.1

A.7 Scene-Scale Sensitivity

This experiment tests whether calibration recovers a consistent relative layout when the same scene is presented at different metric scales. We evaluate MIDI and SAM3D on a fixed subset of 100 3D-FRONT scenes at $s \in \{0.5, 1, 2, 4, 8, 16, 100\}$. At each scale, GT geometry is transformed by $(x, y, z) \mapsto (sx, sy, sz)$ about the fixed camera origin, with metric depth and pointmaps scaled consistently. RGB images, segmentation masks, and camera intrinsics remain unchanged. We rerun generation for each generator and scale, then apply None, ICP, SAM3D-post, and Ours to the same generated meshes using the corresponding scaled observations. MIDI's native generator receives RGB images and masks, while SAM3D additionally receives the pointmap at the current scale. The comparison therefore tests calibration with both a generator whose visual inputs remain fixed and one whose geometric conditioning changes with scale.

Tab. 10 and Tab. 11 report all 56 generator–scale–calibrator configurations. Every entry averages the same 100 scenes with equal weight; results at different scales are kept separate. Evaluation uses 20,480 surface samples per object and exact nearest-neighbor distances. CD-S sums the two directional mean Euclidean distances in metres, FScore-S uses a fixed $0.1\,\mathrm{m}$ distance threshold, and IoU-B averages matched-object bounding-box overlap within each scene. FScore-S and IoU-B lie in $[0, 1]$.

Ours achieves the lowest CD-S and highest IoU-B at every tested scale for both generators. For MIDI, its IoU-B stays between 0.62719 and 0.63085, compared with at most 0.45056 for ICP and 0.10257 for SAM3D-post. For SAM3D, Ours maintains IoU-B between 0.60465 and 0.60565, while the uncalibrated outputs remain between 0.42361 and 0.42465. The stable overlap in Fig. 4 shows that our visible-boundary and depth constraints preserve relative layout over the tested scale range, with improved accuracy over the three alternatives. This behavior is consistent with the calibration design: projected boundaries constrain the image-space footprint, while the rescaled depth anchors the object's metric scale and translation.

Table 10: Metric-scale calibration of MIDI across scene scales on 100 3D-FRONT scenes. Generation and calibration are rerun at each scale; every entry is an equally weighted mean over the same scenes. Ours has the lowest CD-S and highest FScore-S and IoU-B throughout the tested range. CD-S is in metres; FScore-S uses a fixed $0.1\,\mathrm{m}$ threshold, and FScore-S/IoU-B lie in $[0, 1]$. Bold marks the best calibrator for each scale and metric before rounding. Values have five decimal places, with scientific notation for smaller nonzero values.

Scale	Calibrator	CD-S↓	FScore-S↑	IoU-B↑
0.5×	None	2.50592	0.00325	0.00035
0.5×	ICP	0.13973	0.75095	0.08849
0.5×	SAM3D-post	1.38851	0.19342	0.10257
0.5×	Ours	**0.03037**	**0.98594**	**0.63084**
1×	None	5.67170	0.00000	0.00000
1×	ICP	0.09074	0.86938	0.45056
1×	SAM3D-post	2.98408	0.15231	0.10018
1×	Ours	**0.06081**	**0.92920**	**0.63084**
2×	None	12.18268	0.00000	0.00000
2×	ICP	0.35342	0.39561	0.15607
2×	SAM3D-post	6.20685	0.09655	0.08586
2×	Ours	**0.12142**	**0.82586**	**0.63084**
4×	None	25.30048	0.00000	0.00000
4×	ICP	1.07711	0.13914	0.02516
4×	SAM3D-post	12.35761	0.10313	0.09622
4×	Ours	**0.25947**	**0.65953**	**0.62726**
8×	None	51.60184	0.00000	0.00000
8×	ICP	2.58396	0.03950	0.00398
8×	SAM3D-post	25.84097	0.03909	0.08104
8×	Ours	**0.48620**	**0.45028**	**0.63085**
16×	None	104.22597	0.00000	0.00000
16×	ICP	5.59717	0.00836	0.00062
16×	SAM3D-post	53.01192	0.02013	0.08962
16×	Ours	**0.97090**	**0.22671**	**0.63084**
100×	None	656.80977	0.00000	0.00000
100×	ICP	37.22013	0.00013	1.8492×10^{-6}
100×	SAM3D-post	353.60396	0.00091	0.08160
100×	Ours	**6.48640**	**0.00659**	**0.62719**

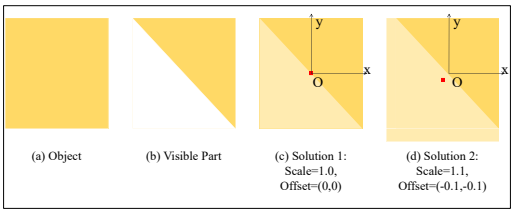

Figure 7: Visible evidence and scale–translation ambiguity. The visible part of the object in (b) is compatible with both candidates in (c) and (d). Solution 1 matches the observed projected bounds, whereas Solution 2 enlarges and shifts the object beyond them. Our boundary and depth constraints favor the candidate supported by the observed extent and mean depth; this preference does not establish uniqueness for every occluded object.

A.8 Visible Evidence and Scale–Translation Ambiguity

Preference for visible evidence. Partial visibility can leave several scale–translation configurations compatible with the visible image region. In Fig. 7, both solutions contain the observed portion of the object, but Solution 2 additionally expands and shifts the object beyond the observed bounds. Our calibrator matches projected extent to the visible-region bounding box and anchors mean depth

Table 11: Metric-scale calibration of SAM3D across scene scales on 100 3D-FRONT scenes. Generation uses the rescaled pointmap, and each calibrator is rerun with the corresponding observations; entries average the same scenes equally. Ours has the lowest CD-S and highest IoU-B at every scale; ICP has the highest FScore-S at $100\times$. CD-S is in metres; FScore-S uses a fixed $0.1\,\mathrm{m}$ threshold, and FScore-S/IoU-B lie in $[0, 1]$. Bold marks the best calibrator for each scale and metric before rounding. Values have five decimal places, with scientific notation for smaller nonzero values.

Scale	Calibrator	CD-S↓	FScore-S↑	IoU-B↑
$0.5\times$	None	0.05314	0.94939	0.42411
$0.5\times$	ICP	0.04756	0.96670	0.40902
$0.5\times$	SAM3D-post	0.05372	0.94553	0.44283
$0.5\times$	Ours	**0.03276**	**0.98057**	**0.60562**
$1\times$	None	0.10625	0.85779	0.42411
$1\times$	ICP	0.09531	0.85685	0.40786
$1\times$	SAM3D-post	0.11304	0.82913	0.41910
$1\times$	Ours	**0.06545**	**0.93353**	**0.60563**
$2\times$	None	0.21238	0.67272	0.42411
$2\times$	ICP	0.19138	0.69201	0.40598
$2\times$	SAM3D-post	0.23300	0.64748	0.41593
$2\times$	Ours	**0.13092**	**0.81030**	**0.60559**
$4\times$	None	0.42531	0.45680	0.42465
$4\times$	ICP	0.38286	0.50759	0.40534
$4\times$	SAM3D-post	0.47615	0.43272	0.40983
$4\times$	Ours	**0.26195**	**0.61935**	**0.60467**
$8\times$	None	0.85041	0.27037	0.42410
$8\times$	ICP	0.76530	0.31861	0.40525
$8\times$	SAM3D-post	0.95933	0.24658	0.40085
$8\times$	Ours	**0.52438**	**0.40475**	**0.60562**
$16\times$	None	1.69958	0.12919	0.42465
$16\times$	ICP	1.53431	0.16148	0.40237
$16\times$	SAM3D-post	1.88706	0.11818	0.41059
$16\times$	Ours	**1.04729**	**0.19992**	**0.60465**
$100\times$	None	10.60211	0.00332	0.42361
$100\times$	ICP	9.56921	**0.00510**	0.40344
$100\times$	SAM3D-post	11.84877	0.00314	0.40902
$100\times$	Ours	**6.56301**	0.00451	**0.60565**

Table 12: Distribution of object occlusion among objects in 3D-FRONT.

Occlusion bin	3D-FRONT Share
0%	61.60%
$(0, 20)\%$	21.59%
$[20, 40)\%$	7.96%
$[40, 60)\%$	2.64%
$[60, 80)\%$	1.83%
$[80, 95)\%$	2.10%
$[95, 100]\%$	2.28%
Total	100.00%

to the observation, favoring Solution 1 in this example. This explains the choice of global visibility constraints: they tie the estimate to observable extent and depth without relying on ambiguous local correspondences.

Identifiability and approximate constraints. A unique update for s, ΔX, ΔY, and ΔZ requires a rank-four constraint matrix. Three informative boundary equations together with the depth equa-

tion can provide four independent constraints in a nondegenerate case, while the fourth boundary equation makes the fit overdetermined. The equation count alone does not guarantee this independence, and a visible mask edge need not coincide with the full object's projected boundary. Least squares fits the available evidence when boundaries are approximate; severe occlusion, incorrect masks, or insufficient depth support can still make the estimate unstable or biased. The favorable comparisons in the evaluated scenes establish the practical usefulness of these constraints, not uniqueness for every possible partial observation.

A.9 Limitations

The modular design inherits the scope of its inputs and pretrained components. Holding orientation fixed improves the separation of pose components but cannot correct a wrong upstream rotation. Heterogeneous latent infusion also requires the refiner to be SLAT-based; fine detail cannot compensate for an incorrect completed structure. However, as a training-free framework, improving base models with better completed structure or rotation is not the focus of this work.

The reported means do not include confidence intervals or significance tests. Future work can incorporate input uncertainty, improve calibration under severe occlusion, and evaluate physical stability and simulation readiness in addition to the object fidelity and layout accuracy studied here.

A.10 Additional Related Work

Scene reconstruction and completion. Existing systems couple geometry generation and layout estimation in different ways and use different forms of geometric input. SceneGen (Meng et al., 2025) predicts a scene in one feed-forward pass, while DepR (Zhao et al., 2025) applies depth-guided diffusion to individual instances. RecGen jointly reconstructs object or part geometry and pose from one or more RGB-D observations (Zadaianchuk et al., 2026). FIRE3D uses a learned compact representation and batched reconstruction to recover instances, poses, and textured assets from posed RGB-D observations; its RGB image or video frontend first estimates geometry and cameras (Xia et al., 2026). Our evaluation separates upstream mesh generation from metric-scale calibration so that the same alignment mechanism can be compared across the evaluated generation priors.

Simulation-oriented reconstruction. Simulation places additional requirements on generated scenes, including stable support relationships and reduced interpenetration. REST3D combines a scene-tree representation with physics-constrained refinement to address floating objects, penetration, and instability (Ma et al., 2026). SimuScene uses diagnostic simulation to guide shape and layout corrections, including gravity-axis stretching and amodal shape resampling (Lee et al., 2026). These methods complement FOMO's focus on complete object structure, detailed geometry and appearance, and metric-scale layout. Our geometric metrics assess asset and placement accuracy; evaluating physical stability would require additional simulation-based criteria.

Structured representations and editing. Structured generative representations provide the interface used by heterogeneous latent infusion. TRELLIS associates active voxel coordinates with local features and decodes them into meshes, radiance fields, or 3D Gaussians (Xiang et al., 2025b); TRELLIS.2 uses O-Voxel to represent geometry and surface attributes (Xiang et al., 2025a). DVD replaces sparse-structure generation with discrete occupancy diffusion and supports voxel editing (Xiang et al., 2026). NANO3D (Ye et al., 2025) integrates FlowEdit (Kulikov et al., 2025) with TRELLIS, using front-view renderings and voxel or structured-latent merging for localized edits. HIVE-3D trains a voxel super-resolution adapter to refine scene components recursively (Zang et al., 2026). FOMO exploits the structure–feature decomposition to combine an occlusion-aware source model's active coordinates with a complementary generator's feature synthesis, without requiring joint training or cross-model feature alignment.

Geometric conditioning and multi-view alignment. Partial observations can guide reconstruction through attention, explicit geometry, or jointly learned representations. Amodal3R uses mask-weighted cross-attention and occlusion-aware conditioning (Wu et al., 2025), while Axolotl3D learns from images, visibility masks, cameras, and partial point clouds to support single- and multi-view completion (Hu & Shugrina, 2026). Mix3R jointly predicts sparse structure, pointmaps, and

cameras with a trained Mixture-of-Transformers architecture, then introduces an overlap-based attention bias during textured generation (Lin et al., 2026). Its second-stage bias needs no additional training, although the joint reconstruction architecture is learned. Our framework orchestrates complementary pretrained capabilities: active coordinates supply occlusion-aware structure, and the inpainting VLM combines complete shape with observed appearance to condition local feature synthesis.

Pose estimation for generated geometry. Pose estimators differ in the observations they require and the variables they optimize. Gen6D estimates model-free pose from reference observations (Liu et al., 2022b); MegaPose, SAM-6D, and FoundationPose use object models, learned matching, rendering, or iterative refinement under their respective input assumptions (Labbé et al., 2023; Lin et al., 2024; Wen et al., 2024). For generated geometry, Zhang et al. (2026) initialize alignment with guarded Sim(3) registration and then fix scale during rigid ICP refinement. SnapPhysics combines global alignment and object-wise Sim(3) optimization, using the resulting metric geometry for physical-property reasoning (Kang et al., 2026). FOMO retains the upstream orientation and fits scale and translation to projected boundaries and mean visible depth through a small linear system. This design motivates the comparison with None, Open3D $\mathrm{Sim}(3)$ ICP with scale estimation enabled (Zhou et al., 2018), and SAM3D-post in Tab. 1, as well as the occlusion and scene-scale analyses that test the use of visible geometry for metric calibration.

B Additional Figures and Tables

These comparisons connect the two mechanisms in FOMO to their visible effects. In each scene matrix, the input image is shown separately above the generated renderings. Fig. 8 and Fig. 9 contrast complete yet coarse objects from strong occlusion-aware baselines with the local geometry and appearance added by heterogeneous latent infusion. Fig. 10–12 examine why neither structural completion nor detail-rich generation alone supplies both capabilities, and Fig. 13 isolates the question of scene placement.

The scene matrices in Fig. 14–32 extend the comparison to CUPID, RecGen, and SAM3D with two detail-rich refiners. Within each row, the source generator is fixed; NONE retains its mesh, while TRELLIS and TRELLIS.2 denote the two refinement settings. The conditioning matrices in Fig. 33–51 compare coarse-mesh renderings with segmentation crops. They pair with Fig. 14–32 in the same scene order: the TRELLIS and TRELLIS.2 columns of each standard RecGen/SAM3D row provide the inpainter-conditioned counterparts. Compare the same source generator and refiner across each pair to assess how image conditioning affects object appearance. These AI-generated scenes provide qualitative evidence about shape and appearance; metric-scale accuracy is evaluated against ground truth in Tab. 1.



Figure 8: Heterogeneous latent infusion improves local detail over SAM3D, including plush surfaces, the toy truck's grille, and audio mixer controls. Input images appear above; numbers give object counts and boxes highlight details. Fig. 9 provides complementary geometry comparisons without textures.



Figure 9: Geometry comparisons separate structural completion from local detail. The upper SAM3D–FOMO pairs expose finer cookie surfaces and individual mixer controls in FOMO; the lower rows compare with PartCrafter, MIDI, and SAM3D on scenes containing small and occluded objects. Compare the toy truck, plush toys, and figure accessories to assess both complete shape and local structure. Solid shading removes texture as an explanation for these differences, complementing the appearance comparison in Fig. 8. Numbers give object counts.



Figure 10: Strong occlusion-aware structure does not by itself provide fine geometric detail. SAM3D and the detail-rich generator TRELLIS.2 both recover the backpack and armor silhouettes, but TRELLIS.2 resolves more of the pockets, straps, and armor panels. Together with the partial-observation failures in Fig. 12, this comparison motivates drawing structure and local detail from complementary pretrained models. The 2M+ label refers to the 2.7 million object meshes used for SAM3D pretraining (SAM 3D Team et al., 2025, Sec. 3.1.1); 800K refers to the approximately 800,000 assets used to train the TRELLIS.2 generative model (Xiang et al., 2025a, Sec. 4). These counts describe different training pipelines, rather than a controlled comparison of data efficiency.

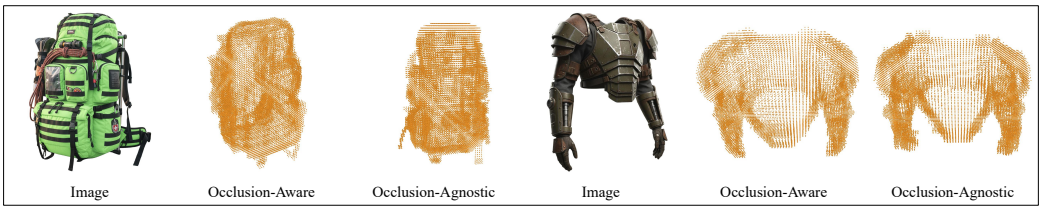

Figure 11: Similar coarse structure can support different levels of decoded detail. The active voxel coordinates from SAM3D and TRELLIS.2 capture comparable backpack and armor extents, whereas the local surface differences in Fig. 10 are not resolved by these structures alone. This illustrates the structure–feature decomposition used by heterogeneous latent infusion: transfer active coordinates from the occlusion-aware generator, then synthesize local features with the detail-rich generator.

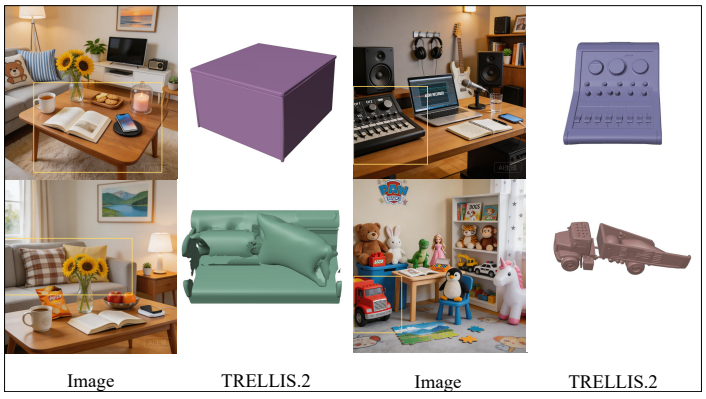

Figure 12: Detail-rich generation alone is sensitive to partial observations. Direct TRELLIS.2 generation collapses the occluded coffee table into a box-like shape and distorts the sofa; the mixer and toy truck also lose structure or detail visible in the corresponding FOMO outputs in Fig. 8 and Fig. 9. Each pair shows the input scene and the target asset. The comparison motivates supplying completed active voxel coordinates and a complete object image to the refiner.

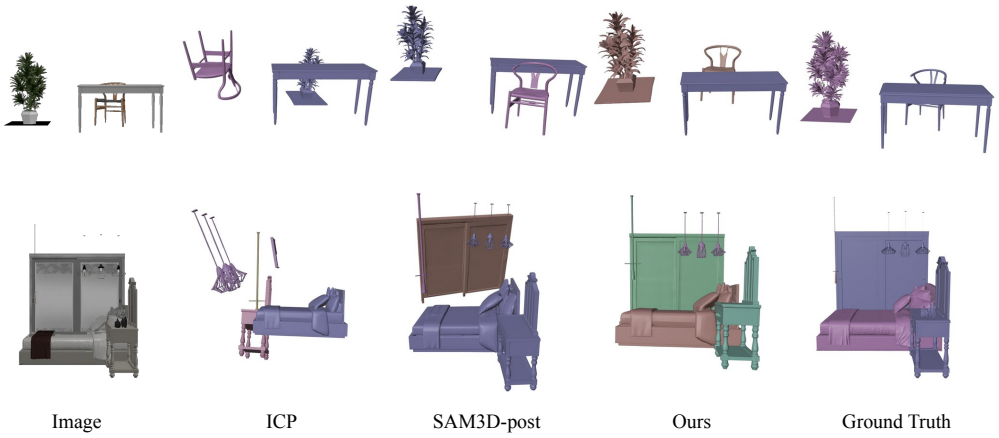

Figure 13: Metric-scale calibration improves relative object placement on 3D-FRONT. Compare the chair and table in the upper example and the hanging lights and bedroom furniture in the lower example against the ground-truth column. Our calibrator recovers their relative placement more closely than ICP and SAM3D-post by constraining projected extent and mean visible depth. From left to right, columns show Image, ICP, SAM3D-post, Ours, and Ground Truth; Tab. 1 gives the broader calibration comparison.

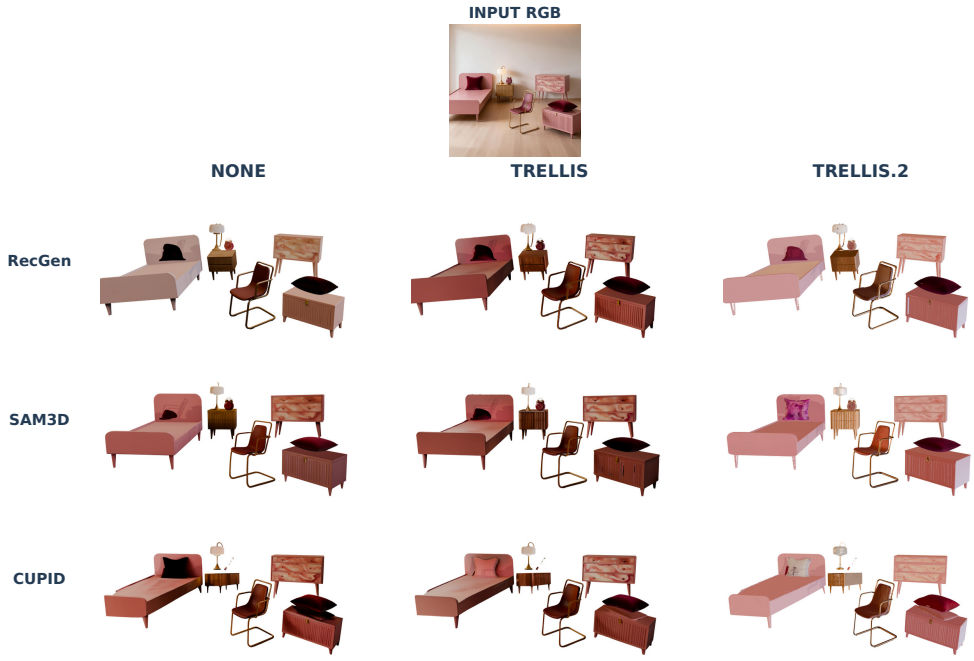

Figure 14: Heterogeneous latent infusion on bedroom example 1. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. The bed and chair retain their overall forms across refiners, while upholstery colors and cabinet surfaces change. Compare these appearance changes with the input to assess the contribution of local feature synthesis.



Figure 15: Heterogeneous latent infusion on bedroom example 2. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. The curved headboard, desk, and chair provide recognizable source geometry. Their wood patterns and surface relief make the changes introduced by the two detail-rich refiners visible beyond the overall silhouettes.



Figure 16: Heterogeneous latent infusion on dining room example 1. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. The table and thin chair frames remain recognizable across the three columns. Cabinet fronts, chair seats, and small tabletop objects expose the distinction between inherited structure and synthesized appearance.

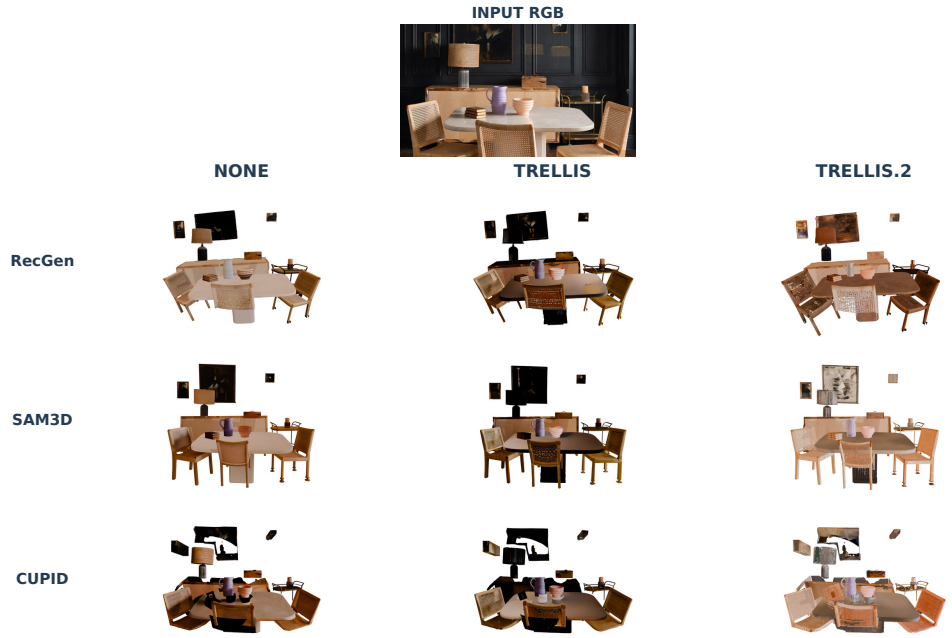

Figure 17: Heterogeneous latent infusion on dining room example 2. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. The chairs surrounding the table test completion under mutual occlusion, while the chair surfaces and wall decorations expose changes in local appearance. Compare within each source-model row.

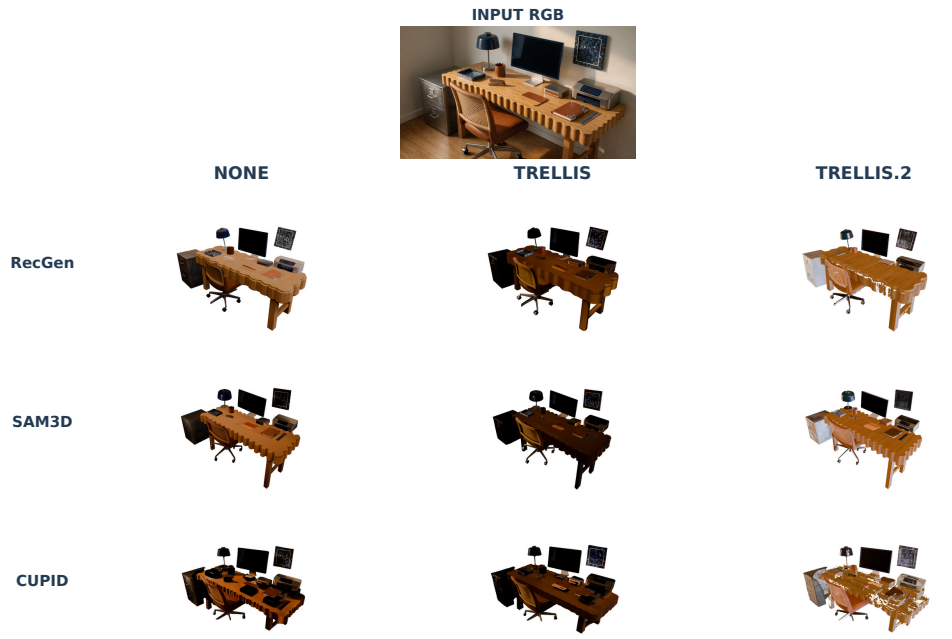

Figure 18: Heterogeneous latent infusion on home office example 1. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. The desk, swivel chair, and monitors remain identifiable after refinement. The desk front and small objects on its surface allow local detail to be compared without relying on the large furniture silhouettes alone.



Figure 19: Heterogeneous latent infusion on home office example 2. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. The chair base and desk legs probe retention of thin structures; the drawer fronts, desktop, and framed image expose appearance differences between the two refiners and the source mesh.



Figure 20: Heterogeneous latent infusion on kitchen example 1. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. Cabinet fronts and the oven provide a clear comparison of local surface structure, alongside the larger refrigerator and counter silhouettes. The refined columns also distinguish the appearance of the small countertop objects.

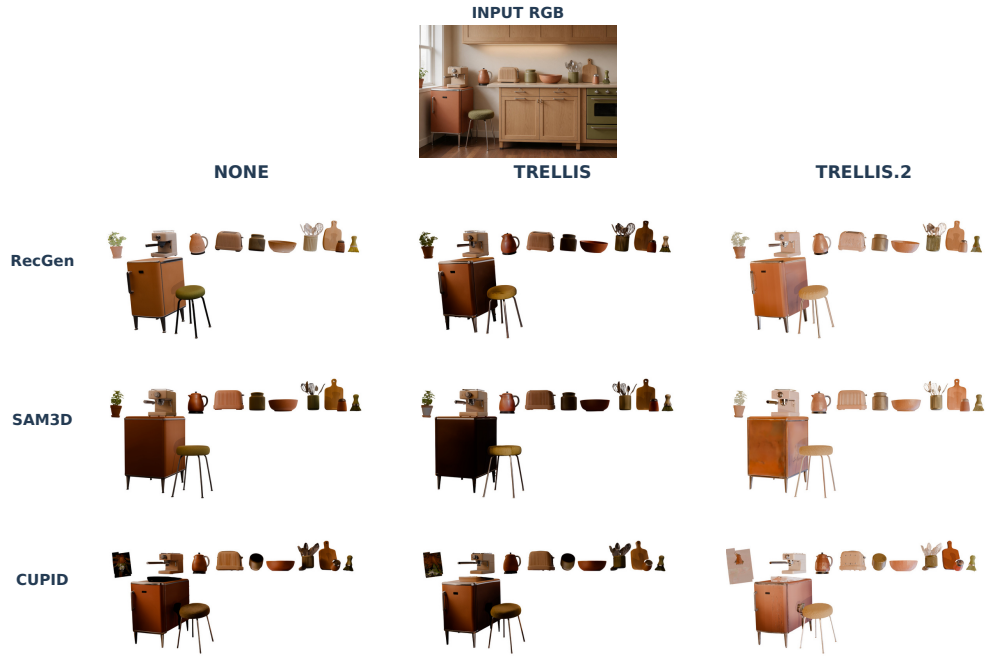

Figure 21: Heterogeneous latent infusion on kitchen example 2. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. This scene separates small kitchenware from larger furniture. Compare the vessels, utensils, and appliances across columns to assess whether refinement adds recognizable local features while retaining individual object shapes.

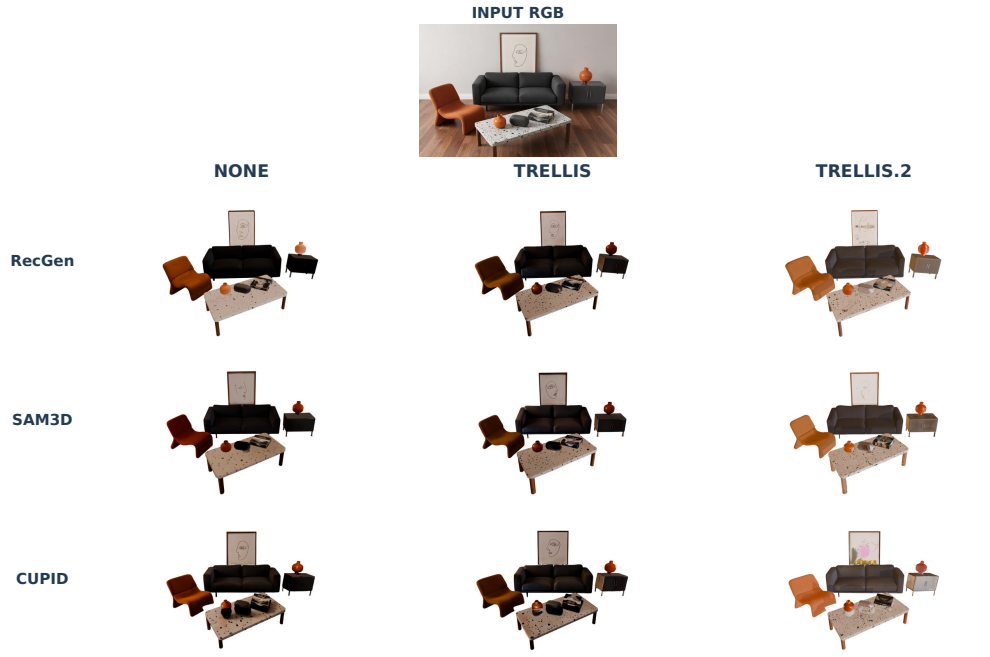

Figure 22: Heterogeneous latent infusion on living room example 1. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. The sofa and armchair retain recognizable forms, while the coffee-table surface, cushions, and framed artwork change with the refiner. These regions distinguish appearance synthesis from coarse structural completion.



Figure 23: Heterogeneous latent infusion on living room example 2. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. The plants, upholstery, and small coffee-table objects place different demands on local feature synthesis. Compare their surface detail across columns alongside the overall seating arrangement inherited from each source model.

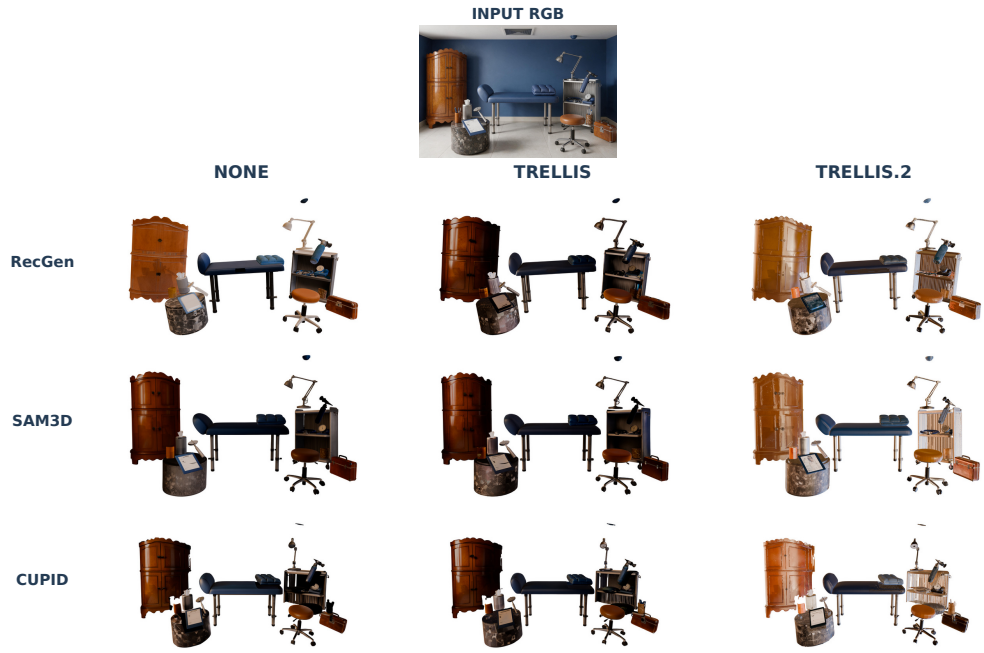

Figure 24: Heterogeneous latent infusion on medical examination room example 1. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. The examination bed and articulated lamps expose the complete source geometry; the storage cabinet, wheeled equipment, and stool reveal changes in surface detail and material appearance after refinement.

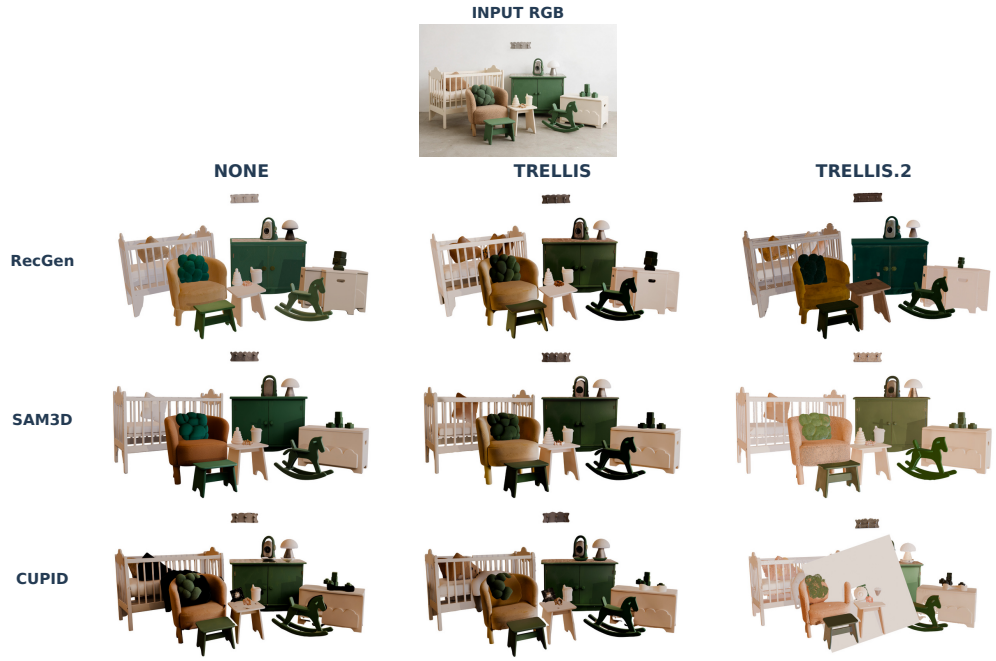

Figure 25: Heterogeneous latent infusion on nursery example 1. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. The crib rails and rocking horse test preservation of thin, completed structures, while the chair and cabinet expose appearance changes. The CUPID–TRELLIS.2 panel also contains a large planar artifact, showing that refinement can introduce visible failures.

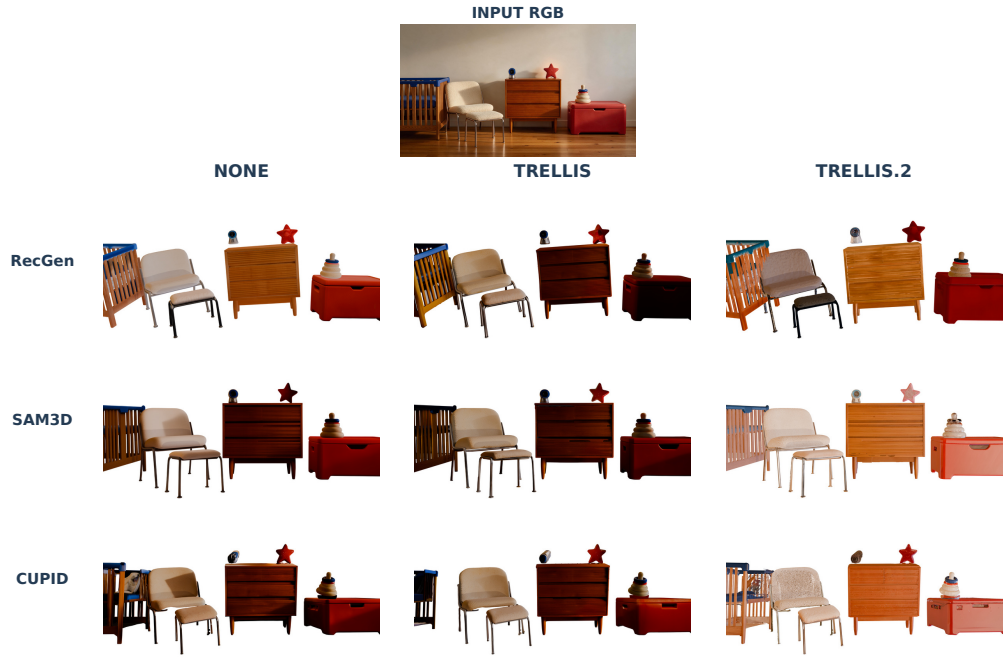

Figure 26: Heterogeneous latent infusion on nursery example 2. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. The crib, chair, dresser, and toy chest retain distinct identities across the configurations. Wood grain, upholstery, and the small objects on the dresser reveal which local appearance cues each refiner supplies.

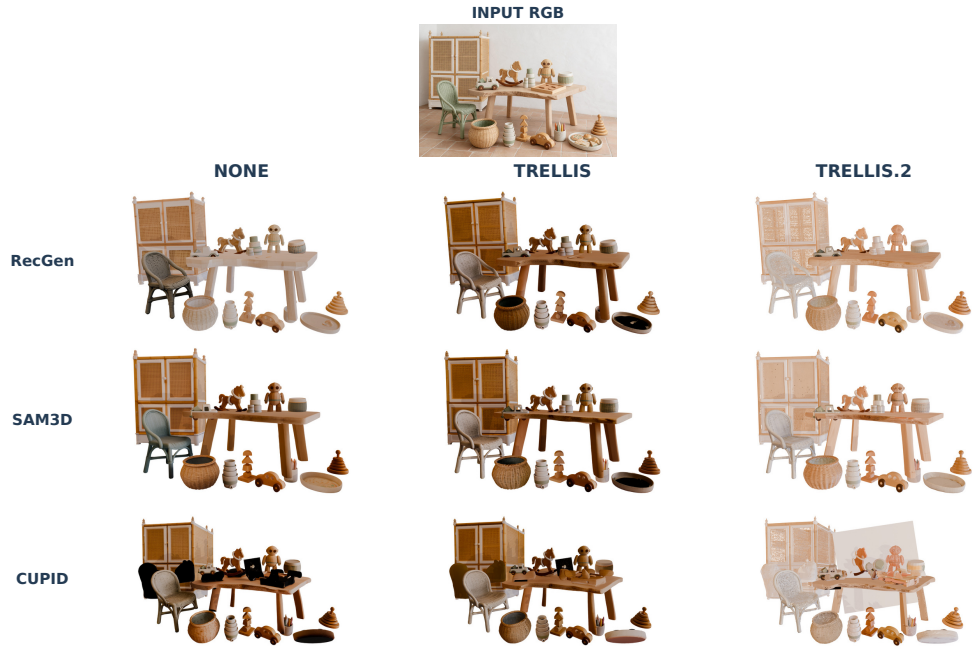

Figure 27: Heterogeneous latent infusion on playroom example 1. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. The toys, baskets, and stacked rings test local detail on small objects alongside a complete table-and-chair structure. The cabinet panels provide a larger surface on which to compare the refiners' appearance synthesis.

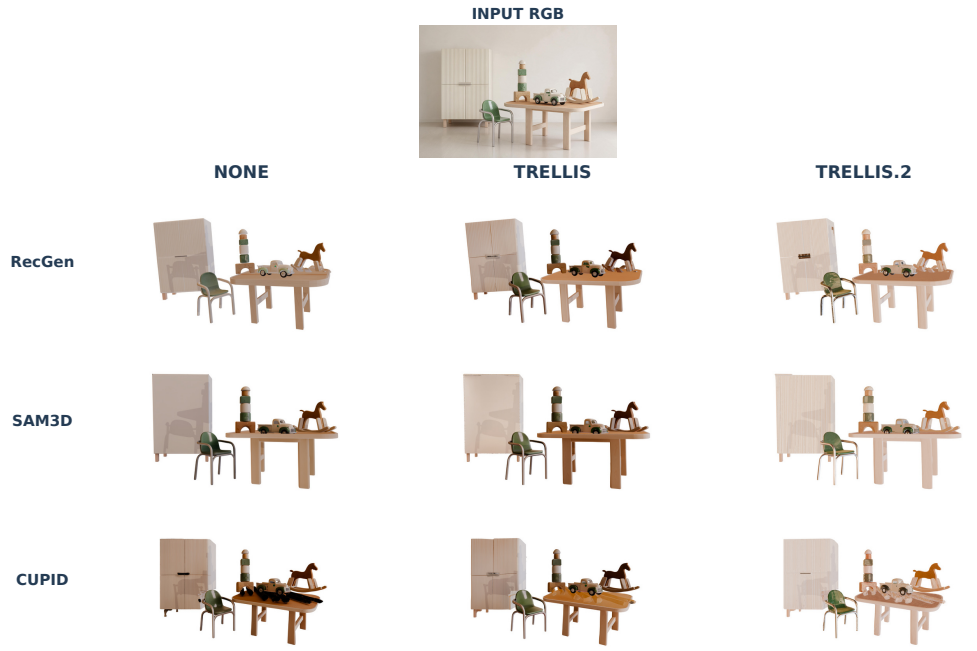

Figure 28: Heterogeneous latent infusion on playroom example 2. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. The toy horse, miniature vehicles, and stacked blocks provide small-scale shape cues beyond the cabinet and table silhouettes. Compare their individual forms and colors across refinement settings.

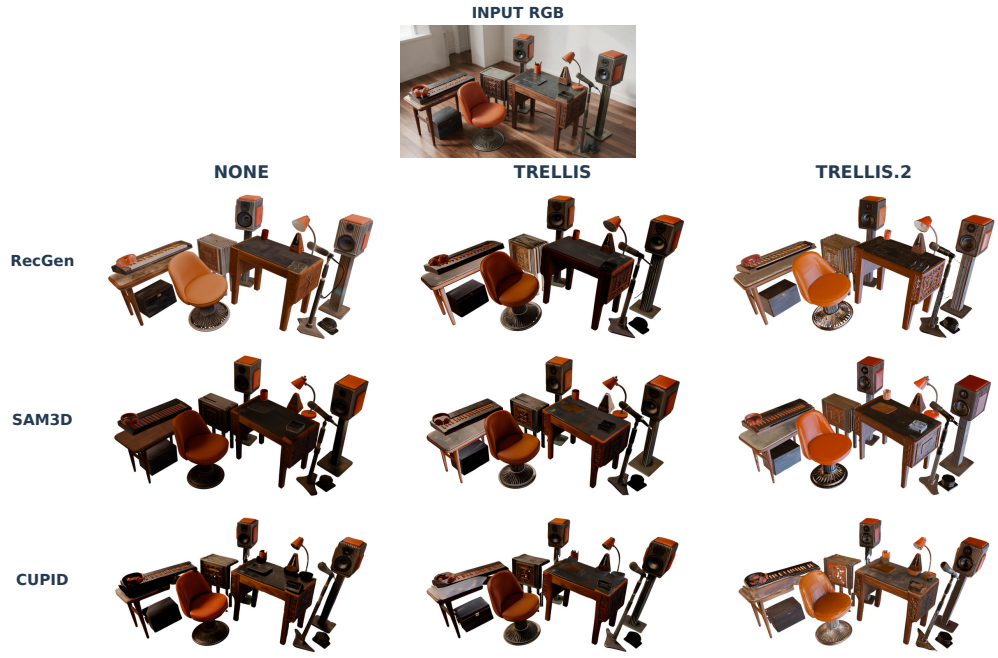

Figure 29: Heterogeneous latent infusion on recording studio example 1. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. The keyboard, speaker drivers, and desk equipment expose local structure, while the chair and speaker stands retain the overall scene organization. Compare these details across columns for each source generator.



Figure 30: Heterogeneous latent infusion on recording studio example 2. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. The mixing surface, speakers, and chair combine repeated controls with smooth material surfaces. Their appearance changes under refinement can be assessed separately from the source-model differences in furniture shape.



Figure 31: Heterogeneous latent infusion on workshop example 1. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. The small tools and containers on the workbench test detailed asset generation within a cluttered scene. Compare their separation and surface appearance across columns, together with the larger cabinet and bench structures.



Figure 32: Heterogeneous latent infusion on workshop example 2. Rows use RecGen, SAM3D, and CUPID; columns compare no refinement (NONE), TRELLIS, and TRELLIS.2. The cabinet doors, curved task lamp, stool, and plant provide complementary shape and appearance cues. Refinement changes surface detail while the major object identities remain recognizable across the source-model rows.

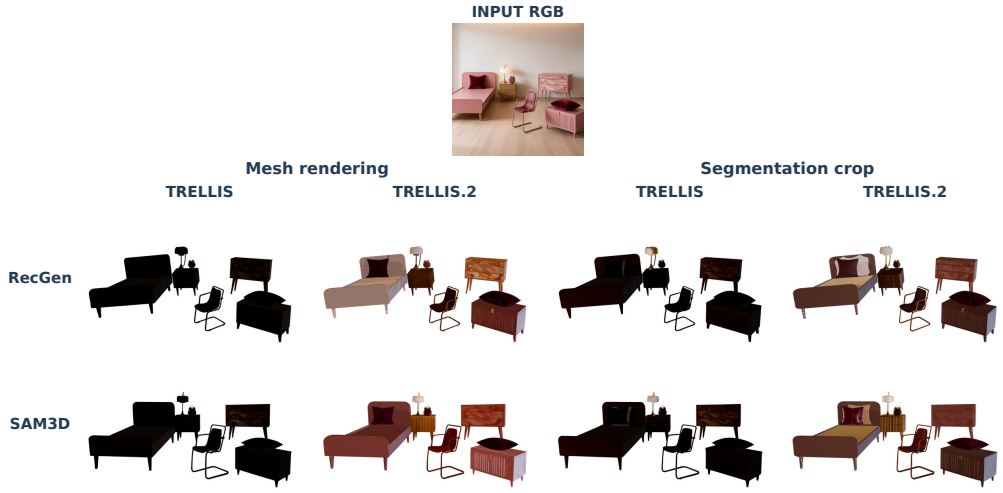

Figure 33: Direct-conditioning comparison for bedroom example 1. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. The bed, chair, and cabinet retain recognizable outlines, but the direct conditions produce markedly different upholstery and wood appearance. Complete geometry alone therefore does not ensure a suitable visual condition for feature synthesis. Both direct conditions omit identity-preserving inpainting; Fig. 14 shows the corresponding scene-generation comparison.



Figure 34: Direct-conditioning comparison for bedroom example 2. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. Compare the headboard and desk surfaces: changing from a complete rendering to a partial crop changes the generated wood and fabric appearance without changing the source generator. The comparison motivates combining complete shape with observed instance appearance. Both direct conditions omit identity-preserving inpainting; Fig. 15 shows the corresponding scene-generation comparison.

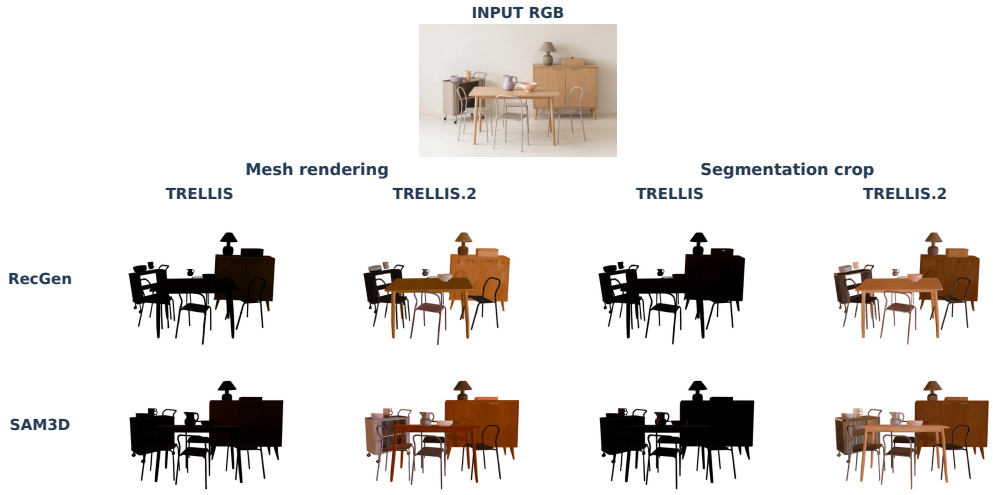

Figure 35: Direct-conditioning comparison for dining room example 1. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. The thin chair frames remain recognizable even where cabinet and seat surfaces lose visible appearance cues. This separates supplying object structure from supplying a detailed, instance-specific visual condition. Both direct conditions omit identity-preserving inpainting; Fig. 16 shows the corresponding scene-generation comparison.

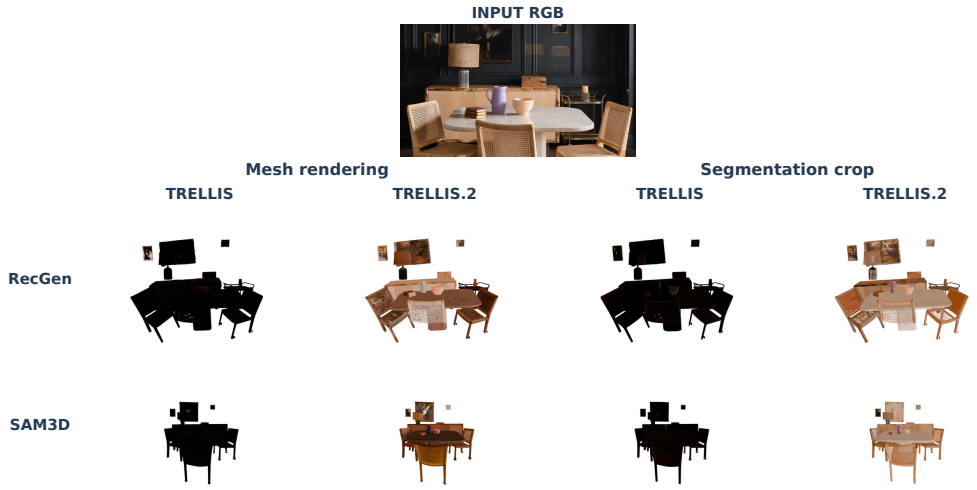

Figure 36: Direct-conditioning comparison for dining room example 2. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. Compare the dining chairs and wall decorations under the same refiner. Both direct conditions can preserve recognizable objects while producing different surface appearance, motivating a completed condition informed by the original scene. Both direct conditions omit identity-preserving inpainting; Fig. 17 shows the corresponding scene-generation comparison.

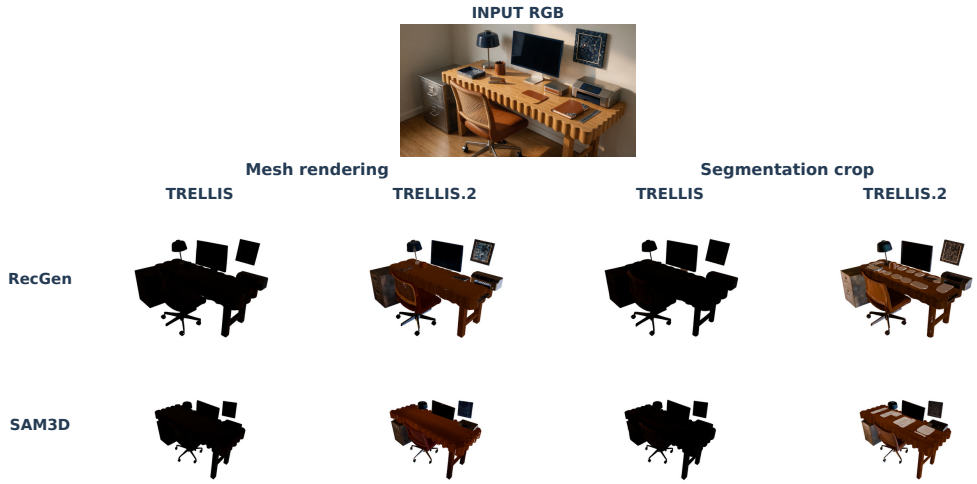

Figure 37: Direct-conditioning comparison for home office example 1. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. The desk, chair, and monitor outlines persist, but the desktop and cabinet surfaces differ substantially across visual conditions. Compare the first and third columns, or the second and fourth, to hold the refiner fixed. Both direct conditions omit identity-preserving inpainting; Fig. 18 shows the corresponding scene-generation comparison.



Figure 38: Direct-conditioning comparison for home office example 2. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. Drawer fronts, the chair seat, and the framed image expose condition-dependent appearance changes. The complete rendering and partial crop provide different evidence, which identity-preserving inpainting combines in the full pipeline. Both direct conditions omit identity-preserving inpainting; Fig. 19 shows the corresponding scene-generation comparison.

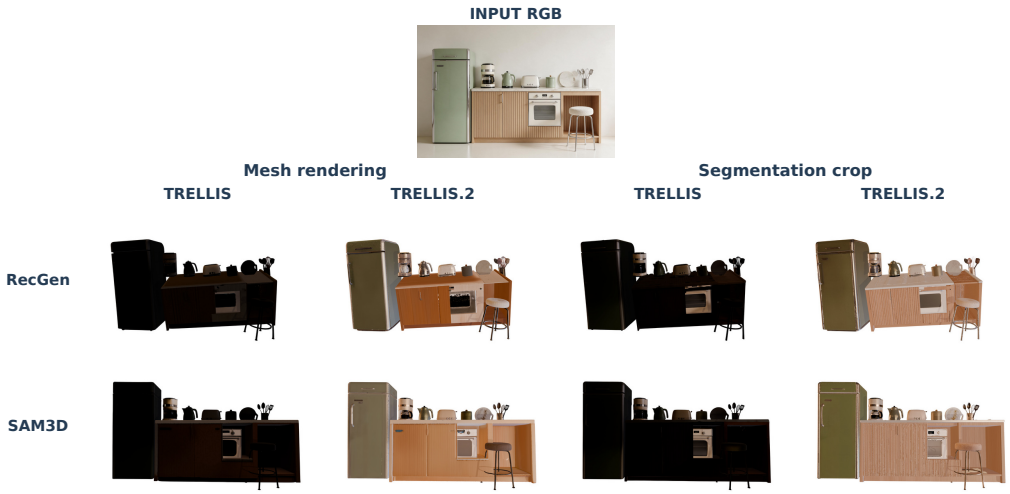

Figure 39: Direct-conditioning comparison for kitchen example 1. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. The refrigerator and counter keep their overall forms, while the cabinet finish and oven surround change with the condition. Compare these surfaces with the input to assess the appearance cues missing from each direct alternative. Both direct conditions omit identity-preserving inpainting; Fig. 20 shows the corresponding scene-generation comparison.

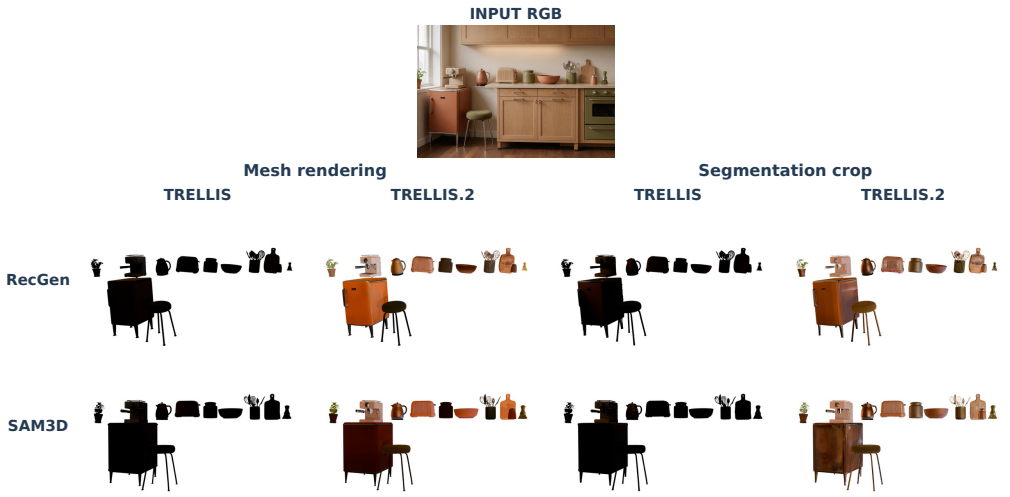

Figure 40: Direct-conditioning comparison for kitchen example 2. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. The many small vessels and appliances make the visual condition consequential beyond the large cabinet. Compare their distinct colors and surfaces under each refiner to assess whether object-specific appearance survives direct conditioning. Both direct conditions omit identity-preserving inpainting; Fig. 21 shows the corresponding scene-generation comparison.



Figure 41: Direct-conditioning comparison for living room example 1. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. The sofa and coffee table remain recognizable, but their surfaces and the framed artwork vary with the visual condition. These differences motivate preserving instance appearance as well as complete shape in the refiner input. Both direct conditions omit identity-preserving inpainting; Fig. 22 shows the corresponding scene-generation comparison.

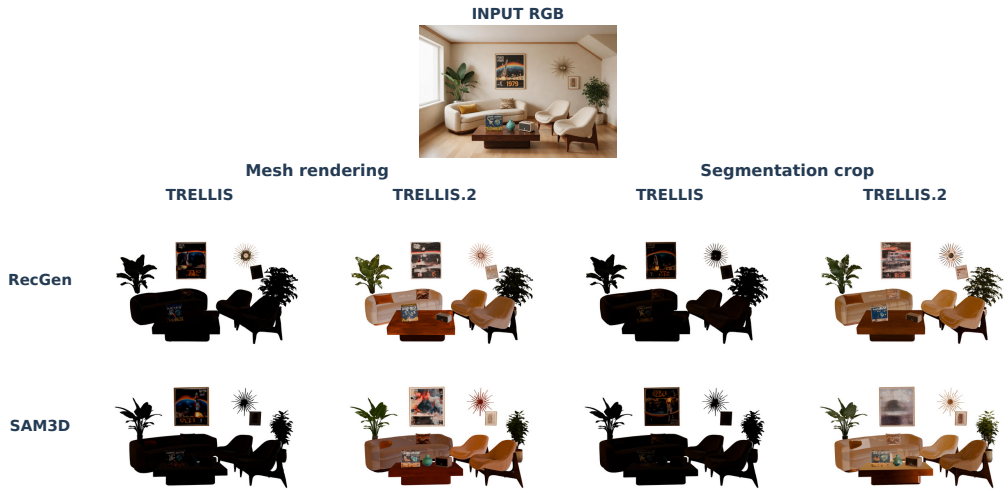

Figure 42: Direct-conditioning comparison for living room example 2. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. The plants, sofa cushions, and tabletop objects expose different appearance changes under rendering-based and crop-based conditioning. Compare the two conditions within one refiner before comparing the two refinement models. Both direct conditions omit identity-preserving inpainting; Fig. 23 shows the corresponding scene-generation comparison.

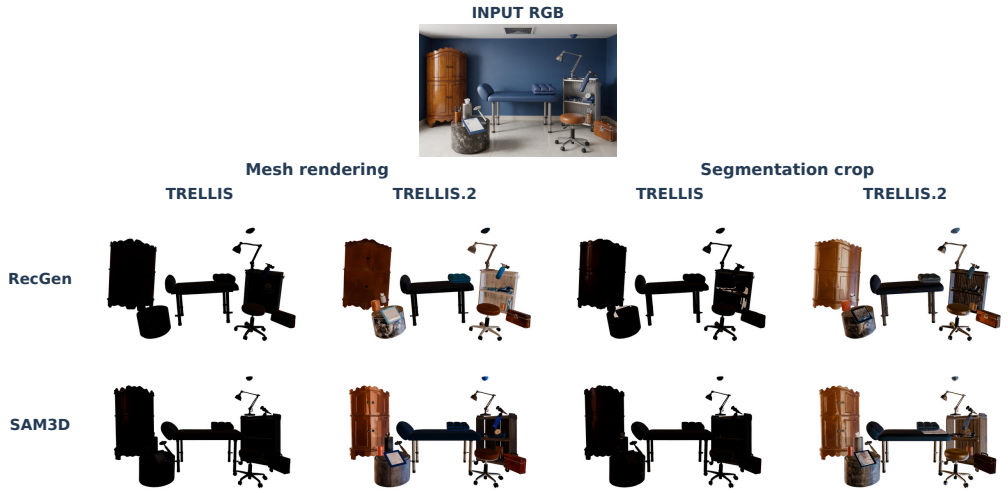

Figure 43: Direct-conditioning comparison for medical examination room example 1. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. Compare the examination bed, storage cabinet, and lamp assembly. Recognizable geometry can coexist with strongly altered surface appearance, motivating a condition that retains observed material cues on the completed object. Both direct conditions omit identity-preserving inpainting; Fig. 24 shows the corresponding scene-generation comparison.



Figure 44: Direct-conditioning comparison for nursery example 1. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. Crib rails and the rocking horse expose completed geometry, while the chair and green cabinet expose instance-specific color cues. The comparison tests whether a direct condition can preserve both types of information. Both direct conditions omit identity-preserving inpainting; Fig. 25 shows the corresponding scene-generation comparison.



Figure 45: Direct-conditioning comparison for nursery example 2. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. The chair upholstery, dresser finish, and toy-chest color vary with the visual condition. These recognizable objects show why completing shape and preserving appearance are separate requirements for the refiner input. Both direct conditions omit identity-preserving inpainting; Fig. 26 shows the corresponding scene-generation comparison.

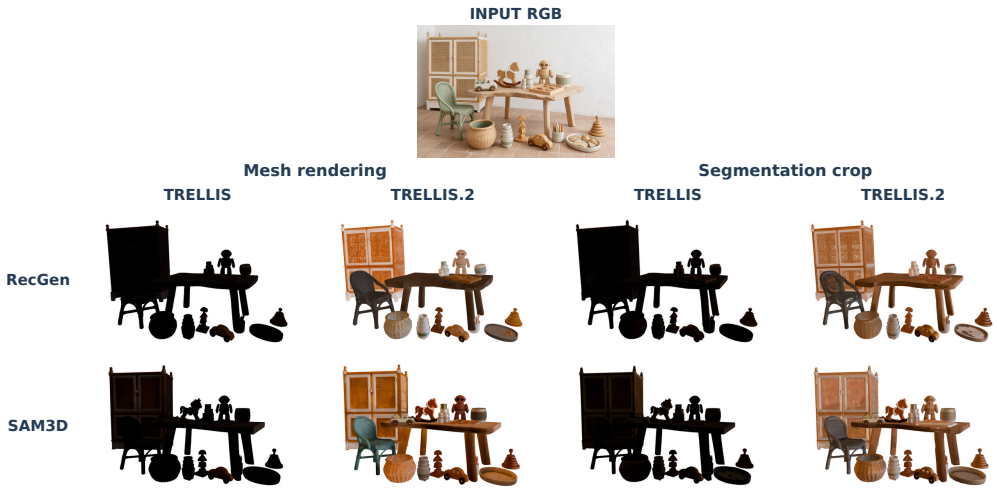

Figure 46: Direct-conditioning comparison for playroom example 1. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. The cabinet, small toys, and stacked rings retain recognizable structure while their surfaces vary across conditions. Compare the small assets as well as the furniture to assess the appearance information supplied to the refiner. Both direct conditions omit identity-preserving inpainting; Fig. 27 shows the corresponding scene-generation comparison.

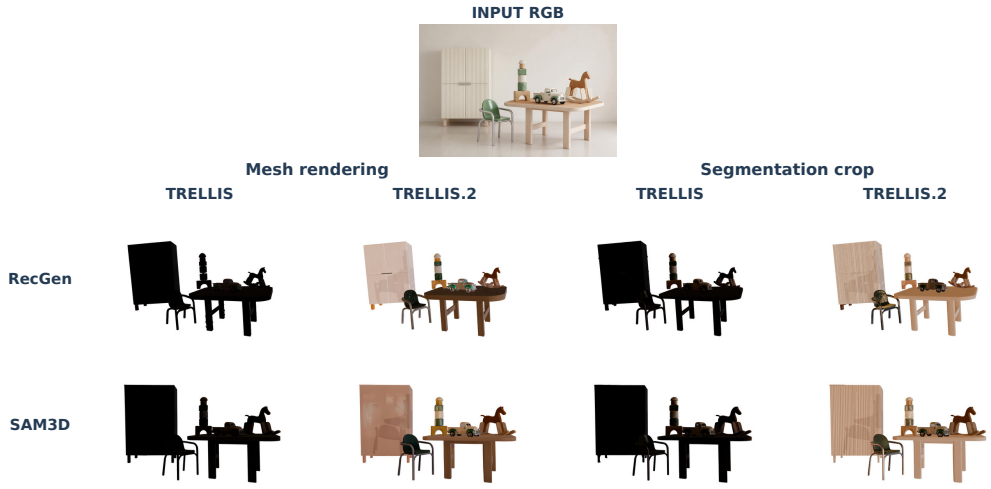

Figure 47: Direct-conditioning comparison for playroom example 2. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. The toy horse, miniature vehicles, and blocks test appearance preservation at small scale. Comparing the same refiner across the two conditions reveals the effect of a complete coarse view versus a partial observed crop. Both direct conditions omit identity-preserving inpainting; Fig. 28 shows the corresponding scene-generation comparison.

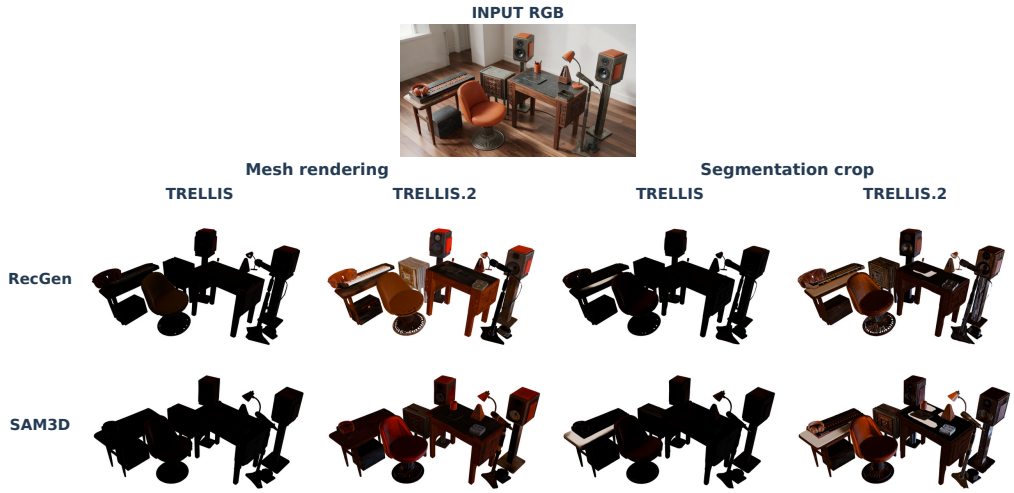

Figure 48: Direct-conditioning comparison for recording studio example 1. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. Compare the keyboard, speaker drivers, and chair surfaces. The complete rendering supplies the object forms, while the crop supplies observed appearance; their different outputs motivate combining these cues before feature synthesis. Both direct conditions omit identity-preserving inpainting; Fig. 29 shows the corresponding scene-generation comparison.

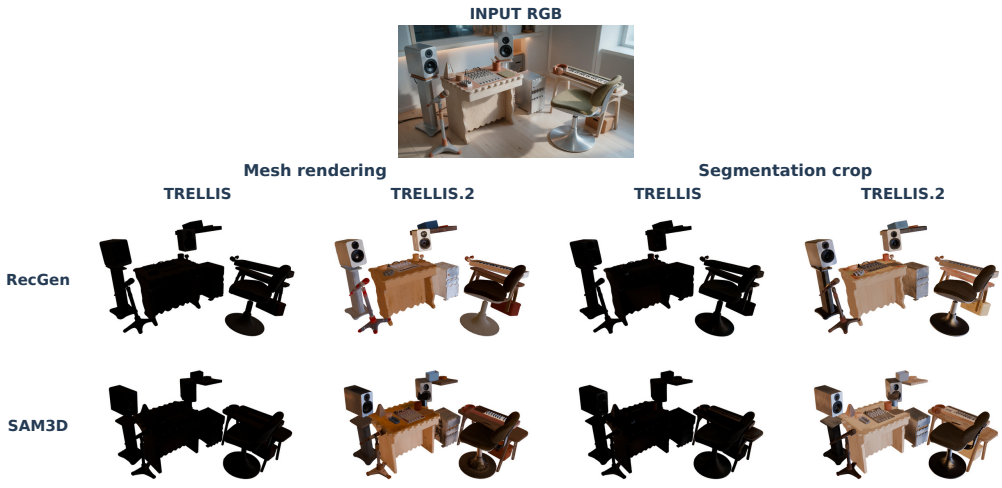

Figure 49: Direct-conditioning comparison for recording studio example 2. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. Speaker cones, the mixing surface, and chair upholstery expose local appearance differences that are not captured by the scene silhouettes. Hold the refiner fixed when comparing the two direct visual conditions. Both direct conditions omit identity-preserving inpainting; Fig. 30 shows the corresponding scene-generation comparison.

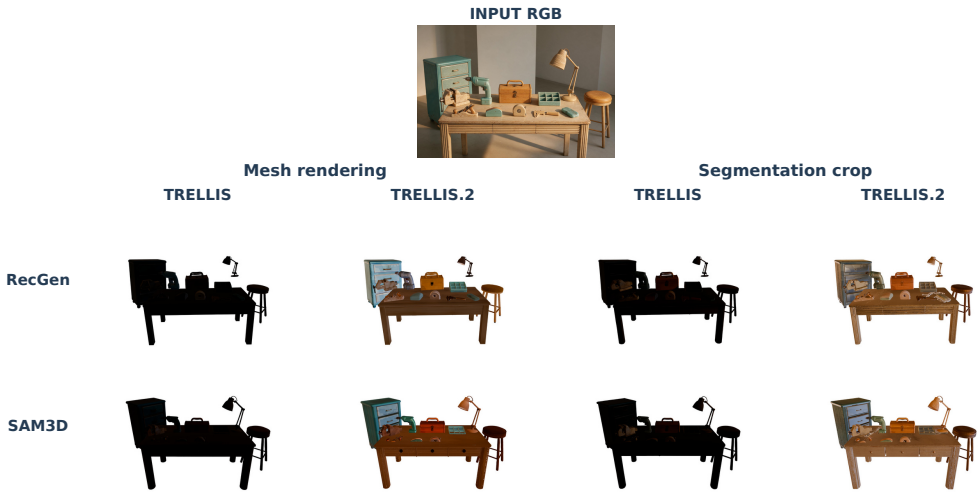

Figure 50: Direct-conditioning comparison for workshop example 1. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. The workbench tools and containers expose whether the condition preserves the distinct appearance of small instances. The cabinet and lamp provide larger reference objects for the same comparison. Both direct conditions omit identity-preserving inpainting; Fig. 31 shows the corresponding scene-generation comparison.

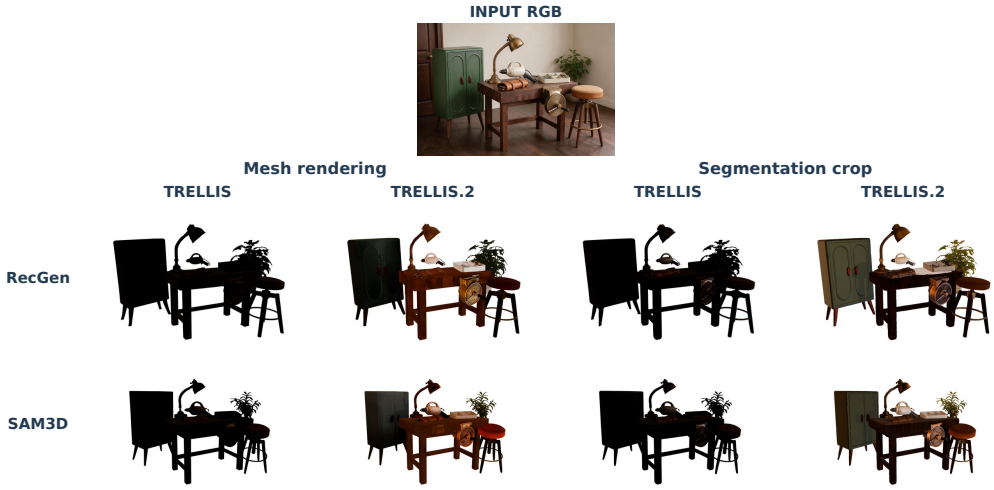

Figure 51: Direct-conditioning comparison for workshop example 2. Rows use RecGen and SAM3D; columns use a coarse-mesh rendering with TRELLIS and TRELLIS.2, followed by a segmentation crop with the same two refiners. The green cabinet, wood bench, and plant remain recognizable but differ in surface appearance under the two conditions. This illustrates the need for a completed visual condition that retains the target object's observed identity. Both direct conditions omit identity-preserving inpainting; Fig. 32 shows the corresponding scene-generation comparison.