POG: POINT-CONDITIONED OBJECT GENERATION FOR COMPOSITIONAL 3D SCENE RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Compositional 3D scene reconstruction seeks to recover complete object-centric assets and their spatial arrangement from images. Native 3D object generation models, which learn powerful shape and appearance priors from large-scale asset collections, have recently enabled the synthesis of complete, textured objects and inspired compositional scene reconstruction from a single image. Yet a single view provides inherently incomplete evidence: occlusion and perspective ambiguity leave both object geometry and spatial arrangement underconstrained. Multi-view images offer complementary observations that can reduce these ambiguities, but leveraging them with canonical object priors requires associating evidence across views and reconciling scene-space observations with the canonical coordinates used by object generators. In this paper, we present PoG, a compositional 3D scene reconstruction method that generates a complete 3D asset for each observed object instance from multi-view RGB images through point-conditioned object generation. Given only multi-view images, an off-the-shelf reconstruction and segmentation frontend establishes cross-view instance observations and derives a partial point cloud for each object; no depth or object geometry is provided as input. PoG augments these points with DINO features and encodes the resulting feature-bearing point set into spatially grounded condition tokens. It then jointly generates canonical object occupancy and predicts the relative similarity transformation between the canonical asset and the normalized observation frame. The predicted transformation canonicalizes the point conditions for subsequent geometry and texture generation and places each completed asset back into the scene, eliminating post-hoc optimization-based registration. Experiments on Toys4K and multi-view 3D-FRONT demonstrate the effectiveness of PoG for object- and scene-level reconstruction.

1 INTRODUCTION

Reconstructing a 3D scene from multi-view images is a long-standing problem in computer vision and graphics, with applications in augmented and virtual reality, robotic simulation, and embodied AI (Ye et al., 2026). Traditional reconstruction methods (e.g., Structure-from-Motion (Schönberger & Frahm, 2016; Pan et al., 2024) and Multi-View Stereo (Schönberger et al., 2016)) and recent feed-forward reconstruction foundation models (Wang et al., 2026; 2025a; 2024) have achieved strong performance in recovering scene geometry from images. However, their geometric outputs typically capture only the observed scene surfaces (Figure 1(a)), whereas many downstream applications require structured, object-centric representations. Such representations should decompose the scene into individual objects, recover complete object geometry beyond the observed regions, and explicitly characterize each object's pose, thereby enabling object-level manipulation, editing, and simulation.

Recent 3D object generation models (Xiang et al., 2025; Zhang et al., 2024; Tencent Hunyuan3D Team, 2025; Wu et al., 2024; Li et al., 2025a) learn native 3D priors from large-scale 3D asset collections, enabling the synthesis of complete and textured objects from a single image. This capability has inspired a line of *compositional* scene generation methods that recover complete object assets along with their spatial arrangement from a single image. These methods typically follow one of two strategies. *Decompose-then-align* approaches generate individual objects from masked image regions and estimate their placement through subsequent registration (Yao et al., 2025; Dogaru

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.

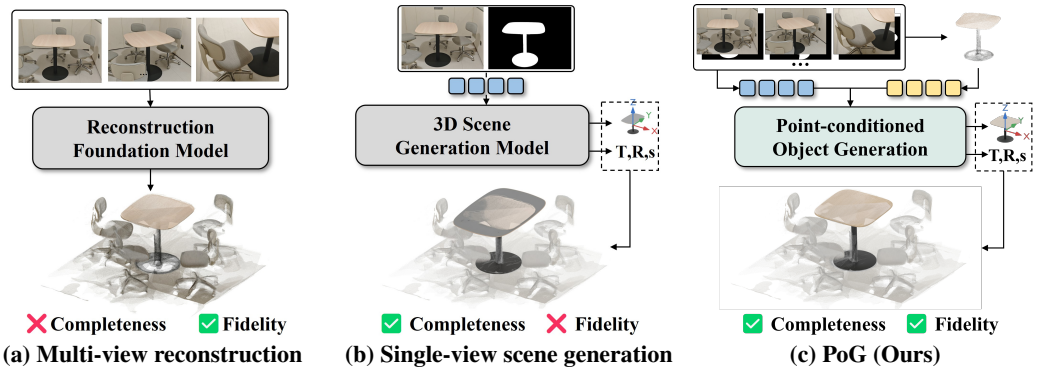


Figure 1: **Combining multi-view reconstruction with 3D generative priors.** (a) Multi-view reconstruction provides reliable scene geometry but produces partial, entangled surfaces. (b) Single-view compositional generation recovers complete object assets and predicts their poses, but inherits ambiguity in occluded geometry and spatial arrangement. (c) PoG conditions a 3D generative model on multi-view object point clouds with lifted image features, producing complete canonical assets together with explicit poses for scene composition.

et al., 2025). Alternatively, methods such as SceneGen and SAM 3D adapt generative models to predict object assets together with their spatial arrangement from image and mask inputs (Meng et al., 2026; Chen et al., 2026), as illustrated in Figure 1(b). Both strategies combine object completion with scene composition, but their reliance on a single view leaves ambiguity in occluded geometry and object placement: distinct shapes and spatial configurations can explain the same visible evidence.

Multiple views can reduce this ambiguity by revealing complementary surfaces and appearance cues. Recent methods integrate multi-view reconstruction features with generative priors (Chang et al., 2026) or lift image features into spatial conditions for scene-scale generation (Schmid et al., 2026). Our focus is to adapt object-level generators to such observations while recovering separate canonical assets and their placement. Feature-bearing instance point clouds provide a shared representation of the multi-view evidence.

A central challenge arises from the coordinate-frame mismatch between partial observations and canonical 3D asset priors. Object generators are trained on complete assets in canonical frames, whereas instance point clouds reside in scene coordinates and cover only the visible surfaces. Normalizing a partial cloud is insufficient: its center and extent vary with viewpoint and occlusion, while its orientation is not tied to the canonical asset convention. Consequently, treating completion and placement as independent steps either presents the generator with frame-inconsistent conditions or requires post-hoc registration whose solution need not agree with the generated geometry. A coherent formulation should instead infer canonical structure and the observation-to-canonical relationship jointly, then reuse this relationship both to canonicalize downstream conditions and to compose the generated assets in the scene.

To realize this formulation, we present **PoG**, a multi-view point-conditioned framework for compositional 3D scene reconstruction (Figure 1(c)). Given associated instance observations, a learned Point Conditioner encodes point positions and attached DINO features into a fixed-length token sequence, which adapts the occupancy, geometry, and texture generators of TRELLIS.2 (Xiang et al., 2026). PoG jointly generates canonical occupancy and predicts a relative similarity transformation between the canonical asset and normalized observation frames. This transformation serves two complementary roles: its inverse maps the observed points into canonical coordinates to provide frame-consistent conditions for geometry and texture generation, while its forward direction places the completed asset back into the scene without optimization-based registration. The resulting per-instance formulation supports object-level supervision and reconstructs scenes by composing independently generated assets without requiring joint generation of all objects.

Our main contributions are summarized as follows:

- We formulate multi-view instance reconstruction through feature-bearing partial point clouds, adapting an object-level generative prior to recover assets for scene composition.
- We couple canonical structure generation with relative placement prediction and use the predicted transformation to provide frame-consistent point conditions for subsequent geometry and texture generation.
- We evaluate multi-view object and scene reconstruction on Toys4K and 3D-FRONT, respectively, demonstrating the effectiveness of the proposed method.

2 Related Work

Feed-forward 3D reconstruction. Building on classical Structure-from-Motion and Multi-View Stereo, recent feed-forward models recover geometry directly from images. DUSt3R (Wang et al., 2024) predicts aligned point maps from image pairs, while VGGT (Wang et al., 2025a) and π^3 (Li et al., 2025b) support multi-view geometry and camera estimation. MoGe (Wang et al., 2025b) focuses on monocular geometry. These models primarily represent observed surfaces rather than complete, individually parameterized assets. Their outputs can provide geometric observations for PoG, whose learned component focuses on object completion and explicit placement.

Generative priors for object reconstruction. Native 3D generators learn object priors from large asset collections. TRELLIS (Xiang et al., 2025) introduces structured 3D latents, TRELLIS.2 (Xiang et al., 2026) develops a sparse-voxel representation for geometry and appearance, and Direct3D (Wu et al., 2024) uses a 3D latent diffusion transformer. PoG adopts the TRELLIS.2 backbone and latent representations. Beyond single-image generation, ReconViaGen (Chang et al., 2026) uses VGGT features to condition multi-view object reconstruction. Its scene extension reconstructs segmented objects separately and registers them using predicted camera poses. AmodalGen3D (Zhou & Tai, 2025) combines amodal image priors with multi-view stereo cues for complete object reconstruction from sparse unposed views. These approaches demonstrate the relevance of generative priors to reconstruction; PoG instead encodes explicit instance point clouds and predicts their relative canonical-to-observation transformation.

Single-view compositional scene generation. Existing methods differ in how they couple asset generation and placement. CAST (Yao et al., 2025) and Gen3DSR (Dogaru et al., 2025) generate individual assets and subsequently align them to scene observations. MIDI (Huang et al., 2025) and SceneGen (Meng et al., 2026) use cross-instance interaction to coordinate generation. In contrast, SAM 3D (Chen et al., 2026) jointly predicts shape and layout for a mask-specified object, optionally using a scene point map. Its shape–layout architecture includes one-way information sharing and stop-gradient, an architectural precedent for our layout branch. 3D-Fixer (Yin et al., 2026) performs in-place completion conditioned on partial geometry and spatially projected appearance features, preserving placement without explicitly factorizing a canonical asset and its pose. PoG uses this explicit factorization with multi-view instance observations and canonicalizes its conditions for later generation stages.

Multi-view generative scene reconstruction. MV-SAM3D (Li et al., 2026a) extends SAM 3D through training-free fusion of per-view denoising predictions and physics-aware pose optimization. GenRecon (Schmid et al., 2026) also builds on TRELLIS.2, lifting multi-view image features into spatial conditioning grids for generation over overlapping scene chunks. It learns from scene-level data and produces a scene mesh rather than separately parameterized canonical objects. PoG instead consolidates observations into feature-bearing instance point clouds and generates each object with an explicit relative placement transformation. Its adaptation uses object-level supervision, and its inference pipeline assembles the predicted assets without post-hoc optimization-based registration.

3 Methodology

PoG reconstructs observed instances as canonical object assets with explicit placement transformations. As shown in Figure 2, it aggregates multi-view observations into point-based conditions, then generates sparse structure and layout, geometry, and texture. We inherit the three-stage generative

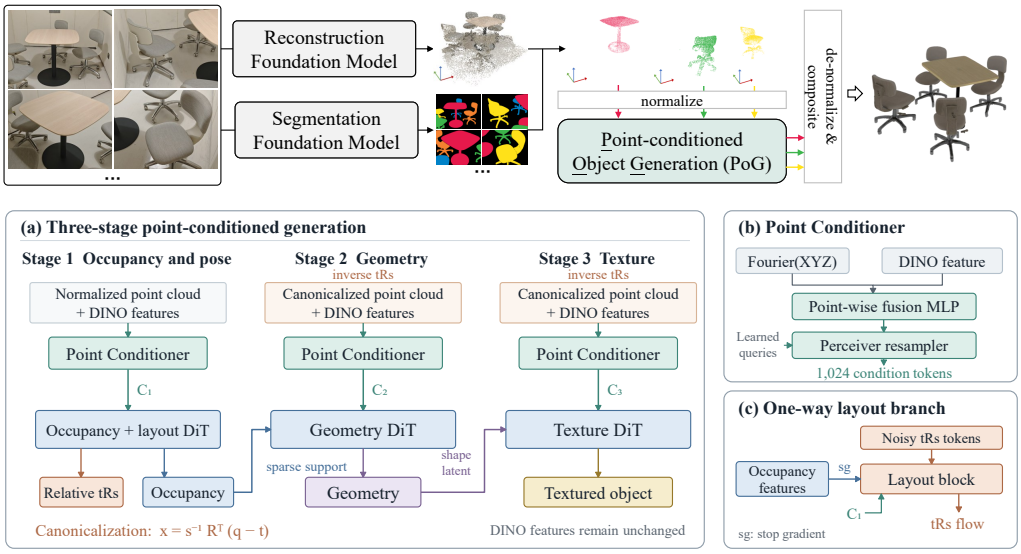


Figure 2: **Overview of PoG.** Multi-view observations are consolidated into per-instance point clouds with lifted DINO features. The occupancy–layout model encodes these observations with a dedicated Point Conditioner and jointly predicts canonical occupancy and a relative similarity transformation $(\mathbf{t}, \mathbf{R}, s)$. Its inverse maps the point conditions into the canonical frame for geometry and texture generation, whereas the forward transformation places the completed asset back into the scene. The inset panels detail point-token construction and the one-way layout branch, which predicts the transformation from detached occupancy features; the geometry and texture models each employ their own Point Conditioner.

backbone and latent representations from TRELLIS.2 (Xiang et al., 2026). Our adaptation replaces image-token conditioning with feature-bearing point conditions and couples the observation and asset coordinate frames through a predicted relative transformation.

3.1 Observations and Point Conditioning

Per-instance observations. We assume V RGB images with per-view depth, camera parameters, and instance masks whose object identities are associated across views. These inputs may come from an upstream reconstruction and segmentation pipeline; instance discovery and cross-view association are outside the learned PoG module. We back-project masked pixels into a common reference-camera frame and attach frozen DINOv3 (Siméoni et al., 2025) descriptors from their source image patches. Aggregation yields $P_i = \{(\mathbf{p}_{ik}, \mathbf{f}_{ik})\}_{k=1}^{M_i}$ for each instance i. Each point retains its source-view feature, without explicit feature averaging across views. Thus, scene composition operates on the supplied instances rather than discovering unobserved objects or learning inter-object relations.

Coordinate frames. The canonical frame is the coordinate system of the preprocessed complete training asset: the asset is centered and uniformly scaled to a fixed canonical cube, while retaining its imported orientation rather than imposing a shared semantic front direction. We normalize the observed points using their axis-aligned bounding-box center $\mathbf{c}_i$ and maximum side length a_i. This *normalized observation frame* differs from the *canonical object frame*. For a canonical point $\mathbf{x}$, we define the relative similarity transformation by

$$\mathbf{q}_{ik} = \frac{\mathbf{p}_{ik} - \mathbf{c}_i}{a_i}, \qquad \mathbf{q} = s_i \mathbf{R}_i \mathbf{x} + \mathbf{t}_i, \tag{1}$$

where $\mathbf{R}_i \in SO(3)$, $\mathbf{t}_i \in \mathbb{R}^3$, and $s_i > 0$. Observation normalization depends on the visible object extent, so alignment requires translation and scale as well as rotation. Once the asset and transformation are generated, a point is placed in the reference frame as $\mathbf{p} = a_i(s_i \mathbf{R}_i \mathbf{x} + \mathbf{t}_i) + \mathbf{c}_i$

and mapped to the scene using the known reference-camera transform. The output inherits the scale of this input frame; predicting s_i does not independently recover metric scale from scale-ambiguous observations.

Point Conditioner. We sample N points per instance and encode each position together with its attached feature:

$$\mathbf{h}_{ik}^{(j)} = \mathrm{MLP}_j\Big[\gamma(\mathbf{u}_{ik}^{(j)}) \,\|\, \mathbf{f}_{ik}\Big], \qquad C_i^{(j)} = \mathrm{Resampler}_j\Big(\{\mathbf{h}_{ik}^{(j)}\}_k\Big), \tag{2}$$

where $j \in \{1,2,3\}$ indexes the occupancy–layout, geometry, and texture models, respectively; γ denotes Fourier position encoding, and $\|$ denotes concatenation. The occupancy–layout model uses $\mathbf{u}^{(1)} = \mathbf{q}$, whereas the geometry and texture models use canonicalized coordinates defined below. An L-block Perceiver-style resampler (Jaegle et al., 2021) uses learned queries to produce K condition tokens of width d_c. These tokens enter the corresponding DiT through cross-attention. The three conditioners share an architecture but have separate trainable parameters. Pairing coordinates with descriptors anchors visual evidence to the observed geometry, while the learned queries provide a fixed token budget across view counts.

3.2 Joint Sparse Structure and Layout Generation

Placement is defined relative to the generated asset, so its prediction should have access to the corresponding shape hypothesis. Following joint shape–layout generation in SAM 3D (Chen et al., 2026), we couple a layout flow to sparse-structure generation. The resulting model jointly denoises an occupancy latent $\mathbf{z}^{\mathrm{occ}}$ and a layout vector $\mathbf{y} = [\mathbf{t}, \mathbf{r}_{\mathrm{6D}}, \log s]$ at the same flow time, conditioned on $C^{(1)}$. The continuous rotation representation (Zhou et al., 2019) is converted to a rotation matrix by orthogonalization. Decoding $\mathbf{z}^{\mathrm{occ}}$ yields occupied voxel coordinates $\mathcal{S}$ in canonical space, while $\mathbf{y}$ determines the relative transformation in Equation 1. This generative formulation permits alternative shape–placement hypotheses; it does not remove the inherent pose ambiguity of symmetric or occluded objects.

We adapt the one-way shape–layout interaction and stop-gradient design of SAM 3D (Chen et al., 2026) to the occupancy backbone. Our implementation appends a lightweight layout head after the occupancy backbone, rather than using the multi-layer mixture-of-transformers architecture of SAM 3D. Separate semantic tokens represent translation, rotation, and scale. The layout head applies token self-attention, cross-attention to detached occupancy features, cross-attention to point-condition tokens, and a feed-forward layer. This gives placement access to the structure hypothesis without feeding layout tokens back into the occupancy stream. Detaching the occupancy keys and values blocks the layout loss along that feature path, but the shared conditioner and time-conditioning parameters remain trainable. Joint sampling supplies structure and placement without a subsequent optimization-based registration step.

3.3 Canonical Geometry and Texture Generation

The geometry and texture latents are defined in canonical object space. Retaining observation-frame coordinates would leave the point conditions misaligned with the spatial support of these latents. We therefore canonicalize the observed points using the predicted relative transformation, with hats denoting estimates:

$$\mathbf{u}_{ik}^{(2)} = \mathbf{u}_{ik}^{(3)} = \widehat{\mathbf{R}}_i^{\top} \frac{\mathbf{q}_{ik} - \widehat{\mathbf{t}}_i}{\widehat{s}_i}. \tag{3}$$

The DINO features remain attached to their original points. Separate geometry and texture conditioners re-encode the transformed point-feature pairs into $C^{(2)}$ and $C^{(3)}$, respectively.

The geometry model generates a sparse latent $\mathbf{z}^{\mathrm{geo}}$ on the predicted occupied coordinates $\mathcal{S}$. Thus, occupancy specifies *where* geometry latents are generated, while the canonical point tokens condition their values through cross-attention. The texture model generates $\mathbf{z}^{\mathrm{tex}}$ on the same support. At each denoising step, the geometry latent is concatenated channel-wise with the noisy texture latent, and $C^{(3)}$ supplies point-grounded appearance information through cross-attention. Frozen decoders convert the generated geometry and texture latents into a textured canonical asset. Applying its predicted similarity transformation and undoing observation normalization composes the final scene

from independently generated objects. The downstream latent flows keep the supplied sparse coordinates fixed: they refine latent values rather than proposing additional support. Consequently, errors in the predicted occupancy and placement can propagate to geometry and texture generation. Scene assembly does not impose additional contact or collision constraints between independently generated objects.

3.4 TRAINING

The three models are trained separately on rendered object-level data, without scene-level supervision. For each model, training samples a variable-size subset of the available observations. One sampled camera defines a random reference frame. Surface-removal and depth-boundary perturbations augment the point observations; the view-sampling and augmentation settings are specified in Appendix A.

We use rectified-flow velocity matching (Liu et al., 2023). For target latent $\mathbf{z}_0$ and Gaussian noise $\boldsymbol{\epsilon}$, the interpolation is $\mathbf{z}_\tau = (1-\tau)\mathbf{z}_0 + [\sigma_{\min} + (1-\sigma_{\min})\tau]\boldsymbol{\epsilon}$. Each model predicts the target velocity $(1-\sigma_{\min})\boldsymbol{\epsilon} - \mathbf{z}_0$ using mean-squared error. The joint objective combines occupancy and component-wise layout losses:

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{occ}} + \lambda \left(\mathcal{L}_{\mathbf{t}} + \mathcal{L}_{\mathbf{r}_{\text{6D}}} + \beta_s \mathcal{L}_{\log s} \right), \tag{4}$$

where β_s weights scale supervision and λ is a clipped exponential moving average of the occupancy-to-weighted-layout loss ratio. Rotation is supervised by velocity MSE in the 6D representation. Occupancy and layout use independent Gaussian noise at a shared logit-normal flow time τ, whereas geometry and texture training sample τ uniformly. The loss-ratio moving average is detached from gradient computation. Numerical settings for the flow distribution, loss weights, moving average, and clipping interval are given in Appendix A.

Training uses ground-truth intermediate targets: the geometry model receives occupied support, and the texture model receives both support and geometry latents. Both use the ground-truth relative transformation, perturbed in rotation, translation, and log-scale, to canonicalize their conditions. At inference, generated occupancy, placement, and geometry replace these ground-truth quantities. We train the full occupancy DiT together with its layout head and conditioner. Geometry and texture adaptation use separate conditioners and rank-r LoRA adapters (Hu et al., 2022), with the pretrained latent encoders and decoders frozen.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

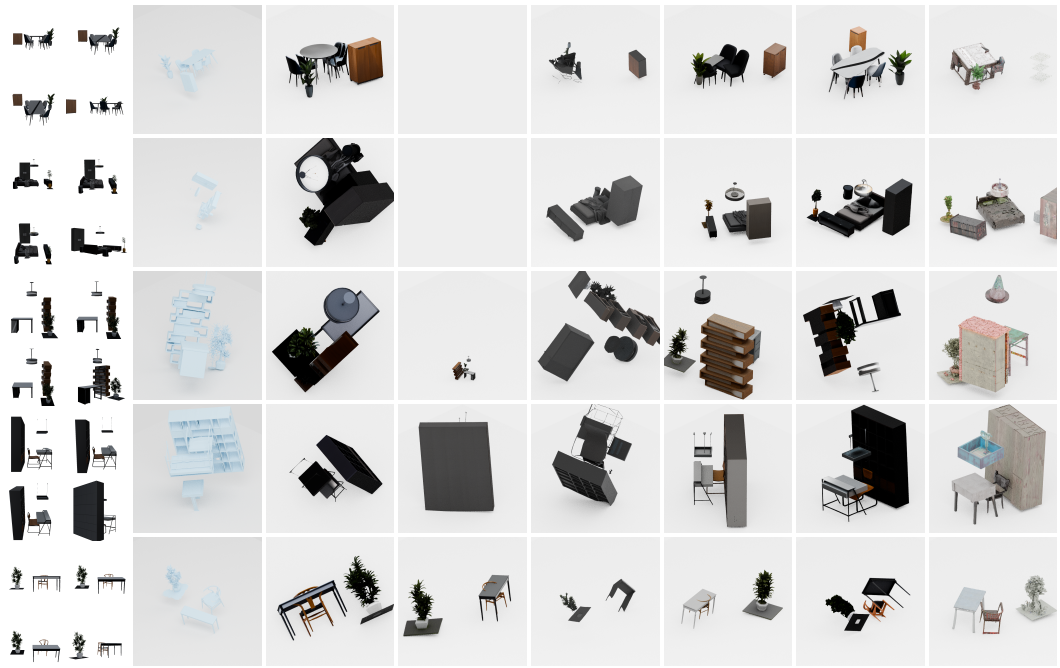
Datasets. We train on the rendered RGB-D subset of TRELLIS-500K (Xiang et al., 2025), with roughly 75K usable objects for shape training and 76K for PBR training. Each object has 16 known-camera RGB-D views, sampled as described in Section 3.4. On Toys4K (Stojanov et al., 2021), PoG and all baselines are evaluated using the same number of objects for geometry and appearance metrics. We additionally evaluate scene geometry on a multi-view 3D-FRONT (Fu et al., 2021) benchmark.

Evaluation metrics. Following prior compositional scene generation work (Huang et al., 2025; Yin et al., 2026), the reported geometry evaluation uses Chamfer Distance (CD; lower is better), F-Score (F1; higher is better), and bounding-box IoU (BBox-IoU; higher is better). Scene-level F1 is evaluated at 5 and 10 cm, whereas object-level F1 is evaluated at 0.1. Scene metrics are averaged over evaluated scenes; object metrics are first averaged within each scene and then across scenes. For Toys4K baselines, we additionally report all-view PSNR and LPIPS means. In the occupancy-generation study, Occ.-CD is the symmetric Chamfer distance between predicted and complete ground-truth boundary-voxel centers, while Obs.-CD is the one-sided mean distance from normalized input depth points to the predicted occupancy boundary.

Baselines. The current scene comparison includes MIDI (Huang et al., 2025), SceneGen (Meng et al., 2026), SAM 3D (Chen et al., 2026), and 3D-Fixer (Yin et al., 2026) with single-view input, and ReconViaGen (Chang et al., 2026) and MV-SAM3D (Li et al., 2026a) with multi-view input. The Toys4K baselines are Pixal3D (Li et al., 2026b), MV-SAM3D, and ReconViaGen.

Table 1: **Geometry comparison on the multi-view 3D-FRONT benchmark.** "S" and "M" denote single-view and multi-view input, respectively. SAM 3D and MV-SAM3D use pose-corrected outputs, whereas ReconViaGen is evaluated as a whole-scene output.

		Scene			Object		
Method	Views	CD ↓	F1@10cm ↑	F1@5cm ↑	CD ↓	F1@0.1 ↑	BBox-IoU ↑
MIDI (Huang et al., 2025)	S	0.2680	0.3221	0.1827	0.2170	0.3722	0.1175
SceneGen (Meng et al., 2026)	S	0.4579	0.4750	0.2855	0.1367	*0.4659*	0.2181
SAM 3D (Chen et al., 2026)	S	0.7728	0.4190	0.2801	0.1474	0.4479	0.1517
3D-Fixer (Yin et al., 2026)	S	*0.1129*	*0.5929*	0.4046	*0.1298*	0.4610	*0.2539*
ReconViaGen (Chang et al., 2026)	M	0.0283	0.7976	0.6054	–	–	–
MV-SAM3D (Li et al., 2026a)	M	0.1758	0.5864	*0.4146*	0.0629	0.6375	0.2686
PoG (Ours)	M	**0.0203**	**0.8250**	**0.6159**	**0.0128**	**0.8677**	**0.4496**



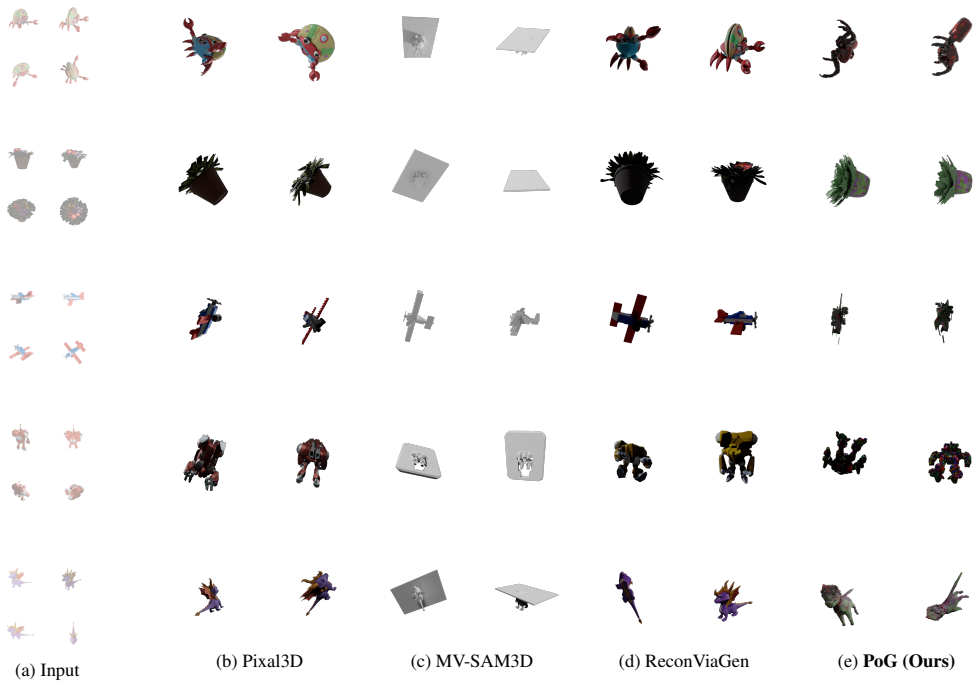
(a) input (b) MIDI (c) SceneGen (d) SAM 3D (e) 3D-Fixer (f) MV-SAM3D (g) ReconViaGen (h) **PoG (Ours)**

Figure 3: **Qualitative comparisons on the multi-view 3D-FRONT test set.**

Implementation details. We use TRELLIS.2 and DINOv3 ViT-L/16. We set $N = 8192$, $L = 4$, $K = 1024$, and $d_c = 1024$ for each Point Conditioner. The joint occupancy–layout model adapts the full occupancy DiT, whereas the geometry and texture models use LoRA adapters with rank $r = 16$. All three conditioners are independently trained, with the pretrained latent encoders and decoders frozen. Full architectural, view-sampling, augmentation, loss, and optimization settings are provided in Appendix A. The occupancy-only conditioning study compares separate adaptation settings, including LoRA and full-model training. For the reported scene evaluations, we construct Point+DINO observations using dataset-provided ground-truth depth, camera parameters, and object-label masks.

4.2 Main Results

Table 1 compares geometric reconstruction on the multi-view 3D-FRONT benchmark. PoG achieves the lowest Chamfer distance and the highest F1 at both the scene and object levels, together with the highest bounding-box IoU. In this evaluation, the available ReconViaGen outputs are whole-scene reconstructions, so object metrics are not reported.



(a) Input (b) Pixal3D (c) MV-SAM3D (d) ReconViaGen (e) **PoG (Ours)**

Figure 4: **Qualitative comparison on the multi-view Toys4K dataset.**

Figure 3 presents qualitative comparisons across diverse indoor scenes. Compared with single-view baselines, PoG leverages complementary multi-view observations to recover more complete object geometry and more consistent spatial arrangements. Relative to existing multi-view methods, its object-centric formulation produces clearer instance separation and individually textured assets while preserving the overall scene structure. These results demonstrate the advantage of combining spatially grounded point conditions with native 3D object priors for compositional scene reconstruction, although challenging objects may still exhibit residual geometric or texture artifacts.

Table 2 reports unaligned object geometry and all-view appearance on Toys4K, with all methods evaluated using the same number of objects. Among the baselines, ReconViaGen has the lowest mean CD (0.0091) and highest mean F1 (0.9091). PoG obtains CD 0.0062 and F1 0.9238, together with PSNR 16.11 and LPIPS 0.2122.

Table 2: **Object geometry and appearance on Toys4K.** Geometry uses no post-hoc alignment; appearance is averaged over all rendered views. All methods are evaluated using the same number of objects. Best/second/third values are bold/underlined/italic, respectively.

	Geometry		Appearance	
Method	CD ↓	F1@0.1 ↑	PSNR ↑	LPIPS ↓
Pixal3D (Li et al., 2026b)	0.0455	0.5500	15.77	*0.1941*
MV-SAM3D (Li et al., 2026a)	*0.0342*	*0.6399*	<u>17.90</u>	<u>0.1876</u>
ReconViaGen (Chang et al., 2026)	<u>0.0091</u>	<u>0.9091</u>	**18.73**	**0.1049**
PoG (Ours)	**0.0062**	**0.9238**	*16.11*	0.2122

Figure 4 complements the aggregate results with matched-view object comparisons. The two matched output views expose differences in global shape, thin structures, and appearance across methods. PoG produces complete textured assets for all five selected objects, although visible shape deviations remain in challenging structures such as the crab, airplane, and dragon.

4.3 Ablation Study

Effect of the number of input views. Figure 5 shows consistent improvement as more observations are provided: mean scene CD decreases from 0.0269 with one view to 0.0127 with seven

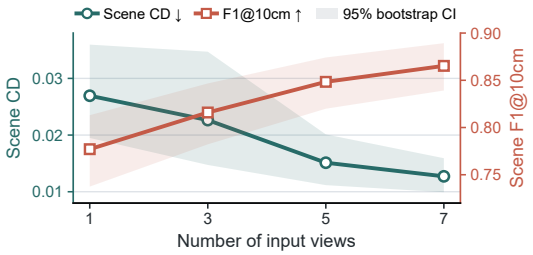


Figure 5: **Effect of the number of input views on MV-3DFront.**

Table 3: **Ablation study of conditioning representations on the multi-view Toys4K subset.**

	Point	DINO	Occ.-CD ↓	Obs.-CD ↓
(a)	×	✓	0.1722	0.1285
(c)	✓	×	0.2552	0.1587
(e)	✓	✓	**0.0902**	**0.0501**

views, while F1@10cm increases from 0.7771 to 0.8653. The largest CD reduction occurs between three and five views, after which both metrics continue to improve more gradually.

Conditioning representations. Table 3 examines conditioning representations for occupancy generation, with the layout branch disabled. Image-DINO (a) is the unmodified single-image TRELLIS.2 baseline. Point-only (c) and Point+DINO (e) condition the model on XYZ coordinates and on paired XYZ–DINO features, respectively. Point+DINO achieves the lowest Occ.-CD and Obs.-CD, indicating that the two modalities provide complementary geometric and semantic evidence.

5 Conclusion

We presented PoG, a framework for adapting object-level generative priors to multi-view instance observations through point-feature conditioning and explicit canonical-to-observation transformations. The model uses these transformations both to align downstream conditions and to compose generated assets into a scene. Geometric evaluations demonstrate promising object and scene reconstruction under the reported input protocols, complemented by selected textured scene visualizations. The aggregate view-count sweep is consistent with improved reconstruction from additional observations. Experiments on multi-view Toys4K and 3DFRONT dataseets demonstrate the effectivness of the propose method.

AI use statement

Generative AI tools were used to provide feedback on experimental design and presentation and to assist with literature search, manuscript drafting and language editing, figure and table design, and code for data processing, visualization, and document preparation. They were not used to generate research data, formulate mathematical claims or proofs, or implement the proposed learning method. The authors reviewed and verified all AI-assisted outputs, including the cited literature, numerical results, code, figures, and manuscript text, and take full responsibility for the final content of this work.

References

Jiahao Chang, Chongjie Ye, Yushuang Wu, Yuantao Chen, Yidan Zhang, Zhongjin Luo, Chenghong Li, Yihao Zhi, and Xiaoguang Han. ReconViaGen: Towards accurate multi-view 3D object reconstruction via generation. In *International Conference on Learning Representations (ICLR)*, 2026.

Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J. Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, Aohan Lin, Jia-Wei 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7220–7232, 2026.

Andreea Dogaru, Mert Özer, and Bernhard Egger. Gen3DSR: Generalizable 3D scene reconstruction via divide and conquer from a single view. In *International Conference on 3D Vision (3DV)*, 2025.

Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, and Hao Zhang. 3D-FRONT: 3D furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Zehuan Huang, Yuan-Chen Guo, Xingqiao An, Yunhan Yang, Yangguang Li, Zi-Xin Zou, Ding Liang, Xiaojuan Cao, and Lu Sheng. MIDI: Multi-instance diffusion for single image to 3D scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Andrew Jaegle, Felix Gimeno, Andy Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning (ICML)*, 2021.

Baicheng Li, Dong Wu, Jun Li, Shunkai Zhou, Zecui Zeng, Lusong Li, and Hongbin Zha. MV-SAM3D: Adaptive multi-view fusion for layout-aware 3D generation. *arXiv preprint arXiv:2603.11633*, 2026a.

Dong-Yang Li, Wang Zhao, Yuxin Chen, Wenbo Hu, Meng-Hao Guo, Fang-Lue Zhang, Ying Shan, and Shi-Min Hu. Pixal3D: Pixel-aligned 3D generation from images. *arXiv preprint arXiv:2605.10922*, 2026b.

Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, and Yan-Pei Cao. TripoSG: High-fidelity 3D shape synthesis using large-scale rectified flow models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–18, 2025a. doi: 10.1109/TPAMI.2025.3633512.

Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Scalable permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025b.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

Yanxu Meng, Haoning Wu, Ya Zhang, and Weidi Xie. SceneGen: Single-image 3D scene generation in one feedforward pass. In *International Conference on 3D Vision (3DV)*, 2026.

Linfei Pan, Daniel Barath, Marc Pollefeys, and Johannes Lutz Schönberger. Global structure-from-motion revisited. In *European Conference on Computer Vision (ECCV)*, 2024.

Katharina Schmid, Nicolas von Lützow, Jozef Hladký, Angela Dai, and Matthias Nießner. GenRecon: Bridging generative priors for multi-view 3D scene reconstruction. *arXiv preprint arXiv:2605.23888*, 2026.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Stefan Stojanov, Anh Thai, and James M. Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1798–1808, 2021.

Tencent Hunyuan3D Team. Hunyuan3D 2.0: Scaling diffusion models for high resolution textured 3D assets generation. *arXiv preprint arXiv:2501.12202*, 2025.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. MoGe: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3D: Scalable image-to-3D generation via 3D latent diffusion transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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, 2025.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14419–14429, 2026.

Kaixin Yao, Longwen Zhang, Xinhao Yan, Yan Zeng, Qixuan Zhang, Lan Xu, Wei Yang, Jiayuan Gu, and Jingyi Yu. CAST: Component-aligned 3D scene reconstruction from an RGB image. *ACM Transactions on Graphics (TOG)*, 44(4), 2025.

Tianwei Ye, Yifan Mao, Minwen Liao, Jian Liu, Chunchao Guo, Dazhao Du, Quanxin Shou, Fangqi Zhu, and Song Guo. 3D generation for embodied AI and robotic simulation: A survey. *arXiv preprint arXiv:2604.26509*, 2026.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12753–12763, 2026.

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.

Junwei Zhou and Yu-Wing Tai. AmodalGen3D: Generative amodal 3D object reconstruction from sparse unposed views. *arXiv preprint arXiv:2511.21945*, 2025.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

A Implementation Details

Canonical assets and relative-transform targets. Asset preprocessing uniformly scales the complete imported geometry by its bounding-box extent and centers it at the origin, fitting it inside $[-0.5, 0.5]^3$. It preserves the imported orientation; no semantic axis alignment is imposed. The voxelization path applies a small boundary-safe shrinkage. For a reference camera with extrinsics $(\mathbf{R}_r, \mathbf{t}_r)$ in these rendering coordinates, $\mathbf{p} = \mathbf{R}_r\mathbf{x} + \mathbf{t}_r$. Combining this relation with observation normalization yields the training targets

$$\mathbf{R}_{\mathrm{gt}} = \mathbf{R}_r, \qquad \mathbf{t}_{\mathrm{gt}} = \frac{\mathbf{t}_r - \mathbf{c}_i}{a_i}, \qquad \log s_{\mathrm{gt}} = -\log a_i. \tag{5}$$

The rotation vector contains the first two columns of $\mathbf{R}_{\mathrm{gt}}$; Gram–Schmidt orthogonalization recovers a rotation matrix at inference.

Point sampling and encoding. We set $N = 8192$ and compute the observation bounding box over all valid points before randomly sampling the point-feature pairs, using replacement only when fewer points are available. Each point receives the 1024-dimensional DINOv3 ViT-L descriptor of the patch containing its source pixel. For each coordinate, the Fourier encoder uses ten frequencies $\omega_k = 2^{6k/9}$, $k = 0, \ldots, 9$, and concatenates the raw coordinate with $\sin(\omega_k x)$ and $\cos(\omega_k x)$. There is no additional factor of π. The resulting 63-dimensional position encoding is concatenated with the descriptor. A $1087 \to 512 \to 512$ MLP with an intermediate GELU produces the point features. We use $L = 4$ resampling blocks, $K = 1024$ learned queries, condition width $d_c = 1024$, eight attention heads, and an MLP ratio of four. Each block applies query self-attention, cross-attention to point features, and a feed-forward layer. A non-affine LayerNorm produces the condition tokens.

Backbone interfaces and adaptation. The occupancy DiT has 30 blocks, width 1536, 12 attention heads, and an 8-channel latent on a 16^3 grid. Its frozen decoder produces a 64^3 occupancy grid; locations with decoder output greater than zero define the sparse support. The implementation converts the decoder's XYZ axis order to a ZYX mask and converts the extracted support coordinates back to XYZ for the sparse flows. The layout head contains one block and operates on three tokens. Geometry and texture flows use 32-channel sparse latents. The texture DiT receives 64 channels by concatenating noisy texture with clean geometry and predicts a 32-channel texture velocity. The geometry and texture DiTs have 30 blocks of width 1536. Their LoRA adapters use rank $r = 16$ and $\alpha = 16$, modifying self-attention QKV/output and cross-attention Q/KV/output projections; their base weights remain frozen. All three Point Conditioners have independent parameters. The latent encoders, decoders, and DINO feature extractor are frozen.

Optimization settings. All three generative models use AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 0.01, and $\epsilon = 10^{-8}$, with bfloat16 mixed precision. Joint occupancy–layout training uses a learning rate of 10^{-4} for the occupancy DiT and 10^{-3} for its conditioner and layout branch, with a batch size of two per GPU. Geometry and texture training use 10^{-4} for both the LoRA parameters and conditioners, with a batch size of four per GPU split into two microbatches. Model-parameter EMA uses decay 0.9999, distinct from the adaptive loss-weight EMA. The occupancy-only study in Table 3 uses the separately reported adaptation strategies and training budgets.

Augmentation and training conditions. View subsets are sampled without replacement, and the first sampled view defines the reference frame. Occupancy–layout training uses 1, 2, or 4 views; geometry and texture training use 1, 2, 4, 8, or 16. The surface-removal and depth-boundary perturbation probabilities are 0.1 and 0.05 for the occupancy–layout model, and 0.5 and 0.3 for the geometry and texture models. All three models use condition dropout with probability 0.1. To perturb downstream pose conditions, we sample a random unit rotation axis and an angle from $\mathcal{N}(0, (5^\circ)^2)$, and left-multiply the resulting rotation with the target rotation. Independent translation noise has standard deviation 0.02 in each normalized-observation coordinate. Scale is multiplied by $\exp(\eta)$ with $\eta \sim \mathcal{N}(0, 0.02^2)$. We set $\sigma_{\min} = 10^{-5}$ and $\beta_s = 0.1$. Occupancy and layout use independent Gaussian noise at a shared time $\tau = \mathrm{sigmoid}(u)$, where $u \sim \mathcal{N}(1, 1)$; their adaptive loss weight uses the worker-averaged, detached ratio $\mathcal{L}_{\mathrm{occ}}/(\mathcal{L}_{\mathrm{layout}} + 10^{-8})$ with EMA decay 0.99, clipped to $[0.2, 5]$.

Predicted-input inference. For each associated instance, inference (i) constructs and normalizes its feature-bearing cloud, (ii) samples occupancy and layout jointly, (iii) decodes occupied support and applies the inverse predicted transformation to the conditioning points, (iv) samples geometry on this support, (v) samples texture conditioned on the generated geometry and canonicalized points, and (vi) decodes and transforms the asset into the scene frame. The flow samplers use explicit Euler integration from noise at $\tau = 1$ to data at $\tau = 0$, with classifier-free guidance. Geometry and texture decoding follows the backbone's sparse decoding interfaces, with the geometry subdivision guide supplied to the texture decoder. Component-level diagnostics instead substitute the ground-truth intermediate quantities identified in their evaluation protocols.