FORGE3D: FAITHFUL OBSERVATION-GROUNDED 3D MESH GENERATION VIA LATENT FORCING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent 3D mesh generators produce visually compelling assets, yet faithfully reconstructing specific objects from real-world observations remains challenging. A practical mesh generation framework must jointly preserve observed 3D geometry and pixel-aligned appearance from one or more views while plausibly completing unseen regions. However, these observations often serve only as coarse conditioning signals, allowing generated geometry and appearance to deviate from the input. Moreover, existing 3D mesh benchmarks provide limited coverage of realistic scenes and challenging observation conditions needed to assess reconstruction faithfulness in real-world scenarios. To address these gaps, we introduce FORGE3D (Faithful Observation-Grounded 3D Mesh Generation), an observation-faithful mesh generation pipeline, and FORGE3DBench, a benchmark for evaluating observation faithfulness under realistic multi-view conditions. FORGE3D directly encodes geometric and visual observations into the generative latent space through geometry and appearance latent forcing, initializing generation from observed evidence. Applied to a pretrained generator, latent forcing already improves geometric fidelity without additional training, and finetuning with correspondence-aware multi-view attention further improves appearance. FORGE3DBench features more than 300 unique assets across 20 simulated backgrounds with 24 camera viewpoints, encompassing varied object placements, lighting conditions, and occlusions. Across three benchmarks, FORGE3D achieves state-of-the-art multi-view reconstruction performance in both geometry and appearance and improves downstream 6D pose estimation. Demo is available at https://forge3d-iclr2027.github.io/.

1 INTRODUCTION

Reconstructing 3D meshes from visual observations is a fundamental task in computer vision, supporting applications in robotics, augmented reality, simulation, and digital content creation. Recent advances in 3D generative models have enabled the creation of detailed, visually compelling meshes from images (Xiang et al., 2025; Chen et al., 2026; Zhao et al., 2025). By learning strong priors over geometry and appearance, these models can synthesize complete objects even from partial observations, making them promising tools for reconstructing objects encountered in real-world environments. Practical applications, however, require the generated mesh to match the observed parts of the particular object, and any deviation of its shape or texture from the observations limits its usefulness in downstream tasks. Extending mesh generation to such applications therefore requires preserving observed geometry and appearance while plausibly completing unseen regions.

Real-world observations provide complementary evidence for this task. Depth measurements from sources such as stereo cameras provide 3D points that should be preserved during generation, while image pixels constrain the appearance of the mesh. Additional views reveal previously occluded surfaces and help resolve ambiguities. Existing generators exploit this evidence through image conditioning (Chen et al., 2026; Xiang et al., 2025), features from 3D reconstruction models (Chang et al., 2026), and multi-view conditioning (Li et al., 2026; Wu et al., 2025; Zhao et al., 2025). Nevertheless, observations supplied as conditioning signals can still be overridden by the generative prior, producing plausible meshes whose local structures or textures deviate from the input. Jointly preserving observed 3D geometry and pixel-aligned appearance thus remains a central challenge.

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.

Assessing this faithfulness also requires evaluation under the conditions in which objects are encountered in practice. Isolated object renderings offer limited coverage of contextual clutter, natural object placements, varying illumination, and occlusions caused by hands (Hampali et al., 2020; Chao et al., 2021) or robot grippers. These conditions determine which surfaces are visible and how reliably their geometry and appearance can be recovered. Evaluation under such conditions is therefore necessary to establish whether a generator preserves available evidence while plausibly completing unobserved regions. Advancing practical mesh generation requires addressing both how observations guide generation and how faithfully the resulting meshes reflect those observations in realistic settings.

To address these connected challenges, we introduce **FORGE3D**, an observation-faithful mesh generation framework, and **FORGE3DBench**, a benchmark for evaluating this capability under realistic multi-view conditions. FORGE3D explicitly initializes generation from observed geometric and visual evidence. FORGE3DBench varies the scenes and observation conditions that determine the evidence available to a generator, enabling evaluation of its ability to reconstruct observed surfaces and complete missing regions.

FORGE3D is based on **geometry and appearance latent forcing** (Figure 1). Our central insight is that observations can directly populate the native latent spaces of a pretrained 3D generator (Xiang et al., 2025; Chen et al., 2026), allowing generation to begin from evidence already available in the input. Geometry latent forcing encodes aggregated 3D observations into the source latent of the geometry-generation stage. Appearance latent forcing uses voxel-to-image correspondences to aggregate DINOv2 image features (Oquab et al., 2024) across the views that observe each voxel and encode them into the structured latent used for mesh generation. Observed locations receive these appearance-derived latents, while unobserved locations receive noise to support completion. This construction grounds both stages in the input observations while retaining the generative prior's ability to synthesize unseen surfaces.

Applied to the frozen pretrained generator, latent forcing already improves geometric fidelity without additional training. We further fine-tune both the sparse-structure and structured-latent flows (Lipman et al., 2023) to generate complete meshes from partially observed latent sources, which also improves appearance. Correspondence-aware multi-view attention complements this initialization by directing voxel queries toward relevant image patches across views while retaining global context from an anchor view. The resulting framework uses aggregated observations to establish the generation source and individual image views to guide its subsequent evolution, improving faithfulness in both geometry and appearance.

FORGE3DBench comprises more than 300 unique assets across 20 realistic simulated scenes with 24 camera viewpoints. It encompasses diverse object placements, lighting conditions, and occlusion patterns, including objects grasped by hands or robot grippers. These variations expose a generator to evidence that is incomplete, occluded, or distributed across viewpoints. The benchmark therefore extends evaluation beyond isolated object reconstruction to the observation conditions that practical mesh generation must accommodate.

Our contributions comprise FORGE3D, which directly incorporates observed geometry and appearance into the generative source through latent forcing, and FORGE3DBench, which evaluates observation faithfulness under diverse and realistic observation conditions. Across three benchmarks, FORGE3D achieves state-of-the-art multi-view reconstruction performance in both geometry and appearance. Our experiments show that latent forcing improves geometric fidelity even without additional training, that fine-tuning with correspondence-aware multi-view attention further improves appearance, and that the reconstructed meshes improve downstream 6D pose estimation (Lee et al., 2025; Wen et al., 2024).

2 Related Work

Latent 3D generation and multi-view conditioning. Recent image-to-3D generators, including TRELLIS (Xiang et al., 2025) and SAM 3D Objects (Chen et al., 2026), generate a sparse voxel structure followed by spatially indexed latent features representing detailed geometry and appearance. Building on these generators, recent methods incorporate multiple observations to improve reconstruction fidelity. ReconViaGen (Chang et al., 2026) uses an external multi-view reconstruc-

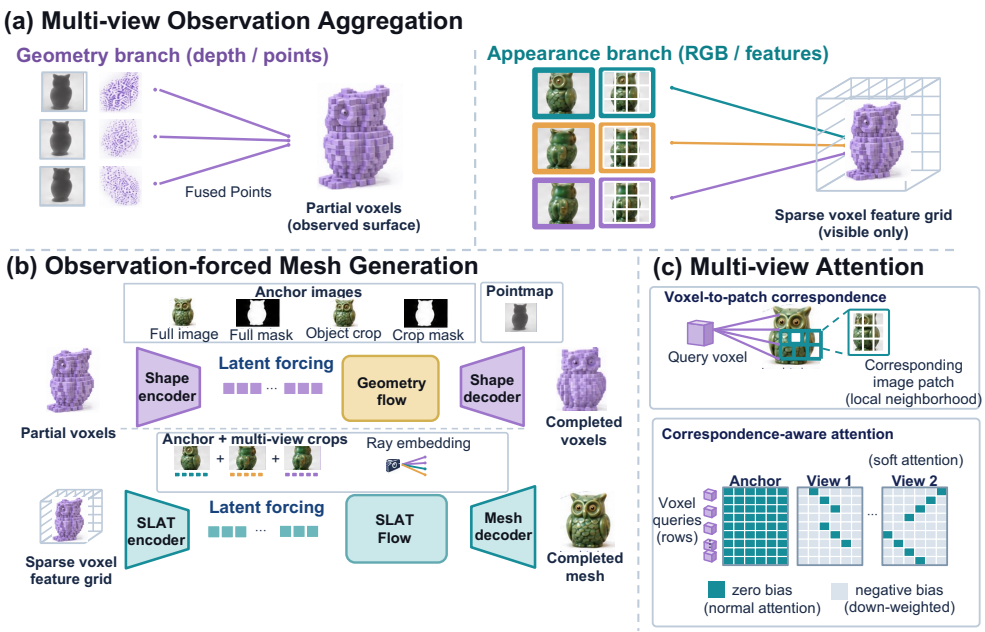

Figure 1: **Overview of FORGE3D. (a)** We back-project per-view depth into 3D, fuse the visible points in a shared canonical space, and use correspondences between voxels and image patches to aggregate multi-view image features at the observed voxels. **(b) Geometry forcing** initializes the sparse-structure flow with the encoded partial voxels, and the flow completes the unobserved geometry. **Appearance forcing** initializes the SLAT flow with the encoded image features at observed voxels and with noise at unobserved voxels; the flow then completes the latent, which is decoded into a textured mesh. **(c)** Correspondence-aware multi-view attention biases each voxel query toward the image patches it projects to, helping preserve observed details while completing unseen regions.

tion encoder to condition generation, while Pixal3D (Li et al., 2026) aggregates voxel-aligned DINO features across calibrated views. Other methods incorporate point-cloud geometry as an additional condition (Hu & Shugrina, 2026; Zadaianchuk et al., 2026), requiring alignment between geometric inputs and the pretrained generator. A closely related approach, CUPID (Huang et al., 2026), encodes pixel-aligned image evidence into SLAT features and concatenates these features with the evolving noisy latents along the channel dimension. We share the motivation of exploiting the generator's latent representation to incorporate observations. Our distinction is to use encoded multi-view evidence directly as the initial state of both geometry and appearance flows, preserving the pretrained latent input interface. This enables latent forcing to improve geometric fidelity even without fine-tuning, with further gains from training the flows to complete these partially observed initializations.

Guiding the sampling process. A separate line modifies the sampling process instead of the conditioning: RePaint (Lugmayr et al., 2022) overwrites known pixels of the noisy state at every step and resamples the rest, SDEdit (Meng et al., 2022) starts sampling from a noised copy of a supplied signal, and classifier guidance (Dhariwal & Nichol, 2021) steers each sampling step with an external gradient. All three intervene at test time on a model trained from pure noise, so nothing obliges the network to preserve the inserted signal. FORGE3D instead encodes the observation into the generator's own latent space and uses it once, as the starting point of the flow in place of noise. Without fine-tuning, this already improves geometry but lowers PSNR and SSIM (Table 3); we therefore also fine-tune both flows to start from the encoded observations.

Observation-grounded reconstruction and 6D pose. DUSt3R (Wang et al., 2024), MASt3R (Leroy et al., 2024) and VGGT (Wang et al., 2025) regress per-pixel point maps without a generative prior: they are accurate where the object is seen but give no geometry else-

where. FORGE3D can use such a point map in place of measured depth as the observation forced into its latent (Table 5). Render-and-compare pose estimators such as MegaPose (Labbé et al., 2022) and FoundationPose (Wen et al., 2024) assume a CAD model at test time; Any6D (Lee et al., 2025) instead generates the mesh from a reference image, making pose accuracy on HO-3D (Hampali et al., 2020) and DexYCB (Chao et al., 2021) a direct probe of mesh faithfulness, the protocol we adopt.

Benchmarks and synthetic data for 3D reconstruction. Several efforts have developed realistic synthetic scenes and annotations to support 3D reconstruction. RecGen (Zadaianchuk et al., 2026) constructs synthetic RGB-D observations of compositional multi-object and part-based scenes to train reconstruction models. Continuing this line of work, FORGE3DBench (Section 4) focuses on evaluating observation-faithful mesh generation from multiple views. It combines unseen objects with realistic scene backgrounds, varied lighting, and both unobstructed and occluded observations.

3 Method

3.1 Preliminaries: 3D Generation and Latent Representations

We build on SAM 3D Objects (Chen et al., 2026), a two-stage image-conditioned 3D generator. Its first stage generates a sparse voxel structure describing the object's coarse geometry, and its second stage generates structured latents (SLAT) on these voxels, which are decoded into a mesh with detailed geometry and appearance. Both stages operate in latent spaces defined by pretrained encoders, providing a direct interface for encoding observed evidence.

The sparse-structure encoder maps a voxel occupancy grid into a geometry latent. Consequently, geometric observations from multiple views can be encoded by transforming their observed 3D points into a shared coordinate frame, voxelizing the combined points, and applying the same encoder. This produces a latent representation of the observed surface, even when the observations cover only part of the object.

The SLAT representation (Xiang et al., 2025) is constructed from multi-view visual features. To encode a complete 3D asset, DINOv2 (Oquab et al., 2024) feature maps are extracted from images rendered around it. Each active voxel is projected into these images to retrieve features from the corresponding patches. When features from multiple views correspond to the same voxel, they are averaged to obtain a single voxel-aligned feature. The SLAT encoder processes these features as spatially indexed tokens and maps them into local latents representing fine geometry and appearance.

This construction naturally allows us to encode partial observations by applying the same procedure to the available views. Depth measurements or geometry estimated by an external model such as VGGT establish correspondences between observed 3D locations and image patches. Because appearance is represented by patch features, this aggregation does not require exact pixel-to-pixel matching across images, although accurate geometric correspondence remains beneficial. The existing encoders therefore allow us to express multi-view geometry and appearance directly in the generator's latent spaces without introducing a separate learned representation for combining these modalities.

3.2 Latent Forcing from Multi-view Observations

We use encoded multi-view observations to initialize the two generative flows. For geometry, we aggregate the observed 3D points in a shared canonical frame and voxelize them into a partial occupancy grid O. Applying the pretrained sparse-structure encoder gives

$$\mathbf{z}_{\mathrm{SS}}^{\mathrm{obs}} = \mathcal{E}_{\mathrm{SS}}(O). \tag{1}$$

We replace the sparse-structure flow's initial noise with this latent. The flow then completes the partial geometric representation, producing the active-voxel set $\mathcal{P}$ on which mesh generation proceeds.

For appearance, we determine which active voxels are visible in each input view using depth consistency or ray-marching through the voxel structure. Let $\mathcal{M}$ denote the voxels visible in at least one view and $\bar{\mathcal{M}} = \mathcal{P} \setminus \mathcal{M}$ the unobserved voxels. For each observed voxel $\mathbf{p}$, we retrieve the corresponding DINOv2 features from the views that see it. When features from multiple views

correspond to the same voxel, we average them to obtain a single voxel-aligned feature $\mathbf{f}^{(\mathbf{p})}$. We construct the encoder input by assigning these features to observed voxels and Gaussian noise to unobserved voxels:

$$\widetilde{\mathbf{f}}^{(\mathbf{p})} = \begin{cases} \mathbf{f}^{(\mathbf{p})}, & \mathbf{p} \in \mathcal{M}, \\ \boldsymbol{\epsilon}^{(\mathbf{p})}, \quad \boldsymbol{\epsilon}^{(\mathbf{p})} \sim \mathcal{N}(0, I), & \mathbf{p} \in \bar{\mathcal{M}}. \end{cases} \tag{2}$$

We then pass the entire voxel-aligned feature tensor through the pretrained SLAT encoder:

$$\mathbf{z}_{\mathrm{SLAT}} = \mathcal{E}\left(\{(\mathbf{p}, \widetilde{\mathbf{f}}^{(\mathbf{p})})\}_{\mathbf{p}\in\mathcal{P}}\right). \tag{3}$$

The SLAT flow starts from $\mathbf{x}_0 = \mathbf{z}_{\mathrm{SLAT}}$ and generates a complete latent representation, which is decoded into a textured mesh.

We call this initialization procedure **latent forcing**. Observations are encoded directly into the generative state, giving generation a starting point grounded in the particular object being reconstructed. The latent evolves freely after initialization, without repeated overwriting. As shown in Table 3, latent forcing already improves geometric fidelity when applied to the frozen pretrained backbone, demonstrating its effectiveness without additional training. We subsequently fine-tune both flows to better exploit these partially observed initializations, as described in Section 3.3.

3.3 Training with Multi-view Correspondence Attention

We propose a lightweight fine-tuning strategy that adapts both generation stages to transform partially observed latent initializations into complete-object latents. Unlike approaches that introduce external encoders or modalities absent from pretraining, we reuse the backbone's pretrained encoders, latent representations, and conditioning pathways. This reduces the need to learn new cross-modal alignments and is designed to retain the backbone's pretrained knowledge. We optimize both stages with a flow-matching objective, encouraging the model to learn to preserve the encoded evidence while completing missing geometry and appearance:

$$\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1, \qquad \mathcal{L}_{\mathrm{flow}} = \mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_1}\left[\|\mathbf{v}_\theta(\mathbf{x}_t, t, c) - (\mathbf{x}_1 - \mathbf{x}_0)\|_2^2\right], \tag{4}$$

where $t \sim \mathcal{U}(0,1)$, $\mathbf{x}_0$ is the initialization constructed by latent forcing, $\mathbf{x}_1$ is the complete-object target latent, and c denotes the conditioning inputs.

Sparse-structure flow. We retain the original anchor-view conditions shown in Figure 1 without introducing additional conditioning branches. Since multi-view geometry is already aggregated into the initial latent, we train this flow to map the encoded partial occupancy to the complete occupancy latent. This incorporates multi-view geometric evidence while preserving the backbone's conditioning architecture.

SLAT flow. We use the same objective to predict the complete-object SLAT from the encoded feature-and-noise initialization $\mathbf{z}_{\mathrm{SLAT}}$, supervising all active voxels. We retain the full anchor-view conditioning and supplement it with DINOv2 patch features from object-centric crops of auxiliary views, using the shared pretrained image projection. These features provide additional evidence about the target's appearance and fine geometry without replicating the full conditioning pathway for every view. We add camera embeddings derived from anchor-relative Plücker rays (Zhang et al., 2024; Tang et al., 2024) through a zero-initialized learned projection, then concatenate the auxiliary tokens with the anchor tokens.

Multi-view correspondence attention. To connect each voxel query with relevant image evidence, we introduce a geometric bias in the SLAT flow's cross-attention, as presented in Figure 1. For auxiliary views, we assign zero bias to the projected patch and its Chebyshev-radius-one neighborhood, yielding a 3×3 window clipped to the patch-grid boundaries; all other patches receive a bias of -10. This soft bias encourages attention to geometrically corresponding features. The bias is computed once per sample and reused across transformer blocks and sampling steps.

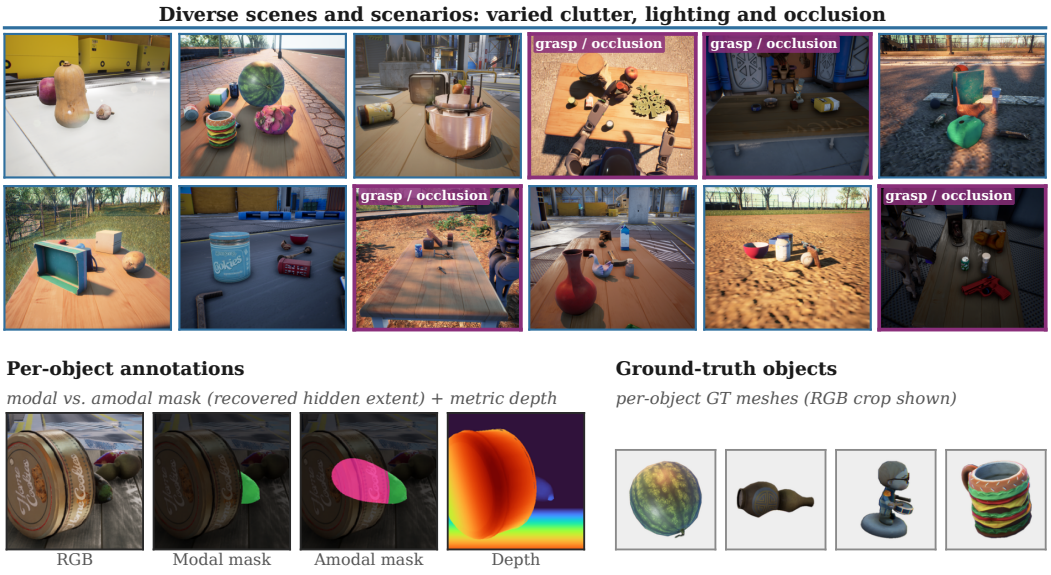

Figure 2: **FORGE3DBench overview.** **Gallery** (top): representative examples from FORGE3DBench, which comprises 20 scenes with diverse object collections and lighting conditions. Each scene is captured from 24 calibrated viewpoints. Highlighted tiles show robot grasping scenarios with partial and heavy object occlusion. **Per-object annotations** (bottom left): for an occluded object, the modal mask (green) covers only the visible region, and the amodal mask (magenta) covers the full object extent, including the parts occluded in that view. The metric depth map is shown alongside. **Ground-truth objects** (bottom right): each object is paired with a GT mesh; the RGB crops shown are cut out with each object's instance mask.

4 FORGE3DBENCH

Evaluating mesh generation for practical use requires diverse unseen objects, realistic object placements and occlusions, and varied scene backgrounds and illumination. Existing benchmarks often evaluate isolated objects against plain backgrounds (Stojanov et al., 2021) or offer limited object diversity (Chao et al., 2021), leaving these conditions insufficiently explored. To address this gap, we introduce **FORGE3DBench** (Figure 2), a benchmark for observation-faithful 3D mesh generation comprising 120 multi-object scenes assembled from more than 300 unique assets.

FORGE3DBench places objects naturally within realistic environments, including parks and factories, with occlusions arising from surrounding objects and supporting surfaces. We also include robot grasping scenarios using Fourier Intelligence's GR-1 (RoboAtlas), where the robot hand grasps the target object. These settings reflect practical applications that require reconstructing manipulated objects for downstream tasks such as grasp affordance prediction. We render the scenes in Unreal Engine (Epic Games, 2024), using physically based rendering to reproduce realistic materials and illumination. By varying lighting conditions and light-source positions, we capture changes in shadows, shading, and visibility encountered in real-world environments. Each scene is captured from 24 calibrated viewpoints at 1440×1440 resolution, with metric depth and per-instance modal and amodal masks. Together, these annotations support evaluation of geometric and appearance fidelity under varying visibility, illumination, and scene context. More examples are shown in Section A.

5 EXPERIMENTS

5.1 SETUP

Datasets. For training, we use the TRELLIS-500K dataset (Xiang et al., 2025) and the synthetic dataset of RecGen (Zadaianchuk et al., 2026), which provides rendered scenes with randomized

object placements. Following the TRELLIS data filtering procedure, we exclude samples below an aesthetic quality threshold and remove assets containing multiple objects or entire scenes. For mesh generation evaluation, we use 300 randomly selected objects from Toys4K (Stojanov et al., 2021), a collection of diverse textured toy assets; 100 subsampled objects from OmniObject3D (Wu et al., 2023), a dataset of scanned real-world objects; and FORGE3DBench (Section 4), our benchmark featuring realistic scenes with varied object placements, lighting, and occlusions. We run 638 generation scenarios in our benchmark. We additionally evaluate downstream 6D pose estimation on HO3D-MV and DexYCB-MV, multi-view settings of HO-3D (Hampali et al., 2020) and DexYCB (Chao et al., 2021) in which each object is reconstructed from at most four RGB-D reference views.

Metrics. We evaluate geometric fidelity using Chamfer-L1 (CD-L1) and F-score at a threshold of 0.01 (F@0.01) in unit-cube coordinates, and appearance quality using LPIPS, SSIM, and PSNR. Unless stated otherwise, each method receives four input views; we keep the front view fixed as an input and randomly sample the additional input viewpoints.For benchmarks with real-world captures or scene backgrounds (Wu et al., 2023), we use the captured images directly as model inputs rather than mesh renderings, and evaluate against additional captured views of the same object. For downstream 6D pose estimation, we report ADD, ADD-S, and BOP Average Recall (AR), which averages recall across the VSD, MSSD, and MSPD criteria.

Baselines. We compare against state-of-the-art textured mesh generation methods, including ReconViaGen (Chang et al., 2026), Amodal3R (Wu et al., 2025), CUPID (Huang et al., 2026), Hunyuan3D-2mv (Zhao et al., 2025), and Pixal3D (Li et al., 2026). We exclude methods that generate only untextured geometry, as our evaluation considers both geometric fidelity and appearance alignment with the input observations. We use the official multi-view implementations of Hunyuan3D-2mv, which accepts up to four input views, and Pixal3D, which additionally uses the ground-truth cameras.

Implementation details. We fine-tune both the sparse-structure and SLAT flows using AdamW with a cosine learning-rate schedule, gradient-norm clipping at 1.0, and an EMA decay of 0.999. We train the sparse-structure flow for 50k iterations with an effective batch size of 32, and the SLAT flow for 60k iterations with an effective batch size of 12. Training uses up to eight input views. To preserve the pretrained generative prior while supporting variable-view inputs, we omit latent forcing with probability 15%, use a single-view seed with probability 35%, and use seeds aggregated from a randomly selected number of views between two and eight with probability 50%. Training uses four NVIDIA A100 GPUs without gradient accumulation or gradient checkpointing and takes approximately four days in total.

5.2 Results

We evaluate the effectiveness of FORGE3D through three research questions:

Q1. Does FORGE3D reconstruct observed objects more faithfully than existing methods, and does latent forcing contribute to this fidelity?

Q2. Can FORGE3D effectively exploit additional input views?

Q3. Do the reconstructed meshes improve downstream 6D pose estimation?

Multi-view generation (Q1). Table 1 evaluates whether FORGE3D faithfully reconstructs both the geometry and appearance of the observed objects. Across Toys4K, OmniObject3D, and FORGE3DBench, FORGE3D achieves the best score in every column. These consistent gains, including under realistic backgrounds and partial occlusion, demonstrate our ability to generate faithful meshes despite variations in lighting, visibility, and scene context. This ability also extends to single-view generation, where FORGE3D achieves the best score on every metric (Table 6 in the appendix). Qualitative comparisons on FORGE3DBench are provided in Figure 4 in the appendix.

6D pose estimation (Q3). We evaluate whether the reconstructed meshes can reliably support downstream 6D pose estimation by using them as substitutes for ground-truth CAD models in an

Table 1: **Four-view mesh reconstruction results.** Each method receives four input views. ↓/↑ indicates lower/higher is better. We denote **best** and second best values within each dataset.

	Appearance			Geometry	
Method	LPIPS ↓	PSNR ↑	SSIM ↑	CD-L1 ↓	F@0.01 ↑
Toys4K *(synthetic objects)*					
ReconViaGen	0.0884	20.70	0.915	0.0147	0.516
Pixal3D	0.0651	23.33	0.932	0.0095	0.749
Amodal3R	0.1223	18.38	0.898	0.0221	0.407
Hunyuan3D-2mv	0.0970	20.46	0.913	0.0165	0.514
CUPID	0.1120	20.01	0.910	0.0227	0.481
FORGE3D	**0.0553**	**24.71**	**0.939**	**0.0080**	**0.845**
OmniObject3D *(real-scanned objects)*					
ReconViaGen	0.2038	16.80	0.847	0.0296	0.369
Pixal3D	0.1950	16.87	0.857	0.0259	0.358
Amodal3R	0.2114	16.00	0.824	0.0302	0.294
Hunyuan3D-2mv	0.2078	16.60	0.858	0.0271	0.339
CUPID	0.2184	16.10	0.847	0.0384	0.320
FORGE3D	**0.1628**	**18.25**	**0.873**	**0.0136**	**0.723**
FORGE3DBench *(multi-object scenes with occlusion)*					
ReconViaGen	0.1644	17.34	0.821	0.0236	0.401
Pixal3D	0.2394	15.26	0.806	0.0476	0.262
Amodal3R	0.1776	16.87	0.806	0.0264	0.342
Hunyuan3D-2mv	0.1767	17.56	0.835	0.0250	0.373
CUPID	0.2401	14.94	0.805	0.0486	0.244
FORGE3D	**0.0912**	**21.73**	**0.888**	**0.0128**	**0.701**

Table 2: **Model-free 6D object pose estimation from generated meshes** (Any6D/FoundationPose protocol). Every mesh is reconstructed from at most four RGB-D reference views of the same object. We denote **best** and second best values in each column.

	HO3D			DexYCB		
Method	ADD ↑	ADD-S ↑	AR ↑	ADD ↑	ADD-S ↑	AR ↑
ReconViaGen	47.0	97.0	53.2	69.0	88.0	69.5
Pixal3D	44.0	98.0	53.5	76.0	82.0	74.7
Amodal3R	7.0	13.0	7.6	41.0	52.0	40.2
Hunyuan3D-2mv	33.0	99.0	43.0	76.0	95.0	76.5
CUPID	24.0	61.0	29.2	44.0	73.0	48.8
FORGE3D	**76.0**	**100.0**	**73.6**	**78.0**	**98.0**	**76.7**

RGB-D pose estimation pipeline based on Any6D (Lee et al., 2025). Given multi-view observations from a video and their corresponding object masks, we first reconstruct the target mesh. For each query RGB-D frame, the pipeline uses the rendered views of the mesh to estimate the object's relative transformation between the reference frame and the query frame. Since pose estimation uses both mesh geometry and rendered appearance, this task evaluates whether reconstruction fidelity translates into practical utility. On HO3D and DexYCB, FORGE3D achieves the best out of the baselines across all metrics. These results demonstrate that the generated meshes can support downstream pose estimation without requiring ground-truth CAD models.

5.3 Ablation and Analysis

Component analysis (Q1). To analyze the role of each component, we progressively apply latent forcing and fine-tune the two generation stages, as shown in Table 3. Latent forcing improves geometric fidelity even when applied at test time to the frozen pretrained backbone, demonstrating

Table 3: **Ablation of latent forcing and fine-tuning.** We denote **best** and second best values in each column. Starting from the pretrained SAM 3D Objects (Chen et al., 2026), we cumulatively add training-free latent forcing, SLAT flow fine-tuning, and sparse-structure (SS) flow fine-tuning. LPIPS, PSNR, and SSIM are averaged over input and novel views.

Setting	LPIPS ↓	PSNR ↑	SSIM ↑	CD-L1 ↓	F@0.01 ↑
Pretrained model	0.1030	20.24	0.912	0.0159	0.524
+ training-free forcing	0.0905	19.36	0.903	0.0097	0.762
+ trained SLAT flow	0.0681	23.00	0.927	0.0095	0.771
+ trained SLAT + SS flow	**0.0554**	**24.72**	**0.939**	**0.0080**	**0.845**

Table 4: **View scaling.** Results of FORGE3D on Toys4K with different numbers of input views.

Views	LPIPS	PSNR	SSIM	CD-L1	F@0.01
1	0.0932	21.02	0.922	0.0223	0.597
2	0.0698	23.20	0.932	0.0131	0.745
4	0.0553	24.71	0.939	0.0080	0.845
8	0.0494	25.65	0.943	0.0067	0.889

Table 5: **Source of the geometry seed.** Geometry seeds from ground-truth depth versus VGGT-predicted point maps on Toys4K with four input views. LPIPS and CD-L1 are lower-is-better; PSNR, SSIM, and F-score are higher-is-better.

Seed	LPIPS	PSNR	SSIM	CD-L1	F@0.01
w/ Depth	0.0552	24.72	0.939	0.0080	0.846
w/ VGGT	0.0829	22.29	0.924	0.0206	0.570

its effectiveness without additional training. Fine-tuning the SLAT flow substantially improves appearance while retaining most of the geometric gains. Further fine-tuning the sparse-structure flow improves every metric. These findings highlight the complementary roles of test-time latent forcing and adaptation of the two training stages.

View scaling (Q2). Table 4 examines whether additional input views translate into better 3D mesh generation. The largest gain comes from adding a second view, and keep improves as more views are provided. This suggests that the method can integrate complementary evidence as additional views reveal previously unseen surfaces.

Performance without ground-truth depth. To evaluate applicability to real-world settings where ground-truth depth may be unavailable, we replace the depth-derived seed with a point-map seed predicted by VGGT. Table 5 compares these two sources on Toys4K using four input views, testing whether FORGE3D can retain reconstruction fidelity when its geometric evidence is estimated from images.

6 Conclusion

We presented FORGE3D, a framework for generating 3D meshes that remain faithful to observed geometry and appearance. Its central insight is that observations can directly initialize a generator's latent state, giving the generative prior a starting point grounded in the object being reconstructed. Latent forcing improves geometric fidelity even without additional training, and fine-tuning both flows further improves appearance as well as gains in geometry. Across our evaluations, these gains extend to additional input views and downstream 6D pose estimation. We also introduced FORGE3DBench to evaluate observation fidelity under realistic scene context, lighting variation, and occlusion. A limitation is that FORGE3D requires object masks, known cameras, and depth or estimated point maps for the input views. Moreover, since the seed is inserted only once at initialization, latent forcing encourages but does not guarantee the preservation of observed details. Reducing these input requirements is a natural direction for future work. Together, our method and benchmark advance 3D generation toward reconstructing the particular object observed.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods, cleaning and reformatting datasets, and assisting with translation. We have not used generative AI tools for proposing or refining hypotheses, developing theoretical models or conceptual frameworks, designing or providing feedback on research methodology or experiments, or interpreting results. Generating synthetic data sets, formulating mathematical claims, providing ingredients for or writing proofs, and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for drafting parts of the research paper. We have reviewed all AI-assisted work. The implemented methods are the model and training code, and the reformatted datasets are the human and robot demonstrations that our pipelines convert into point trajectories; two authors verified and tested this code for correctness. For the manuscript, AI tools assisted translation and drafting, and the authors reviewed, revised, and finalized every passage of the paper and its appendix. Three authors carried out and verified every analysis, including recomputing each reported success rate from the evaluation logs. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

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.

Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S. Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, Jan Kautz, and Dieter Fox. DexYCB: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

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.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Epic Games. Unreal Engine. Computer software, 2024. URL https://unrealengine.com.

Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. HOnnotate: A method for 3D annotation of hand and object poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. HO-3D dataset.

Anita Hu and Maria Shugrina. Axolotl3d: a unified framework for faithful 3d shape completion. In *European Conference on Computer Vision (ECCV)*, 2026.

Binbin Huang, Haobin Duan, Yiqun Zhao, Zibo Zhao, Yi Ma, and Shenghua Gao. CUPID: Generative 3D reconstruction via joint object and pose modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12741–12752, 2026.

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 *Conference on Robot Learning (CoRL)*, 2022.

Taeyeop Lee, Bowen Wen, Minjun Kang, Gyuree Kang, In So Kweon, and Kuk-Jin Yoon. Any6D: Model-free 6D pose estimation of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3D with MASt3R. In *European Conference on Computer Vision (ECCV)*, 2024.

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. In *ACM SIGGRAPH 2026 Conference Papers*, pp. 1–12, 2026. doi: 10.1145/3799902.3811175.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024.

RoboAtlas. Fourier GR-1. https://www.roboatlas.ai/en-US/products/fourier-gr-1. Accessed: 2026-09-25.

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)*, 2021. Toys4K dataset.

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 (ECCV)*, 2024.

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)*, 2025.

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

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 (CVPR)*, 2024.

Tianhao Wu, Chuanxia Zheng, Frank Guan, Andrea Vedaldi, and Tat-Jen Cham. Amodal3R: Amodal 3D reconstruction from occluded 2D images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9181–9193, 2025.

Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. OmniObject3D: Large-vocabulary 3D object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

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)*, 2025.

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. In *European Conference on Computer Vision (ECCV)*, pp. 498–519, 2026.

Jason Y. Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. In *International Conference on Learning Representations (ICLR)*, 2024.

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, et al. Hunyuan3D 2.0: Scaling diffusion models for high resolution textured 3D assets generation. *arXiv preprint arXiv:2501.12202*, 2025.

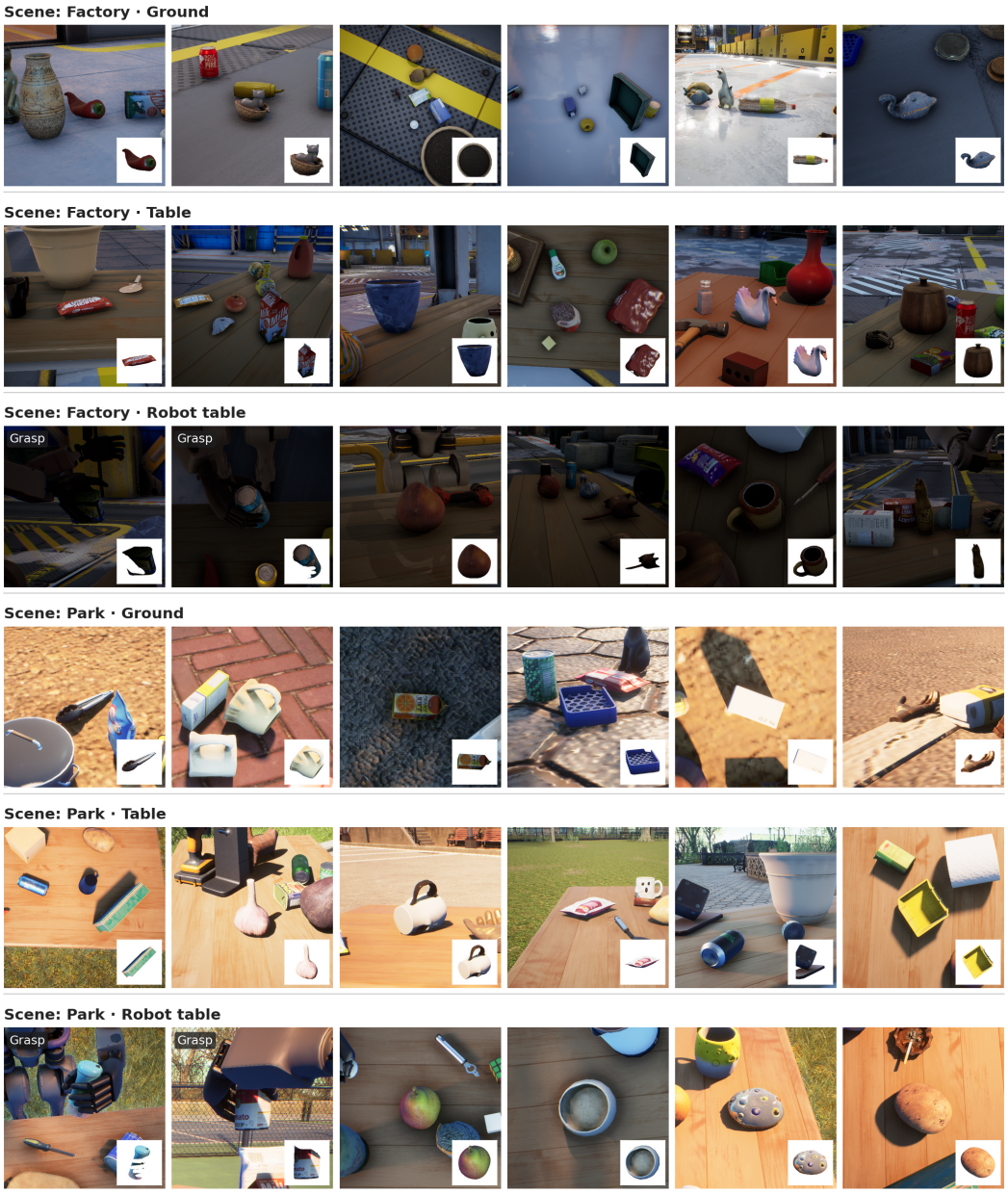

Figure 3: **Example scenes from FORGE3DBench.**

A FORGE3DBENCH EXAMPLES

Figure 3 shows example scenes from FORGE3DBench. The rows cover two environments, a factory and a park, each with three layouts: objects on the ground, objects on a table, and a table with the GR-1 robot, where tiles marked "Grasp" show the robot grasping the target object. The inset in each tile shows the target object cropped with its modal mask, so regions hidden by other objects or by the robot hand are missing. The examples span cluttered arrangements, varied floor and table materials, and illumination ranging from dim scenes to strong sunlight with hard shadows, which together determine how much of each target is observed.

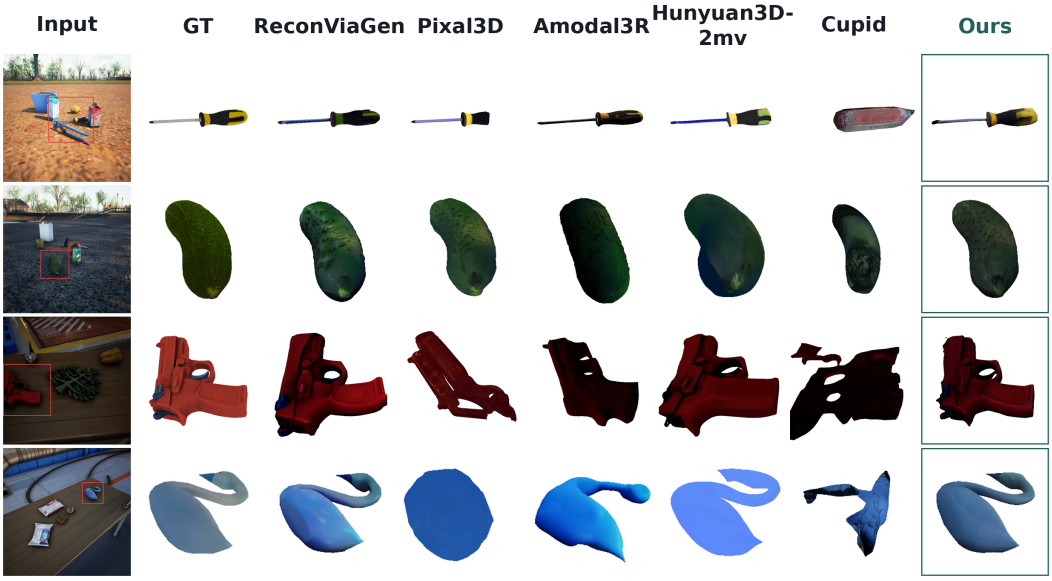

Figure 4: **Qualitative comparison on FORGE3DBench (multi-view input).** Input shows the anchor view given to each method, with the target object boxed in red; GT is the ground-truth rendering used for evaluation. Zoom in for details.

B SINGLE-VIEW RESULTS

We evaluate the single-view setting on the same 300 Toys4K objects as Table 1, where every method receives only the front view of each object, and compute the metrics as in Table 1. FORGE3D uses the same model as in the multi-view experiments: latent forcing encodes the observations of the front view alone, and the SLAT flow is conditioned on this anchor view without auxiliary-view tokens. Since 35% of the training samples use a single-view seed (Section 5.1), this setting requires no separate model.

Table 6 shows that FORGE3D achieves the best score on every metric. Compared with CUPID, the strongest baseline in this setting, it reduces LPIPS by 19% (0.1154 $\rightarrow$ 0.0932), improves PSNR by 1.43 dB (19.59 $\rightarrow$ 21.02), and raises F-score from 0.445 to 0.597. The margin in Chamfer-L1 is smaller (0.0237 $\rightarrow$ 0.0223), likely because a single view leaves most of the object unobserved, so the completed shape depends more on the generative prior. Notably, the single-view F-score of FORGE3D already exceeds those of four of the five baselines that receive four views in Table 1. These results show that the benefits of FORGE3D extend to single-view mesh generation, even without complementary views.

Table 6: **Single-view mesh reconstruction results on Toys4K.** We denote **best** and second best values in each column. Every method receives only the front view of each object. Appearance is evaluated at the input view and at novel views, and geometry using Chamfer-L1 (CD-L1) and F-score at a threshold of 0.01 in unit-cube coordinates. ↓/↑ indicates lower/higher is better.

	Appearance			Geometry	
Method	LPIPS ↓	PSNR ↑	SSIM ↑	CD-L1 ↓	F@0.01 ↑
ReconViaGen	0.1259	18.68	0.902	0.0270	0.375
Pixal3D	0.1321	19.07	0.907	0.0444	0.327
Amodal3R	0.1384	17.68	0.892	0.0289	0.350
Hunyuan3D-2mv	0.1175	19.57	0.907	0.0245	0.435
CUPID	0.1154	19.59	0.910	0.0237	0.445
FORGE3D	**0.0932**	**21.02**	**0.922**	**0.0223**	**0.597**