IMAGEO: BIDIRECTIONAL IMAGE–GEOMETRY CO-REASONING FOR 3D GENERATION

Anonymous authors
Paper under double-blind review

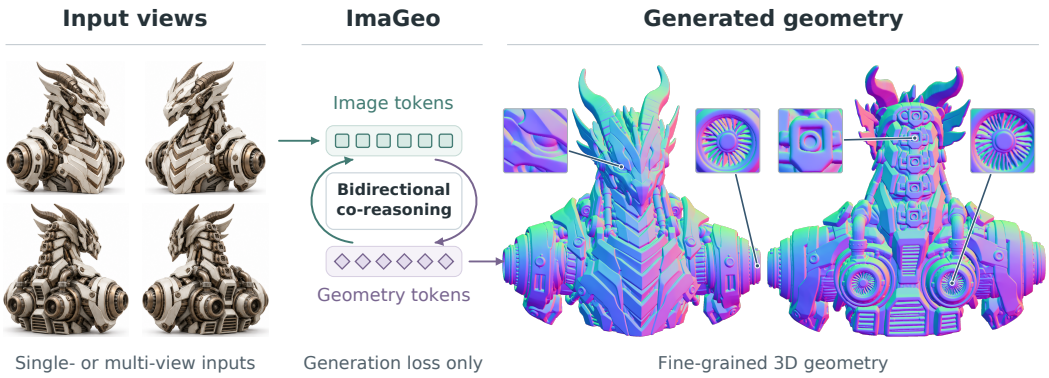

Figure 1: Bidirectional image–geometry co-reasoning in ImaGeo. Image evidence guides geometry, while geometry feeds back into image-token updates across transformer blocks. The same model supports one or multiple views with geometry-generation supervision.

ABSTRACT

Image-conditioned 3D generation requires fidelity to a single image's visible structure and details, and consistency with complementary observations when multiple views are available. We present ImaGeo, a diffusion transformer that generates 3D geometry from one or more input images. Its denoising blocks combine per-view image and geometry self-attention with joint attention, connecting all views to a shared geometry representation. Updated image features carry geometric and cross-view context into subsequent blocks while retaining per-view spatial processing. The same design operates in coarse and fine generation, trained solely through geometry flow matching without a dedicated reconstruction branch or auxiliary reconstruction losses. It supports variable-view inputs without input camera parameters. On GSO and Toys4K, ImaGeo improves single-view image–geometry consistency and sparse-view reconstruction accuracy. Evaluations on Internet-sourced and AI-generated images also show improved image–shape similarity. In controlled single-view fine-stage comparisons, our 1.5B denoiser outperforms a parameter-matched cross-attention baseline and achieves normal-map quality comparable to a 5B baseline.

1 INTRODUCTION

Image-to-3D generators can produce detailed 3D assets (Zhang et al., 2024; Xiang et al., 2025; Lai et al., 2026), but generating a plausible shape and faithfully matching the input images are distinct requirements. A single image constrains visible structure and surface detail, while additional views provide complementary observations that must be reconciled in the same geometry. Both single-view generation and sparse-view reconstruction therefore depend on how effectively visual evidence conditions the generative process. We study how image representations interact with geometry during generation to improve fidelity to the input observations.

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.

In single-image-conditioned 3D generation, a common design is to update geometry tokens through cross-attention to image features that remain fixed across denoising blocks. Hunyuan3D 2.0 (Zhao et al., 2025) provides a precedent for joint image–shape token processing, but does not report a controlled comparison isolating its benefit over cross-attention-based conditioning. Subsequent generators, including Direct3D-S2 (Wu et al., 2025), LATTICE (Lai et al., 2026), and TRELLIS.2 (Xiang et al., 2026), continue to use cross-attention for image conditioning.

With multiple input images, conditioning additionally requires organizing information within and across views. Reconstruction models such as VGGT (Wang et al., 2025) and π^3 (Wang et al., 2026) alternate per-image and cross-image attention. ReconViaGen (Chang et al., 2026) and RecGen3D (Huang et al., 2026b) incorporate reconstruction cues into geometry generation, while Mix3R (Lin et al., 2026) couples reconstruction and generation branches with additional geometric supervision.

We introduce ImaGeo, a diffusion transformer that generates 3D geometry from one or more input images (Figure 1). Each block applies per-view image self-attention with shared weights and separate geometry self-attention, followed by joint attention that updates both streams. Updated image tokens are passed to the next block's per-view processing. With multiple inputs, joint attention also exchanges information across views.

We adopt LATTICE's coarse-to-fine geometry representation (Lai et al., 2026) and use our image–geometry interaction blocks in both the coarse VecSet and fine VoxSet denoisers. Input images condition both overall shape formation and surface geometry refinement. The architecture supports single-image and sparse multi-view inputs without requiring camera parameters as inputs. Both denoisers are trained solely with geometry flow matching, without a dedicated reconstruction branch or auxiliary point-map or camera-prediction losses.

In single-view fine-stage generation, our 1.5B denoiser outperforms a parameter-matched cross-attention baseline and achieves normal-map quality comparable to a 5B baseline. Our experiments on GSO and Toys4K show that ImaGeo produces geometry more consistent with a single input image and reconstructs objects more accurately from sparse views. We also observe higher image–shape similarity on Internet-sourced and AI-generated inputs.

Our contributions are:

- We introduce an image–geometry co-reasoning block that combines per-view spatial processing with joint updates of image and geometry representations.
- We enable coarse-to-fine 3D generation conditioned on one or more images from different viewpoints, without requiring input camera parameters or a dedicated reconstruction branch, using geometry flow matching alone.
- We demonstrate the effectiveness of the proposed conditioning architecture through controlled comparisons and evaluations of single-view generation and sparse-view reconstruction.

2 Related Work

Image-conditioned 3D generation and alignment. Image-to-3D methods range from direct prediction, such as LRM (Hong et al., 2024), to diffusion and flow models over compact 3D representations (Zhang et al., 2023; 2024; Wu et al., 2024; Li et al., 2025a; Hunyuan3D et al., 2025). TRELLIS (Xiang et al., 2025), TRELLIS.2 (Xiang et al., 2026), and Direct3D-S2 (Wu et al., 2025) use structured or sparse representations, while LATTICE (Lai et al., 2026) and UltraShape (Jia et al., 2025) adopt coarse-to-fine pipelines for detailed geometry generation. To improve image alignment, Hi3DGen (Ye et al., 2025) uses predicted normals, CUPID (Huang et al., 2026a) jointly models shape and pose before pixel-aligned refinement, and Pixal3D (TRELLIS.2) (Li et al., 2026) back-projects image features onto geometry. We adopt LATTICE's geometry representation and focus on image conditioning through joint updates of image and geometry features within the denoiser.

Multi-view reconstruction with generative priors. Feed-forward methods such as DUSt3R (Wang et al., 2024b), VGGT (Wang et al., 2025), and π^3 (Wang et al., 2026) recover geometry from multiple images; VGGT and π^3 alternate per-image and cross-image attention

to organize information within and across views. Several methods combine reconstruction cues with generative priors. ReconViaGen (Chang et al., 2026) uses reconstruction-aware features and rendering-based guidance to improve consistency with the input views. RecGen3D (Huang et al., 2026b) aligns reconstruction and generation in a shared canonical space, using reconstructed geometric anchors and latent-augmented conditioning to guide completion and refinement. Mix3R (Lin et al., 2026) couples pretrained reconstruction and generation branches through a Mixture-of-Transformers to jointly predict coarse geometry, point maps, and cameras. ImaGeo learns multi-view image–geometry interactions within both coarse and fine denoisers through geometry generation alone, without a dedicated reconstruction branch or auxiliary point-map or camera-prediction losses.

Joint image–geometry modeling. In text-to-image generation, MM-DiT (Esser et al., 2024) enables bidirectional image–text interaction with modality-specific weights. For image-to-3D generation, Hunyuan3D 2.0 (Zhao et al., 2025) adopts FLUX-inspired dual- and single-stream blocks for joint image–shape token processing. PF-LRM (Wang et al., 2024a) jointly processes image and triplane tokens for reconstruction and pose estimation. A different form of geometry feedback is used by 3D-Adapter (Chen et al., 2024), which feeds rendered RGBD views back into image diffusion. Our block places separate per-view image and geometry self-attention before joint attention over all image and geometry tokens, and carries the updated image tokens into the next block's per-view processing.

3 METHOD

3.1 PROBLEM SETUP AND REPRESENTATION

Given N images of an object, $\mathcal{I} = (I_1, \ldots, I_N)$, our goal is to generate geometry consistent with the observed views while completing unobserved surfaces. The formulation covers single-view generation ($N = 1$) and sparse-view reconstruction ($N > 1$). We adopt the two-stage representation of LATTICE (Lai et al., 2026): Stage 1 generates coarse geometry using VecSet latents, and the decoded coarse mesh provides spatial anchors for fine geometry generation with VoxSet latents in Stage 2.

A vision encoder maps each image to patch tokens $\mathbf{v}_i^{\text{enc}} \in \mathbb{R}^{L_i \times d_v}$, while a 3D VAE encodes the target geometry as a clean latent $\mathbf{z}_1 \in \mathbb{R}^{M \times d_z}$ after fixed latent scaling. Given a noisy geometry latent $\mathbf{z}_t$ at time t, the DiT (Peebles & Xie, 2023) predicts a latent velocity $\hat{\mathbf{u}}_t = f_\theta(\mathbf{z}_t, t, \mathcal{I}) \in \mathbb{R}^{M \times d_z}$. For Stage 2, the denoiser additionally uses the spatial anchors obtained from the coarse mesh; we omit this dependence from the notation for brevity.

3.2 BIDIRECTIONAL IMAGE–GEOMETRY CO-REASONING

Figure 2 contrasts static cross-attention with our co-reasoning block. We project the noisy geometry latents to transformer width d, $\mathbf{h}_g^0 = P_z(\mathbf{z}_t) \in \mathbb{R}^{M \times d}$, and embed the timestep as $\mathbf{q}_t^0 \in \mathbb{R}^{1 \times d}$. At block ℓ, $\mathbf{g}^\ell = [\mathbf{q}_t^\ell; \mathbf{h}_g^\ell]$ denotes the combined timestep and geometry sequence.

Static cross-attention. For a single input image, let $\mathbf{v}_1 = P_v(\mathbf{v}_1^{\text{enc}})$ be its fixed patch features, where P_v is a learnable projection from image-feature dimension d_v to transformer width d. The baseline applies geometry self-attention followed by cross-attention:

$$\tilde{\mathbf{g}}^\ell = \mathbf{g}^\ell + \text{SA}_{3D}^\ell(\text{LN}(\mathbf{g}^\ell)), \tag{1}$$

$$\mathbf{g}_{\text{CA}}^\ell = \tilde{\mathbf{g}}^\ell + \text{CA}^\ell(\text{LN}(\tilde{\mathbf{g}}^\ell), \mathbf{v}_1). \tag{2}$$

Only the geometry stream is updated; $\mathbf{v}_1$ remains fixed across blocks.

Our block instead updates both streams through modality-specific self-attention followed by joint attention. We initialize each image sequence by prepending five learnable tokens to its projected patch features:

$$\mathbf{v}_i^0 = [\mathbf{r}_i; P_v(\mathbf{v}_i^{\text{enc}})] \in \mathbb{R}^{(L_i+5) \times d}. \tag{3}$$

Stage 1 uses $\mathbf{r}_1 = \mathbf{r}_{\text{ref}}$ for the first view and $\mathbf{r}_i = \mathbf{r}_{\text{other}}$ for $i > 1$. During multi-view training, this distinction identifies the first view as the orientation reference for the generated geometry. Stage 2

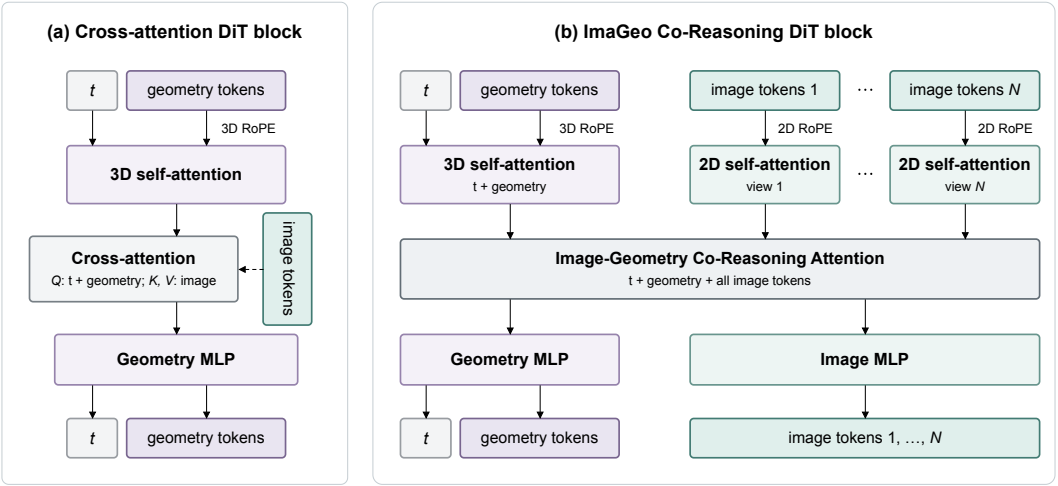

Figure 2: Comparison of DiT blocks. (a) Cross-attention updates geometry tokens using fixed image features. (b) ImaGeo jointly updates image and geometry tokens through modality-specific self-attention and joint attention. MLPs following modality-specific self-attention are omitted for clarity. The shown 3D RoPE is used only in Stage 2.

inherits the geometry coordinate frame from the spatial anchors produced by Stage 1 and uses $\mathbf{r}_i = \mathbf{r}_{\text{shared}}$ for all views. Each parameter set lies in $\mathbb{R}^{5\times d}$. The resulting image tokens $\mathbf{v}_i^\ell$ are updated alongside $\mathbf{g}^\ell$ in each block.

Modality-specific spatial processing. Self-attention first processes geometry tokens and each image's tokens separately:

$$\bar{\mathbf{g}}^\ell = \mathbf{g}^\ell + \mathrm{SA}_{3D}^\ell(\mathrm{LN}(\mathbf{g}^\ell)), \tag{4}$$

$$\bar{\mathbf{v}}_i^\ell = \mathbf{v}_i^\ell + \mathrm{SA}_{2D}^\ell(\mathrm{LN}(\mathbf{v}_i^\ell)), \qquad i = 1, \ldots, N. \tag{5}$$

For Stage 2, SA_{3D} uses 3D RoPE (Su et al., 2021) with the coordinates of the VoxSet tokens; the timestep token is assigned $(0, 0, 0)$. Stage 1 omits 3D RoPE. SA_{2D} is applied independently to each image with shared weights; 2D RoPE uses patch-grid coordinates and applies only to patch tokens, not to the five learnable prefix tokens.

Joint image–geometry updates. Image–Geometry Co-Reasoning Attention (IGCA) then attends over the concatenated timestep, geometry, and image sequences:

$$\mathbf{s}^\ell = [\bar{\mathbf{g}}^\ell; \bar{\mathbf{v}}_1^\ell; \ldots; \bar{\mathbf{v}}_N^\ell], \tag{6}$$

$$\hat{\mathbf{s}}^\ell = \mathbf{s}^\ell + \mathrm{IGCA}^\ell(\mathrm{LN}(\mathbf{s}^\ell)). \tag{7}$$

We split the output into timestep, geometry, and per-view image tokens for the next block. Each modality-specific self-attention step is followed by an independent MLP, and additional MLPs follow joint attention. These MLP updates are omitted from the attention equations for brevity. Updated image tokens can influence subsequent geometry updates through joint attention in later blocks. Only geometry tokens after the final block are projected to the output velocity.

The block shares its per-image processing weights and accepts a variable number of input views. With multiple views, joint attention connects image tokens both across views and with the shared geometry sequence. Appendix A.1 describes the transition between stages and the inference settings.

3.3 Learning through Geometry Generation

We train both stages with a rectified-flow velocity objective (Liu et al., 2023) in their respective 3D latent spaces. Noise is applied to the geometry latents, while image features are updated within the denoiser.

Given Gaussian noise $\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and a clean 3D latent $\mathbf{z}_1$, we sample $t \sim \mathcal{U}(0, 1)$ and construct the linear interpolation $\mathbf{z}_t = (1 - t)\mathbf{z}_0 + t\mathbf{z}_1$. The target velocity is constant along this path, $\mathbf{u}_t = \mathbf{z}_1 - \mathbf{z}_0$, yielding the objective

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{t,\mathbf{z}_0,\mathbf{z}_1,\mathcal{I}} \left[\| f_\theta(\mathbf{z}_t, t, \mathcal{I}) - (\mathbf{z}_1 - \mathbf{z}_0) \|_2^2 \right]. \tag{8}$$

Although supervision is applied only to geometry prediction, gradients propagate through joint attention to train the image-processing layers. No auxiliary point-map, camera-prediction, or image-reconstruction losses are used to train the denoiser. At inference, we integrate the predicted velocity field from $t = 0$ to $t = 1$, undo the latent scaling, and decode the result with the VAE.

For Stage 1, the initial single-view training phase uses point clouds in the canonical pose stored with each asset. During subsequent multi-view training, we rotate each point cloud into alignment with the first input view before VAE encoding. We vary the number and subset of conditioning images, including single-image inputs, while retaining first-view-aligned targets throughout this phase. Target orientation is therefore determined by the training phase rather than the sampled number of views. All configurations use the same velocity-prediction objective.

3.4 Implementation Details

We train on 325K 3D assets selected from Objaverse-XL (Deitke et al., 2023a) using an aesthetic classification model to filter for high quality. Both stages use progressive token-length curricula followed by multi-view training (Table 6 in Appendix A.1). Multi-view training in both stages samples one to six input views per example. At inference, Stage 1 uses token-length extrapolation: it generates $8{,}192$ VecSet tokens despite a maximum training length of $4{,}096$. Appendix A.1 provides the inference settings.

Our cross-attention baselines follow the LATTICE DiT design (Lai et al., 2026). For the Stage 2 architecture comparison, the 1.5B cross-attention baseline has a hidden size of $1{,}536$ and 37 cross-attention DiT blocks. Our 1.5B ImaGeo model uses a hidden size of $1{,}536$ and 21 image–geometry co-reasoning blocks.

3.5 High-Quality 3D Data Synthesis

Single-image conditioning leaves unseen surfaces underconstrained. DetailVerse in Hi3DGen (Ye et al., 2025) converts generated images into meshes using TRELLIS and filters them for quality. We instead condition on complementary views from an image generation model or selected frames of a generated video. These views should preserve object identity and shape across viewpoints. ImaGeo generates coarse and fine geometry from these views, and an image-conditioned Detail Refiner adds local detail. Retaining the views and refined meshes yields paired image–3D data. We demonstrate this pipeline; large-scale dataset construction and downstream training utility remain future work.

Detail refinement. The refiner uses a single conditioning image and operates on the decoded mesh at 1536^3 resolution (Appendix A.2). It is used only for data synthesis and the teaser (Figure 1), and excluded from all quantitative evaluations and qualitative method comparisons. Appendix C details the synthesis procedure.

4 Experiments

4.1 Evaluation Setup

We randomly select 100 objects from Google Scanned Objects (GSO) (Downs et al., 2022) and 100 from Toys4K (Stojanov et al., 2021) for single-view generation and multi-view reconstruction. All quantitative evaluations and qualitative method comparisons use ImaGeo without the Detail Refiner. We evaluate surface structure using normal-map PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018), and use foreground-mask IoU in the architecture ablation. For 3D geometry, CD-L1 and CD-L2 average bidirectional nearest-neighbor Euclidean distances and squared distances, respectively, over sampled point clouds.

4.2 SINGLE-VIEW 3D GENERATION

Evaluation on GSO and Toys4K. Each method generates geometry from a single frontal image without ground-truth coarse conditioning. For evaluation, we normalize each predicted mesh and align its pose using a loss against ground-truth (GT) normal maps with fixed cameras (Appendix A.3). ImaGeo and Pixal3D (TRELLIS.2) retain their input-aligned rotations and optimize translation only; the other methods optimize both rotation and translation. We report normal-map PSNR, SSIM, and LPIPS after this GT-assisted pose alignment. Table 1 compares ImaGeo with TRELLIS.2 (Xiang et al., 2026), Pixal3D (TRELLIS.2) (Li et al., 2026), CUPID (Huang et al., 2026a), Direct3D-S2 (Wu et al., 2025), and UltraShape (Jia et al., 2025). ImaGeo achieves the best scores on all three metrics on both datasets.

Table 1: Single-view generation on GSO and Toys4K (100 objects each). Metrics use GT-normal-assisted pose alignment (Appendix A.3). Best and second-best results are bold and underlined.

	GSO			Toys4K		
Method	N-PSNR ↑	N-SSIM ↑	N-LPIPS ↓	N-PSNR ↑	N-SSIM ↑	N-LPIPS ↓
TRELLIS.2	22.515	0.7917	0.1330	22.782	0.8499	0.1058
Pixal3D (TRELLIS.2)	22.450	0.7714	0.1464	<u>24.386</u>	<u>0.8692</u>	<u>0.0857</u>
CUPID	23.115	0.8173	0.1248	23.303	0.8617	0.0976
Direct3D-S2	20.714	0.7503	0.1715	21.202	0.8085	0.1434
UltraShape	<u>23.732</u>	<u>0.8234</u>	<u>0.1013</u>	23.396	0.8573	0.0949
ImaGeo	**24.518**	**0.8376**	**0.0855**	**24.831**	**0.8863**	**0.0740**

Evaluation on Internet-sourced and AI-generated images. We evaluate on 100 images from Internet and AI-generated sources. The images depict objects from a range of semantic categories with elaborate shapes and fine surface structures. Each method predicts its own coarse geometry. ULIP-2 (Xue et al., 2024) and Uni3D (Zhou et al., 2024) measure similarity between sampled mesh point clouds and the input image in their joint 3D–image embedding spaces. CLIP-N measures cosine similarity between rendered normal maps and the input image using CLIP (Radford et al., 2021) image embeddings. Higher is better for all three scores.

ImaGeo achieves the highest scores across all three embedding-based metrics (Table 2), indicating improved image–shape agreement under this evaluation.

Qualitative results. Figure 3 highlights differences in local surface structure across four examples, including the shield ornament and roof tiles. ImaGeo reproduces these input-visible details more clearly than several baselines. Additional examples are provided in Appendix B.

Table 2: Single-view generation on Internet-sourced and AI-generated images. Best and second-best results are bold and underlined.

Method	ULIP-2 ↑	Uni3D ↑	CLIP-N ↑
TRELLIS.2	0.4597	0.4200	<u>0.6449</u>
Pixal3D (TRELLIS.2)	0.4462	<u>0.4203</u>	0.6312
CUPID	0.4512	0.4163	0.6244
Direct3D-S2	<u>0.4601</u>	0.4175	0.6359
UltraShape	0.4579	0.4163	0.6435
ImaGeo	**0.4644**	**0.4245**	**0.6493**

4.3 MULTI-VIEW 3D RECONSTRUCTION

We render 24 random 1024×1024 views per GSO/Toys4K object. Four ordered input views are fixed across methods; the remaining 20 are used to compute normal-map PSNR, SSIM, and LPIPS. Scores are averaged over views and then over objects. All methods use 24 PCA-based initial poses followed by similarity-transform ICP for alignment to ground truth. The same transform is applied for CD-L1/CD-L2 evaluation and normal-map rendering. Final CD uses fresh samples of 100,000 points per mesh; Appendix A.3 details alignment sampling and the VGGT point-cloud exception.

Comparison with prior methods. Table 3 compares ImaGeo with the multi-image variant of TRELLIS (Xiang et al., 2025), ReconViaGen (Chang et al., 2026), Mix3R (Lin et al., 2026), RecGen3D (Huang et al., 2026b), and VGGT-1B (Wang et al., 2025). For VGGT-1B, we evaluate its direct point-map output and report only CD-L1 and CD-L2. We evaluate ReconViaGen v0.5 and



Figure 3: Qualitative comparison of single-view 3D generation on four examples. Each example pairs a full-object view with enlarged details from the red box. Best viewed zoomed in.

Mix3R-V2. ImaGeo achieves the best results on all reported metrics on both datasets. Compared with RecGen3D, the strongest baseline, the improvements extend across both Chamfer distance and held-out-view normal-map metrics. Together, these results indicate better agreement with the ground-truth geometry in both 3D surface positions and rendered surface normals.

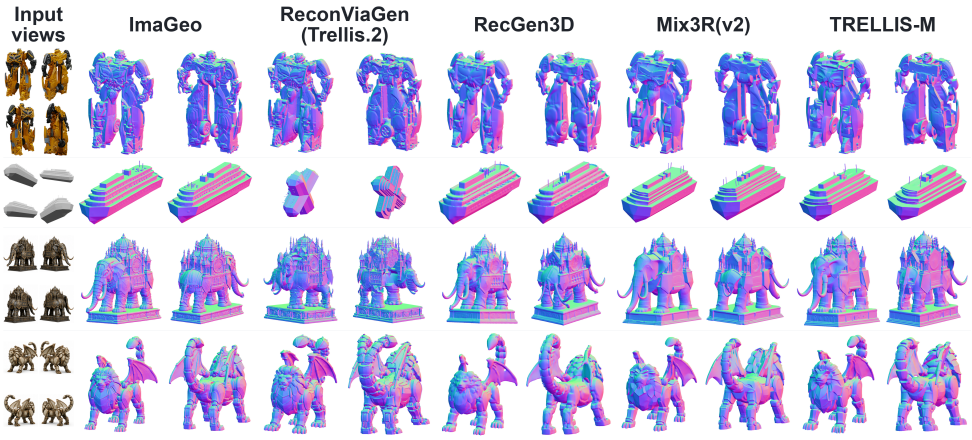

Figure 4: Multi-view reconstruction from rendered (top two rows) and generated (bottom two rows) images. Each example shows four inputs and front/back normal maps from competing methods and ImaGeo. Best viewed zoomed in.

Table 3: Multi-view 3D reconstruction on GSO and Toys4K (100 objects each). Best and second-best results are shown in bold and underlined, respectively.

	GSO					Toys4K				
Method	CD-L1 ↓	CD-L2 ↓	N-PSNR ↑	N-SSIM ↑	N-LPIPS ↓	CD-L1 ↓	CD-L2 ↓	N-PSNR ↑	N-SSIM ↑	N-LPIPS ↓
TRELLIS-M	0.01159	0.000322	19.863	0.7845	0.1439	0.01282	0.000699	19.540	0.7966	0.1508
Mix3R-V2	0.01285	0.000439	19.566	0.7842	0.1495	0.01083	0.000390	20.311	0.8112	0.1279
ReconViaGen v0.5	0.01502	0.000974	19.486	0.7702	0.1594	0.01485	0.000767	18.921	0.7855	0.1623
VGGT-1B	0.01768	0.001479	–	–	–	0.01692	0.001315	–	–	–
RecGen3D	<u>0.00898</u>	<u>0.000166</u>	<u>21.389</u>	<u>0.8092</u>	<u>0.1158</u>	<u>0.00864</u>	<u>0.000223</u>	<u>21.528</u>	<u>0.8328</u>	<u>0.1018</u>
ImaGeo	**0.00855**	**0.000155**	**21.794**	**0.8164**	**0.0923**	**0.00809**	**0.000218**	**22.025**	**0.8460**	**0.0818**

Qualitative results. Figure 4 compares front and back normal-map renderings for four objects. The first two rows use images rendered from 3D models in GSO and Toys4K; the last two use generated multi-view images. ImaGeo preserves local structures such as the robot's body panels and the ship's layered decks. In the illustrated elephant example, it retains the overall shape and surface ornamentation despite inconsistencies among the generated input views.

4.4 Ablation Study

Stage 2 architecture comparison. We compare the 1.5B ImaGeo model with 1.5B and 5B cross-attention DiTs on 40 high-fidelity objects randomly selected from Objaverse (Deitke et al., 2023b) and our private dataset (Table 4), with all training objects strictly excluded. All variants use the same 3D VAE, ground-truth Stage 1 representations, training data, single-view training phase, and inference settings. Their vision encoders use the same architecture and pretrained initialization, but each encoder is jointly trained with its respective denoiser. The reported 1.5B and 5B parameter counts refer only to the denoisers, excluding DINOv2 and the VAE. The 5B baseline has a hidden size of 3,072 and 30 cross-attention DiT blocks, and is trained on 96 NVIDIA A100 GPUs, using the same number of training iterations and the same per-GPU batch size at each curriculum phase as the 1.5B cross-attention baseline (Table 6).

Table 4: Controlled single-view architecture comparison on 40 objects. PSNR, SSIM, and LPIPS are measured on normal maps. Best and second-best results are shown in bold and underlined, respectively.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	IoU ↑
Cross-attention baseline (1.5B)	22.919	0.8949	0.0628	0.9633
Cross-attention baseline (5B)	**23.256**	<u>0.9014</u>	<u>0.0544</u>	**0.9664**
ImaGeo (1.5B)	<u>23.233</u>	**0.9028**	**0.0538**	<u>0.9654</u>

ImaGeo improves every metric over the 1.5B baseline, including a 0.314 dB PSNR gain and a 14.3% reduction in LPIPS. With roughly one third of the parameters, it is within 0.023 dB PSNR and 0.001 IoU of the 5B baseline and achieves slightly higher SSIM and lower LPIPS. Figure 5 shows the corresponding visual comparison.

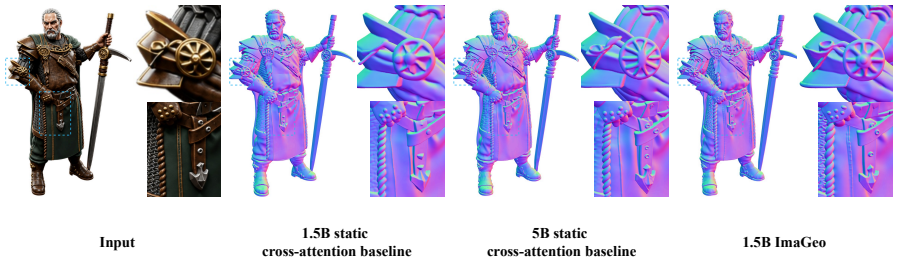

Figure 5: Qualitative comparison of denoiser architectures and model sizes. Our 1.5B model improves over the 1.5B cross-attention baseline, with surface detail comparable to the 5B baseline.

Stage 1 orientation alignment in multi-view reconstruction. Table 5 evaluates Stage 1 output orientation using final Stage 2 meshes on GSO/Toys4K. Both variants use identical input views, training budgets, inference settings, and the same Stage 2 checkpoint. Output orientation is distinct from the similarity alignment used for evaluation. First-view alignment improves all five metrics, reducing CD-L1 by 32.1% on GSO and 35.9% on Toys4K.

Table 5: Stage 1 output-orientation ablation on GSO and Toys4K (100 objects each, four views). Metrics use final Stage 2 meshes. PSNR, SSIM, and LPIPS use normal maps; best results are bold.

	GSO					Toys4K				
3D orientation	CD-L1 ↓	CD-L2 ↓	PSNR ↑	SSIM ↑	LPIPS ↓	CD-L1 ↓	CD-L2 ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Canonical pose	0.01260	0.000328	19.655	0.7819	0.1388	0.01261	0.000632	20.308	0.8086	0.1322
First-view-aligned pose	**0.00855**	**0.000155**	**21.794**	**0.8164**	**0.0923**	**0.00809**	**0.000218**	**22.025**	**0.8460**	**0.0818**

Number of input views. Appendix B.1 evaluates one, two, four, and six input views. CD-L1, CD-L2, and all three normal-map metrics improve with increasing view count on both datasets.

5 Discussion and Limitations

Our experiments show that jointly updating image and geometry features improves 3D generation quality. Joint attention integrates complementary views through cross-view and bidirectional image–geometry interactions. These interactions are learned solely through geometry flow matching, without a dedicated reconstruction branch or auxiliary geometric prediction losses.

The model can produce duplicated parts and spurious interior geometry (Appendix B.3). Inconsistent input views and limited observations of interior regions may contribute to these errors. Joint attention processes longer sequences as input views are added, and its runtime and memory scaling require further evaluation. The data-synthesis pipeline is demonstrated through examples, while its downstream training utility remains untested.

6 Conclusion

We presented ImaGeo, a coarse-to-fine diffusion transformer combining per-view processing with joint image–geometry attention. Trained through geometry flow matching, both denoisers accept variable-view inputs without camera parameters or auxiliary reconstruction objectives. Controlled comparisons support the conditioning design at a fixed parameter budget; task evaluations show improved single-view image–shape similarity and sparse-view reconstruction accuracy. We also demonstrate paired image–3D synthesis using generated views and an additional detail refiner.

AI USE STATEMENT

We used generative AI tools to polish and translate the manuscript, assist in creating scientific figures, and generate some of the input images used for 3D generation in this work. The authors reviewed the polished and translated text sentence by sentence to ensure that the technical meaning, terminology, numerical values, and conclusions were preserved. We checked the structures, labels, and workflows in AI-assisted figures for consistency with the actual methods. We also manually inspected and selected AI-generated input images for suitability for their intended experimental use. The authors take full responsibility for the final content of this paper, including all AI-assisted text, figures, and other artifacts.

ETHICS STATEMENT

This work studies image-conditioned 3D generation and reconstruction. Potential concerns include unauthorized reproduction of protected designs and the use of generated assets in misleading content. Generated geometry may contain plausible but inaccurate details, particularly in regions not observed in the input images, and should not be treated as a verified representation of a real object. Synthetic image–3D data may also inherit biases and artifacts from the image generation models. Downstream use should respect applicable data and asset licenses and include validation appropriate to the intended application.

REPRODUCIBILITY STATEMENT

Section 3 describes the model architecture, training objective, and implementation details. Appendix A.1 provides the training schedules, hardware configurations, image conditioning, and two-stage inference settings. Section 4 describes the evaluation datasets, metrics, and comparison protocols. Appendix A.3 details the pose-alignment and geometric evaluation procedures. Appendix A.2 documents the Detail Refiner, while Appendix C describes the image–3D data synthesis pipeline. The Detail Refiner is excluded from all quantitative evaluations and qualitative method comparisons, as specified in the experimental setup. We will publicly release the source code and model weights upon acceptance of the paper.

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*, volume 2026, pp. 90387–90401, 2026.

Hansheng Chen, Bokui Shen, Yulin Liu, Ruoxi Shi, Linqi Zhou, Connor Z Lin, Jiayuan Gu, Hao Su, Gordon Wetzstein, and Leonidas Guibas. 3D-Adapter: Geometry-consistent multi-view diffusion for high-quality 3D generation. *arXiv preprint arXiv:2410.18974*, 2024.

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-XL: A universe of 10m+ 3D objects. *Advances in Neural Information Processing Systems*, 36:35799–35813, 2023a.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3D objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13142–13153, 2023b.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3D scanned household items. In *2022 international conference on robotics and automation (ICRA)*, pp. 2553–2560. IEEE, 2022.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Xianglong He, Zi-Xin Zou, Chia-Hao Chen, Yuan-Chen Guo, Ding Liang, Chun Yuan, Wanli Ouyang, Yan-Pei Cao, and Yangguang Li. SparseFlex: High-resolution and arbitrary-topology 3D shape modeling. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14822–14833. IEEE, 2025.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3D. In *International Conference on Learning Representations*, volume 2024, pp. 50678–50702, 2024.

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*, pp. 12741–12752, 2026a.

Zhisheng Huang, Jiahao Chen, Cheng Lin, Chenyu Hu, Hanzhuo Huang, Zhengming Yu, Mengfei Li, Yuheng Liu, Zekai Gu, Zibo Zhao, Yuan Liu, Xin Li, and Wenping Wang. RecGen3D: Reconstruction-guided 3D generation in a shared canonical space. In *SIGGRAPH Asia 2026 Conference Papers*, 2026b.

Team Hunyuan3D, Shuhui Yang, Mingxin Yang, Yifei Feng, Xin Huang, Sheng Zhang, Zebin He, Di Luo, Haolin Liu, Yunfei Zhao, et al. Hunyuan3D 2.1: From images to high-fidelity 3D assets with production-ready PBR material. *arXiv preprint arXiv:2506.15442*, 2025.

Tanghui Jia, Dongyu Yan, Dehao Hao, Yang Li, Kaiyi Zhang, Xianyi He, Lanjiong Li, Yuhan Wang, Jinnan Chen, Lutao Jiang, et al. UltraShape 1.0: High-fidelity 3D shape generation via scalable geometric refinement. *arXiv preprint arXiv:2512.21185*, 2025.

Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu, Qingxiang Lin, Jingwei Huang, Chunchao Guo, and Xiangyu Yue. LATTICE: Democratize high-fidelity 3D generation at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19982–19992, 2026.

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 *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–12, 2026.

Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. TripoSG: High-fidelity 3D shape synthesis using large-scale rectified flow models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025a.

Zhihao Li, Yufei Wang, Heliang Zheng, Yihao Luo, and Bihan Wen. Sparc3D: Sparse representation and construction for high-resolution 3D shapes modeling. *Advances in Neural Information Processing Systems*, 38:118582–118600, 2025b.

Siyou Lin, Zhou Xue, Hongwen Zhang, Liang An, Dongping Li, Shaohui Jiao, and Yebin Liu. Mix3R: Mixing feed-forward reconstruction and generative 3D priors for joint multi-view aligned 3D reconstruction and pose estimation. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–12, 2026.

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*, 2023.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205. IEEE, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Tianchang Shen, Jacob Munkberg, Jon Hasselgren, Kangxue Yin, Zian Wang, Wenzheng Chen, Zan Gojcic, Sanja Fidler, Nicholas Sharp, and Jun Gao. Flexible isosurface extraction for gradient-based mesh optimization. *ACM Transactions on Graphics*, 42(4):37:1–37:16, 2023. doi: 10.1145/3592430.

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

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025.

Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. PF-LRM: Pose-free large reconstruction model for joint pose and shape prediction. In *International Conference on Learning Representations*, volume 2024, pp. 18187–18208, 2024a.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709. IEEE, 2024b.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, volume 2026, pp. 10481–10497, 2026.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

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. *Advances in Neural Information Processing Systems*, 37:121859–121881, 2024.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, and Yao Yao. Direct3D-S2: Gigascale 3D generation made easy with spatial sparse attention. *Advances in Neural Information Processing Systems*, 38:170778–170804, 2025.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480. IEEE, 2025.

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, et al. Native and compact structured latents for 3D generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14419–14429, 2026.

Le Xue, Ning Yu, Shu Zhang, Artemis Panagopoulou, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, et al. ULIP-2: Towards scalable multimodal pre-training for 3D understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27091–27101. IEEE, 2024.

Chongjie Ye, Yushuang Wu, Ziteng Lu, Jiahao Chang, Xiaoyang Guo, Jiaqing Zhou, Hao Zhao, and Xiaoguang Han. Hi3DGen: High-fidelity 3D geometry generation from images via normal bridging. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25050–25061. IEEE, 2025.

Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3DShape2VecSet: A 3D shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)*, 42(4):1–16, 2023.

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.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

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

Peng Zheng, Dehong Gao, Deng-Ping Fan, Li Liu, Jorma Laaksonen, Wanli Ouyang, and Nicu Sebe. Bilateral reference for high-resolution dichotomous image segmentation. *arXiv preprint arXiv:2401.03407*, 2024.

Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3D: Exploring unified 3D representation at scale. In *International Conference on Learning Representations*, volume 2024, pp. 46766–46782, 2024.

A Additional Implementation Details

A.1 Co-Reasoning Backbone

Training schedule. Table 6 summarizes the progressive token-length curricula, training iterations, and hardware for both stages and the cross-attention baseline.

We train both stages of the co-reasoning backbone using Adam. Table 6 reports the learning rates and per-GPU batch sizes for the specified training phases. The backbone and cross-attention baselines are trained without gradient accumulation, so the global batch size equals the per-GPU batch size multiplied by the number of GPUs. At each phase of the Stage 2 architecture comparison, the 5B baseline therefore uses a global batch size six times that of the 1.5B models, with the same number of training iterations.

Table 6: Training curriculum and optimization settings for ImaGeo. Batch sizes are per GPU. Stage 1 uses canonical targets during single-view training and first-view-aligned targets during multi-view training.

Training phase	Tokens	Iterations	Learning rate	Batch/GPU	Hardware
Stage 1: single-view	1,024	80K	1×10^{-4}	16	16×A100
	2,048	60K	6×10^{-5}	12	
	4,096	20K	4×10^{-5}	8	
Stage 1: multi-view	4,096	100K	4×10^{-5}	8	16×A100
Stage 2: base curriculum	1,024	40K	1×10^{-4}	16	16×A100
	2,048	20K	8×10^{-5}	12	
	4,096	20K	6×10^{-5}	8	
	8,192	20K	5×10^{-5}	4	
	16,384	20K	2×10^{-5}	2	
	32,768	10K	1×10^{-5}	1	
Stage 2: multi-view	32,768	6K	1×10^{-5}	1	16×A100

The 1.5B and 5B cross-attention Stage 2 baselines follow the Stage 2 base curriculum, with the same token lengths, training iterations, and per-GPU batch sizes at each phase. They use 16×A100 and 96×A100 GPUs, respectively.

Joint attention. IGCA shares the query, key, and value projection weights across image and geometry modalities and applies no positional encoding within joint attention.

Image conditioning. The backbone uses a DINOv2 (Oquab et al., 2023) ViT-g/14 image encoder. We use patch features without the class token and train the image encoder along with the denoiser.

Two-stage inference. Stage 1 generates $8{,}192$ VecSet tokens with 50 Euler sampling steps, conditioned on 518×518 input images. This intentionally extrapolates beyond the maximum training sequence length of $4{,}096$ tokens. We decode the tokens into a coarse mesh, sample its surface, and voxelize the sampled positions to obtain the spatial anchors for Stage 2. Conditioned on these anchors and the same input images at 1022×1022 resolution, Stage 2 generates up to $32{,}768$ VoxSet tokens, with the actual sequence length determined by the number of anchors. It uses 20 Euler sampling steps and decodes the final mesh. Both stages use the same input views and a DINOv2 ViT-g/14 encoder, yielding $1{,}369$ and $5{,}329$ patch tokens per view in Stage 1 and Stage 2, respectively, excluding the learnable prefix tokens. We use classifier-free guidance scales of 3 and 5 for Stage 1 and Stage 2, respectively, for all evaluated view counts.

A.2 Image-Conditioned Detail Refinement

The Detail Refiner is used for the teaser in Figure 1 and the data-synthesis application in Section 3.5. All quantitative evaluations and qualitative method comparisons exclude this refinement step. The refiner VAE uses a sparse-grid design similar to TripoSF (SparseFlex) (He et al., 2025) and Sparc3D (Li et al., 2025b).

Geometry and image encoding. We normalize the mesh to $[-0.475, 0.475]^3$ and sample near-surface signed-distance values and normals on a sparse 1536^3 grid. Distances are divided by $1.8/1536$ and clipped to $[-1, 1]$ before being concatenated with the three normal components. The VAE downsamples by a factor of 8 along each spatial axis, yielding 32-channel tokens on a sparse 192^3 latent grid. Its encoder and decoder each use 12 transformer blocks with hidden width 768 and 8 attention heads. The conditioning image is foreground-masked, recentered, and processed at 1022×1022 resolution by a DINOv2 (Oquab et al., 2023) ViT-L/14 encoder. We use the input alpha mask when available and estimate one with BiRefNet (Zheng et al., 2024) otherwise.

Sparse VAE training. To limit GPU memory use, we render 256×256 position maps of the ground-truth mesh from six random cameras, map visible surface positions to latent-grid cells, and expand the selected cells with a $5 \times 5 \times 5$ neighborhood. Only sparse SDF inputs in these regions enter the VAE. If their count exceeds the budget of two million sparse cells, we adjust the cameras and repeat the selection.

The decoder predicts a sparse signed-distance field and the deformation and interpolation parameters used by FlexiCubes (Shen et al., 2023) to extract a differentiable mesh. We supervise SDF values with an L1 loss and pruning decisions with binary cross-entropy against target sparse-voxel masks. We render the decoded and target meshes from the same cameras and apply L1 losses to their normal, depth, and silhouette maps. The objective also includes posterior KL regularization. The weights for SDF, pruning, normal, depth, silhouette, and KL terms are 0.1, 0.05, 5, 4, 2, and 0.001, respectively. We train the VAE on 1024^3 grids with AdamW, a learning rate of 2×10^{-5}, and weight decay 10^{-2}.

Refinement DiT. The denoiser has 21 transformer blocks, hidden width 2048, and 16 attention heads. Geometry self-attention alternates between unshifted and shifted windows of size 64, with a shift of 32 on the latent grid. Each block then applies cross-attention to the image features and a feed-forward layer. Unlike the co-reasoning backbone, this denoiser keeps image features fixed across blocks.

DiT training pairs and objective. We obtain coarse meshes by reconstructing training objects with the $4{,}096$-token VoxSet VAE. For each training pair, ground-truth geometry determines the latent-block coordinates. Within these GT-defined blocks, we resample the coarse mesh to obtain its SDF values and normals, retaining cubes even where the coarse surface does not intersect the corresponding GT block. We encode the coarse and ground-truth inputs with the refiner VAE and store the resulting latents in the same coordinate order. This yields paired latents $\mathbf{z}_c$ and $\mathbf{z}_{\mathrm{gt}}$ with matching spatial support and token correspondence for the interpolation below. We perturb $\mathbf{z}_c$ with spatially varying DDPM noise to obtain $\tilde{\mathbf{z}}_c$: selected spatial blocks use timestep 600 and the remaining blocks use timestep 300. The refinement objective is

$$\mathbf{z}_t = (1 - t)\tilde{\mathbf{z}}_c + t\mathbf{z}_{\mathrm{gt}}, \tag{9}$$

$$\mathcal{L}_{\mathrm{refine}} = \mathbb{E}\Big[\|f_\theta(\mathbf{z}_t, t, I) - (\mathbf{z}_{\mathrm{gt}} - \tilde{\mathbf{z}}_c)\|_2^2\Big]. \tag{10}$$

Here I is the conditioning image. We use AdamW with learning rate 3×10^{-5}, weight decay 10^{-2}, and conditioning dropout probability 0.1.

Inference and mesh extraction. We add Gaussian noise to the encoded coarse mesh at DDPM timestep 300, then apply five Euler steps without classifier-free guidance, keeping sparse latent coordinates fixed. FlexiCubes decodes the refined latents at resolution 1536^3, followed by quadric-error decimation with a target of two million faces, preserving boundaries and topology. For multi-view reconstruction, the refiner still uses a single conditioning image.

A.3 Geometric Evaluation Details

Single-view evaluation. Table 1 evaluates complete mesh generation from a single input image. The known cameras are used for evaluation-time pose fitting and rendering, not as inputs to ImaGeo. For every method, we apply its fixed coordinate-axis convention, remove small disconnected fragments, center the predicted mesh at its own bounding-box center, and normalize its longest bounding-box side to one. These steps use no ground-truth mesh. A native pose in this protocol means the standardized pose, not the original file coordinates.

Pose fitting uses a normal loss against the ground-truth frontal normal map. ImaGeo and Pixal3D (TRELLIS.2) already produce geometry aligned with the input image, so we fix their rotations and optimize only the three translation parameters. For the other methods, we optimize three rotation and three translation parameters. Camera intrinsics and extrinsics, mesh shape, and normalized scale remain fixed; translation along the camera depth axis can change the projected size.

After fixing the fitted pose, we render the frontal normal map and compute PSNR, SSIM, and LPIPS against the ground-truth reference. Ground-truth normals are used for evaluation-time pose fitting, not for geometry generation. The reported scores therefore measure frontal normal-map agreement after GT-assisted pose alignment; they do not measure the accuracy of the original output pose.

Multi-view reconstruction evaluation. Alignment and final CD evaluation use separate point sets. For mesh predictions, alignment uses 2,000 points sampled from each of the predicted and ground-truth surfaces, with 24 PCA-based initial poses followed by similarity-transform ICP. After alignment, we freshly sample 100,000 points from each surface to compute CD-L1 and CD-L2, rather than reusing the alignment samples. The same alignment transform is applied to the predicted mesh for normal-map rendering and evaluation.

VGGT-1B produces a point cloud rather than a mesh, so its prediction cannot be resampled from a mesh surface. Final CD evaluation uses up to 100,000 existing predicted points. Our exported VGGT-1B point clouds contain at most 50,000 points, so all retained prediction points are used. The ground-truth mesh is still freshly sampled with 100,000 surface points.

B More Results

B.1 Number of Input Views

We evaluate ImaGeo on 100 objects each from GSO and Toys4K with one, two, four, and six input views (Table 7). For each object, the input-view sets are nested: $\mathcal{V}_1 \subset \mathcal{V}_2 \subset \mathcal{V}_4 \subset \mathcal{V}_6$, where $\mathcal{V}_k$ contains k input views. We follow the same alignment and evaluation protocol as in the main multi-view experiments, but use different input views from those in Table 3. Consequently, the four-view results in Table 7 are obtained with a different conditioning view set and are not directly comparable to the four-view results in Table 3. CD-L1, CD-L2, and all three normal-map metrics improve with each increase in view count on both datasets. From one to six views, CD-L1 decreases by 52.0% on GSO and 48.6% on Toys4K. Overall, the results show that ImaGeo benefits from additional observations.

Table 7: Effect of the number of input views on GSO and Toys4K (100 objects per dataset). PSNR, SSIM, and LPIPS are measured on normal maps. Best results within each dataset are shown in bold.

	GSO					**Toys4K**				
Views	CD-L1 ↓	CD-L2 ↓	PSNR ↑	SSIM ↑	LPIPS ↓	CD-L1 ↓	CD-L2 ↓	PSNR ↑	SSIM ↑	LPIPS ↓
1	0.015534	0.000647	19.5372	0.77966	0.17003	0.014084	0.000599	19.6176	0.80201	0.15321
2	0.010179	0.000236	21.1741	0.80896	0.11639	0.010180	0.000342	21.1033	0.82764	0.11097
4	0.008358	0.000171	22.1055	0.82554	0.09346	0.007995	0.000224	22.2215	0.84898	0.08235
6	**0.007456**	**0.000117**	**22.5798**	**0.83577**	**0.08155**	**0.007245**	**0.000194**	**22.8278**	**0.85894**	**0.07459**

B.2 Additional Single-View Comparisons

Figure 6 provides six additional single-view comparisons, covering mechanical characters, surface ornament, and thin structures such as feathers and weapons.

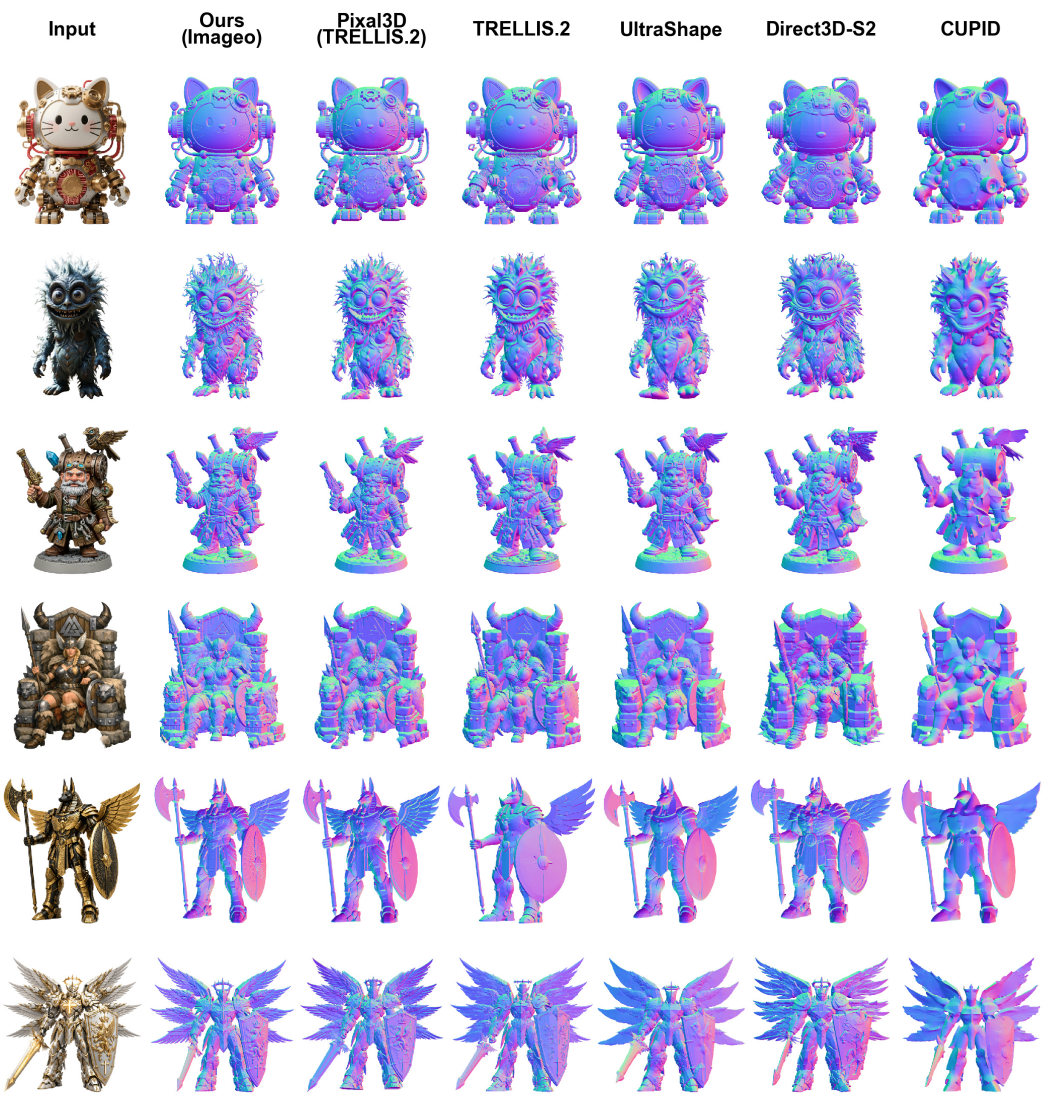

Figure 6: Additional qualitative comparisons of single-view 3D generation on six examples. Each row shows the input image and normal-map renderings from ImaGeo, Pixal3D (TRELLIS.2), TRELLIS.2, UltraShape, Direct3D-S2, and CUPID. ImaGeo results do not use the Detail Refiner. Best viewed zoomed in.

B.3 Failure Case Analysis

Figure 7 shows two failure modes. In (a), inconsistencies among the generated input views may cause the model to reconstruct conflicting tail appearances as separate structures. In (b), limited observations of the box interior may leave empty space insufficiently constrained, resulting in spurious internal geometry. These are plausible explanations based on visual inspection; their precise causes remain to be verified.

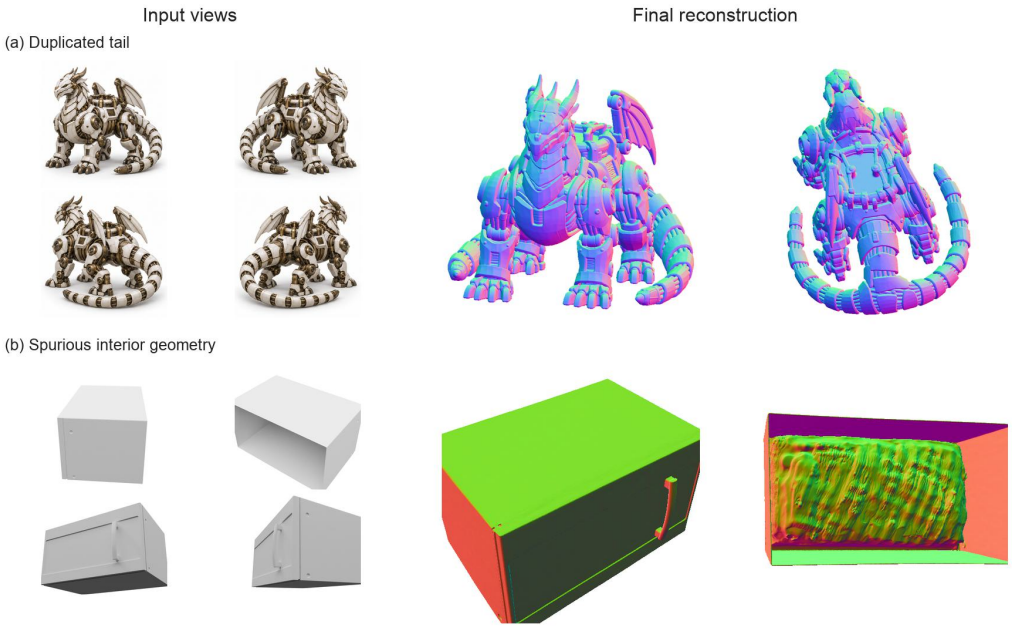

Figure 7: Failure cases in multi-view reconstruction. Each row shows four input views (left) and the Stage 2 reconstruction without the Detail Refiner (right). (a) A duplicated tail in the reconstructed dragon. (b) Spurious interior geometry in the reconstructed box, revealed by the cutaway view on the far right.

C HIGH-QUALITY 3D DATA SYNTHESIS

Multi-view input generation. We use GPT Image 2.5 to generate the multi-view inputs for the data-synthesis pipeline in Section 3.5. The generated views depict the same object from different viewpoints and are intended to preserve its identity and shape while providing complementary observations of its surfaces.

Roles of the three stages. Figure 8 illustrates the progression from coarse geometry to fine surface detail. The generated image set jointly conditions both stages of ImaGeo. **Stage 1** generates VecSet latents that establish the overall shape, part layout, and silhouette. Its decoded mesh provides the spatial anchors for the next stage. **Stage 2** generates VoxSet latents conditioned on these anchors and the input views, resolving finer geometric structures within the coarse shape. This is the output used in our quantitative evaluations and qualitative method comparisons. **The Detail Refiner** further processes the Stage 2 mesh with a single conditioning image, refining local surface relief at 1536^3 resolution (Appendix A.2). In the illustrated example, the full views show the overall shape retained across stages, while the enlarged views highlight changes in the surface ornamentation. These intermediate outputs illustrate the roles of the stages rather than a quantitative ablation of their individual contributions. Retaining the generated input views alongside the refined mesh yields paired image–3D data.



Figure 8: Progressive geometry generation for data synthesis. Stage 1 establishes the overall shape; Stage 2 generates finer geometry using spatial anchors from the coarse mesh; the image-conditioned Detail Refiner further refines local surface relief. Full-object views and enlarged regions show the evolution of shape and surface detail. The Detail Refiner is used for data synthesis and the teaser only.

This application presents a pipeline for future large-scale data synthesis; we do not construct or release a large-scale dataset in this work.

Synthesis examples. Figure 9 presents 32 examples of paired multi-view images and synthesized 3D geometry, spanning decorative objects, animals, architecture, and mechanical designs. Each group shows four input views and two normal-map renderings of the resulting mesh. All examples include the Detail Refiner described in Appendix A.2.



Figure 9: Examples of high-quality 3D data synthesis with ImaGeo and the Detail Refiner. Each of the 32 groups contains four generated input views (left) and two normal-map renderings of the synthesized geometry (right). The examples cover diverse object categories and surface structures. Best viewed zoomed in.