NOFACE3D: NOISE-SCHEDULED FOCUSED ANCHORING AND CONTROLLED ENHANCEMENT FOR 3D GENERATION

Anonymous authors
Paper under double-blind review

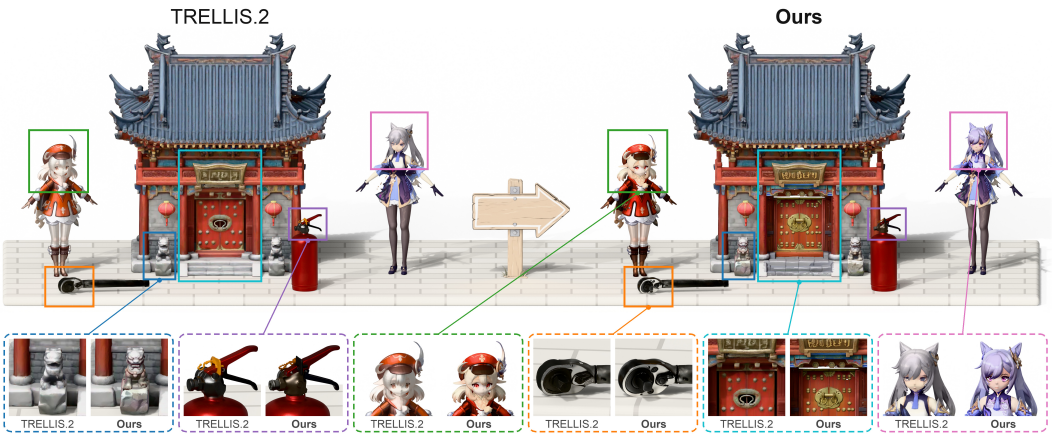

Figure 1: **NOFACE3D: training-free localized refinement of image-to-3D assets.** Left: a scene of assets, each generated from a single image by TRELLIS.2. Right: the same assets after our method refines the boxed regions, preserving the rest of each asset. Dashed panels show enlarged comparisons: the base generation leaves fine details soft, while ours recovers them with sharp geometry and texture, with no extra training or supervision.

ABSTRACT

Recent image-to-3D models generate complete, high-quality assets from a single image, with global shape and appearance that are already convincing; what remains is the fine scale, where small, thin, elongated, or detail-rich regions are reconstructed softly and under-resolved. This gap can arise in generators with fixed-resolution spatial grids: the same spatial sampling density is used for a large, smooth, low-curvature surface and for a thin, high-curvature structure that carries most of the perceived detail. To this end, we introduce NOFACE3D, a training-free framework that refines these detail-critical regions while preserving the rest of the asset. Given a generated mesh, its input image, and a refinement box, NOFACE3D extracts the selected submesh, scales it up to fill the generator's input volume, and re-encodes it into the full-resolution latent grid. Conditioned on the image crop, it refines the region through flow-based sampling with a spatially varying noise schedule that anchors boundary latents. The refined region is then integrated back into the original asset, preserving the surrounding geometry and appearance. We implement NOFACE3D on TRELLIS.2 and its sparse voxel representation; all of it operates purely at inference time on the frozen pretrained generator. We evaluate NOFACE3D on PAniC-3D, TexVerse and BIGDETAIL, a new benchmark of 339 assets with annotated detail-critical regions. On all three, it generates sharper geometry and finer texture detail than the baselines.

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.

1 INTRODUCTION

Single-image 3D generation now produces complete, textured assets in seconds. Feed-forward reconstruction models predict 3D representations directly from image features (Hong et al., 2024; Tang et al., 2024; Xu et al., 2024), while native 3D latent generative models trained on large asset collections (Deitke et al., 2023) synthesize objects with convincing proportions, silhouettes, and appearance (Zhang et al., 2024; Li et al., 2025b; Zhao et al., 2025). Structured-latent generators such as TRELLIS and TRELLIS.2 (Xiang et al., 2025b;a) attach a local code to each occupied voxel of a sparse grid. Their output is already convincing on large, low-curvature surfaces.

However, fine details such as fingers, springs, and engravings often appear soft or blurred. This gap can arise from how spatial capacity is allocated: a fixed-resolution grid provides the same sampling density to large, smooth surfaces and small, high-curvature structures. For example, one full-body character generated by TRELLIS.2 (Xiang et al., 2025a) occupies approximately $4{,}000$ voxels in the 64^3 shape-latent grid (1.5% of its cells), with only about $1{,}400$ voxels (0.5%) representing the head crop. The challenge is therefore to give detail-critical regions more spatial capacity while preserving the parts that are already well reconstructed.

Increasing the global grid resolution offers finer spatial sampling, but also increases computational cost across the entire asset, even when only a small region needs refinement (Wu et al., 2025). Refiners trained on synthetic mesh degradation may not fully address detail lost through limited spatial sampling during generation (Deng et al., 2024). Render-space enhancement can improve appearance, but does not directly increase the latent capacity allocated to a selected 3D region (Ryu et al., 2025).

To address this challenge, we propose NOFACE3D, a training-free framework built on a frozen image-to-3D generator. We extract the target submesh, rescale it to the generator's input cube, and re-encode it with the pretrained encoders. The generator's full-resolution latent grid now represents this region alone, increasing its effective resolution in the original asset's coordinates. The input image crop guides regeneration.

A spatially varying noise schedule anchors latents near attachment boundaries, with stronger regeneration farther away. Finally, we map the refined region back to the original coordinate frame and combine it with the surrounding mesh.

Our evaluation focuses on whether refinement produces sharper geometry and finer texture detail while preserving overall asset quality. We evaluate NOFACE3D on PAniC-3D and TexVerse, and introduce BIGDETAIL to assess localized refinement on diverse objects with detail-critical regions annotated in 2D and 3D. Where artist ground truth is available, we report appearance metrics and pointwise geometry measures. We interpret Chamfer distance and F-score as geometry checks, since they are dominated by large, smooth surfaces (Wang et al., 2018; Chen et al., 2025) and may not reflect fine detail that artist references carry in textures rather than geometry. To assess visible improvements in geometry and texture, we use qualitative comparisons and GPTEval3D evaluations across all three datasets, alongside a human preference study on PAniC-3D. Across these evaluations, NOFACE3D outperforms the original TRELLIS.2 results and the refinement and editing baselines.

Our contributions are as follows.

- NOFACE3D, a training-free framework for localized refinement that scales up a selected region to fill the generator's input volume and re-encodes it into the full-resolution latent grid.
- A spatially varying noise schedule for flow-based sampling that anchors boundary latents and increases refinement strength with distance from attachment boundaries.
- BIGDETAIL, a benchmark of 339 assets reconstructed from generated images, with detail-critical regions annotated in both 2D and 3D. Qualitative comparisons and GPTEval3D evaluations on PAniC-3D, TexVerse, and BIGDETAIL demonstrate sharper geometry and finer texture detail.

2 Related Work

2.1 Text- and image-to-3D generation

Early text-to-3D methods lift 2D diffusion priors into 3D by score distillation, optimizing a per-scene representation so its renderings match a pretrained image model (Poole et al., 2023; Lin et al., 2023; Wang et al., 2023; Chen et al., 2023a); they are flexible but slow, prone to view inconsistency, and tend toward oversmoothed geometry, which multi-view diffusion mitigates by generating several consistent views before lifting (Liu et al., 2023a; Shi et al., 2024; Long et al., 2024). Feed-forward large reconstruction models instead predict a 3D representation in a single pass, trading per-scene optimization for speed (Hong et al., 2024; Li et al., 2024; Tang et al., 2024; Xu et al., 2024). The now dominant family trains native 3D latent generators on large asset collections, using compact vector-set latents (Zhang et al., 2023; Zhao et al., 2023; Zhang et al., 2024; Li et al., 2025b; Zhao et al., 2025) or structured sparse-voxel latents (Xiang et al., 2025b;a). Because dense volumetric capacity scales cubically, recent work attacks this cost with sparse-volume attention (Wu et al., 2025) and non-uniform representations that place resolution where occupancy is (Xiong et al., 2025; Ren et al., 2024), while part-based generators disperse capacity across semantic parts so each is generated at higher effective resolution (Lin et al., 2025; Yang et al., 2025a;b; Sun et al., 2026). A parallel autoregressive line generates meshes as triangle token sequences, targeting artist-like topology directly (Siddiqui et al., 2024; Chen et al., 2024; Tang et al., 2025). NOFACE3D shares the motivation of the capacity-aware methods, but instead of redesigning the representation or retraining, it reallocates the existing budget at inference time to a single semantic region.

2.2 Mesh and geometry refinement

A dedicated line enhances the geometry of an existing coarse asset. DetailGen3D models coarse-to-fine enhancement as a data-dependent latent flow with token matching, trained on synthetically degraded pairs (Deng et al., 2024). Elevate3D refines view by view, alternating a high-frequency-preserving SDEdit (Meng et al., 2022) variant for texture with monocular-geometry updates for shape, explicitly targeting the trade-off a single global noise level imposes (Ryu et al., 2025). MeshReGen unifies enhancement, reconstruction, and editing as one regeneration problem and reports that re-encoding degraded geometry into a structured latent and resampling does not recover fine detail (Park et al., 2026). A related line bridges through predicted normals to raise detail fidelity before geometry synthesis (Ye et al., 2025a), while no-ground-truth surface metrics (Lee et al., 2026) and region-annotated character benchmarks (Chen et al., 2023b) support scoring such refinements. NOFACE3D shares this goal but is training-free and localized: it learns no degradation-specific mapping, operates on a frozen generator, and refines one region at increased effective resolution while leaving the geometry outside it essentially unchanged.

3 Method

NOFACE3D takes a generated asset, its conditioning image, and a bounding box marking a detail-critical region, and regenerates only that region with the frozen generator. The region is scaled up to fill the generator's input volume and re-encoded into its full-resolution latent grid (Section 3.2); a boundary-distance field controls the noise strength and release time of each latent token (Section 3.3); geometry and texture are resampled under that field (Sections 3.4, 3.5); and the result is cut back into the asset in mesh space (Section 3.6). Implementation details and hyperparameters for every step are in Appendix A.

3.1 Preliminaries

We build on TRELLIS.2 (Xiang et al., 2025a), which first predicts a sparse structure (SS) of active voxels, then generates separate structured latents (SLats) for shape and texture. The asset is represented in a fixed cube $\mathcal{C} = [-\frac{1}{2}, \frac{1}{2}]^3$, and generation uses flow models conditioned on DINOv3 (Siméoni et al., 2025) features of one image. Our inference pipeline comprises four sampling stages: sparse structure (SS), low-resolution shape, high-resolution shape, and texture. The SS model samples a dense 16^3 latent that decodes to coarse occupancy; shape SLats are then sampled on a sparse 32^3 support and a sparse 64^3 grid, and a texture SLat on the same 64^3 coordinates. The decoders lift

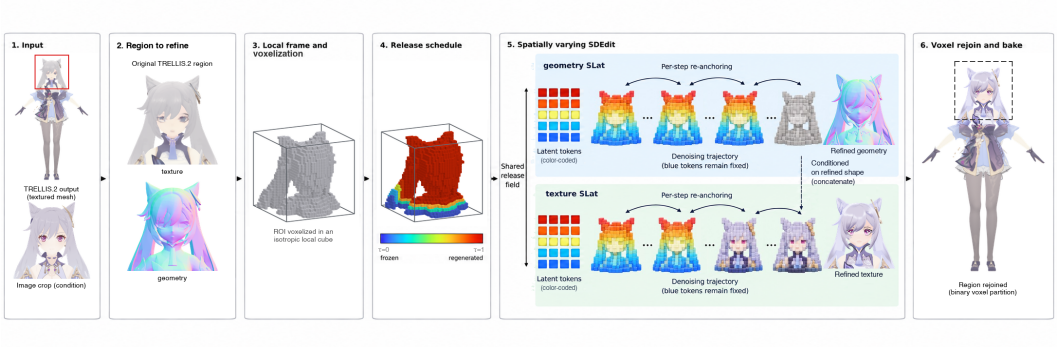

Figure 2: **Overview of NOFACE3D.** Given an asset generated by an off-the-shelf image-to-3D model and a user-marked region, we scale up the region to fill the generator's input volume and re-encode it into the full-resolution latent grid, assign each token a release level τ from a boundary-distance field (anchored near attachment boundaries, with stronger regeneration farther away), and resample geometry and texture with spatially varying SDEdit (Meng et al., 2022), re-anchoring unreleased tokens to the base asset at every step. The refined mesh replaces the base faces inside the crop box.

the shape and texture SLats to geometry and PBR attributes on 1024^3 voxels, from which a textured mesh is extracted. Each stage follows the flow-matching path (Lipman et al., 2023; Liu et al., 2023b)

$$x_t = (1-t)\,x_0 + \big(\sigma_{\min} + (1-\sigma_{\min})\,t\big)\,\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{1}$$

where x_0 is a clean latent, $t \in [0,1]$ controls the noise level, and $\sigma_{\min}$ is the residual noise scale at $t = 0$. At $t = 1$, x_t is pure noise. Ordinary generation integrates the learned velocity v_θ from $t = 1$ to 0 using Euler steps. We reuse the pretrained models in a cascaded inference pipeline to refine the selected region after rescaling and re-encoding it.

3.2 Region Selection and Local Re-encoding

The refinement region is specified by a 3D bounding box B, with a corresponding 2D bounding box Ω in the conditioning image. The 3D box can be supplied directly, as in BigDetail, or derived from the image region. We extract the submesh selected by B without splitting triangles, then center it and scale it uniformly to fit the generator's input cube, preserving its proportions.

Only this rescaled submesh is converted to the generator's 1024^3 sparse voxel representation and re-encoded with the pretrained encoders to obtain the base latents for each stage. Rescaling increases the region's effective spatial resolution across all latent stages and the decoded voxel representation. Regeneration is conditioned on the crop of the input image within Ω, passed through the generator's standard preprocessing.

3.3 Boundary-distance release field

Our refinement requires at least one *attachment boundary*: a crop-box face crossed by the base mesh where the selected region connects to the surrounding asset. We anchor nearby latents to the base asset and allow stronger regeneration farther away. Each latent token receives a *release level* $\tau \in [0, \tau_{\max}]$. The same τ sets its initial noising time $t = \tau$ and determines when it stops following the noised base latent and follows the learned flow instead (Section 3.4). Larger τ means stronger initial noising and earlier release. Tokens with $\tau = 0$ remain anchored and are restored to their base values at the end; tokens with $\tau = 1$ start from pure noise. The per-stage maximum $\tau_{\max} \le 1$ controls refinement strength.

For a token at position u, we measure its distance to each attachment boundary as a fraction of the box extent in that direction, and take the smallest value $d(u)$. We set $\tau = 0$ within an *anchored band* of width g, then increase it smoothly over a *transition band* of width r:

$$\tau(u) = \tau_{\max}\,\rho^2(3 - 2\rho), \qquad \rho = \mathrm{clamp}\Big(\frac{d(u) - g}{r},\, 0,\, 1\Big). \tag{2}$$

Beyond the transition band ($d(u) \geq g + r$), τ reaches $\tau_{\max}$.

Each stage uses this distance-based rule with its own empirically chosen band widths (g, r). We choose the widths for sparse structure (SS), shape, and texture generation to satisfy

$$g_{\text{shape}} + r_{\text{shape}} \leq g_{\text{SS}}, \qquad g_{\text{tex}} + r_{\text{tex}} \leq g_{\text{shape}}, \tag{3}$$

so each group's anchored and transition bands lie within the preceding group's anchored band.

3.4 Spatially varying flow sampling

Let z be the base latent for a given stage and $\bar{z}$ be its per-channel standardized version (the SS latent is used as is), following the TRELLIS.2 implementation (Xiang et al., 2025a). The release level τ_i for token i is given by Eq. 2. We sample Gaussian noise ϵ once per stage and reuse it throughout that stage's sampling. Using Eq. 1, we obtain the noised base latent of token i at time t as

$$x_i^{\text{src}}(t) = (1 - t)\, \bar{z}_i + \big(\sigma_{\min} + (1 - \sigma_{\min})\, t\big)\, \epsilon_i.$$

Sampling runs over a descending time ladder $t_0 > t_1 > \cdots > t_K = 0$; its construction differs between SS and SLat stages. Each token is initialized at its own level, $x_i(t_0) = x_i^{\text{src}}(\tau_i)$, which need not match the first model time t_0. After each Euler step $\tilde{x} = x(t_k) - (t_k - t_{k+1})\, v_\theta(x(t_k), t_k)$, with the image condition and classifier-free guidance (Ho & Salimans, 2022) of the generator, unreleased tokens are reset to the noised base latent at the next sampling time. We call this operation *re-anchoring*:

$$x_i(t_{k+1}) = \begin{cases} x_i^{\text{src}}(t_{k+1}) & \text{if } \tau_i < t_{k+1}, \\ \tilde{x}_i & \text{otherwise,} \end{cases} \quad k < K - 1, \qquad x_i(t_K) = \begin{cases} \bar{z}_i & \text{if } \tau_i < t_{K-1}, \\ \tilde{x}_i & \text{otherwise.} \end{cases} \tag{4}$$

Through self-attention, the flow model can use unreleased tokens as base-asset context at the corresponding noise level when updating released tokens. At the end of sampling, tokens that were never released, including those in the anchored band, are restored exactly to their base latents. The construction is the structured-latent counterpart of per-pixel edit strength in 2D diffusion (Levin & Fried, 2023; Kim et al., 2024); it shares the known-region replacement of RePaint (Lugmayr et al., 2022) but uses a graded release instead of a binary mask and performs no resampling jumps.

The re-anchored sampling of Eq. 4 runs at every stage of the shape cascade with τ evaluated at that stage's coordinates, so coarse structure changes only where it is released. We retain the generator's guidance and $\sigma_{\min}$ for these stages. The shape SLats additionally blend their output toward the base latent by release level, and the SS stage copies the base occupancy wherever τ vanishes.

3.5 Texture refinement

Texture is refined after geometry, on the refined shape coordinates; the decoded refined mesh is never re-encoded. The base texture SLat is encoded in the same local frame from the crop seed and the base asset's PBR attributes, and the texture model uses the same boundary-distance field, evaluated with the texture-stage g and r, and the same re-anchored sampling of Eq. 4. By Eq. 3, the texture anchored and transition bands lie inside the shape SLat's anchored band, where base texture values are available at matching coordinates. The result decodes to PBR channels on the refined 1024^3 voxels.

3.6 Reintegration in mesh space

The refined region is decoded to a mesh with per-vertex PBR values and mapped back to the original coordinate frame using the inverse transforms. This restores the region's position, orientation and scale without an additional alignment fit. The cut is made in mesh space with the same crop box B used for selection: base faces whose centroid c_f lies inside B are removed, and the refined mesh takes their place,

$$\mathcal{M} = \{f \in \mathcal{M}_{\text{base}} : c_f \notin B\} \cup \mathcal{M}_{\text{ref}}. \tag{5}$$

The two parts are concatenated, with no mesh boolean, vertex welding or stitching. The cut is binary although τ is smooth: the transition band controls how strongly latents are released, while anchoring encourages compatible geometry and a smooth transition in color at attachment boundaries. The joined mesh is then re-voxelized at 1024^3 and remeshed for export.

Table 1: Comparison with refinement and editing methods. Whole-asset metrics against GT; CD and SNE $\times 100$; IoU at 128^3; FID is set-level. GPT-5.6: pairwise win share (1–10, chance 5.0) for overall quality and geometry detail.

	Geometry					Appearance					GPT-5.6	
Method	CD ↓	$F1_{0.25\%}$ ↑	$F1_{1\%}$ ↑	IoU ↑	SNE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DINO ↑	FID ↓	Ovr. ↑	Geo. ↑
PAniC-3D (N=329)												
GT					*reference*						8.52	6.83
TRELLIS.2	**2.938**	0.138	0.389	0.383	1.12	15.80	0.882	0.118	0.792	43.17	6.48	6.82
Easy3E	3.955	0.074	0.265	0.294	1.24	13.17	0.878	0.128	0.588	98.74	2.44	3.02
Elevate3D	3.059	0.115	0.372	0.377	1.26	16.59	0.876	0.122	0.671	68.25	3.81	0.71
DetailGen3D	2.939	0.127	**0.393**	0.386	**1.04**	–	–	–	–	–	–	4.55
NOFACE3D	2.939	**0.142**	0.389	**0.391**	1.12	**16.78**	**0.883**	**0.115**	**0.812**	**33.02**	**8.36**	**8.07**
TexVerse (N=151)												
GT					*reference*						7.30	6.33
TRELLIS.2	7.27	0.123	0.306	0.334	1.97	18.02	0.860	0.135	0.728	66.56	3.84	3.09
Easy3E	11.7	0.060	0.219	0.250	1.81	17.25	0.851	0.145	0.663	74.87	6.13	5.61
Elevate3D	13.7	0.058	0.194	0.198	2.16	16.97	0.836	0.158	0.620	73.94	3.89	3.14
DetailGen3D	35.4	0.009	0.032	0.055	**1.23**	–	–	–	–	–	–	5.32
NOFACE3D	**7.22**	**0.140**	**0.320**	**0.337**	1.61	**19.57**	**0.863**	**0.134**	**0.741**	**64.92**	**7.22**	**6.50**

4 EXPERIMENTS

We compare NOFACE3D with refinement and editing methods on PAniC-3D and TexVerse, and with TRELLIS.2 on BIGDETAIL, and in a human study (Section 4.2). We then ablate the noise schedule (Section 4.3) and show that the method also supports localized 3D editing (Section 4.4).

4.1 SETUP

Datasets and baselines. PAniC-3D (Chen et al., 2023b) pairs single images of full-body characters with artist meshes. Their head and upper body carry much of the perceived identity but occupy only a small fraction of the latent grid, so the marked region is a semantic crop of that part. We compare with baselines on 329 of the 4,892 assets, because Elevate3D takes about 25 minutes per asset on an NVIDIA RTX A6000 (Ryu et al., 2025). NOFACE3D runs the TRELLIS.2 cascade once per region, close to the cost of one TRELLIS.2 generation, which takes about 17 s at 1024^3 on an NVIDIA H100 (Xiang et al., 2025a). TexVerse (Zhang et al., 2025) adds 151 general objects with authored PBR materials, none from Objaverse. We also propose BIGDETAIL, a benchmark of objects whose detail sits in a small region that a uniform latent grid under-resolves. Each subject pairs a large, smooth body with a compact cluster of named detail parts (e.g. a steel tank with its valve and trigger assembly), spanning everyday and industrial objects. We generate one image per subject with Nano Banana 2, reconstruct it with the original TRELLIS.2 model, and annotators mark each detail-critical region as an image rectangle and an oriented 3D box. The resulting 339 assets have no artist mesh, so they are judged pairwise against the original TRELLIS.2 result. The baselines, DetailGen3D (Deng et al., 2024), Elevate3D (Ryu et al., 2025) and Easy3E (Hu et al., 2026), are each applied to the original TRELLIS.2 result.

Metrics. Against the registered GT mesh, we report Chamfer distance, F-score, surface IoU at 128^3, and Sharp Normal Error (SNE) (Chen et al., 2025), following prior refinement and editing work (Deng et al., 2024; Li et al., 2025a; Ye et al., 2025b). On matched RGB views we report PSNR, SSIM(Wang et al., 2004) and LPIPS (Zhang et al., 2018), as in 3D editing benchmarks (Li et al., 2025a; Ye et al., 2025b), plus DINOv2-large (Oquab et al., 2024) cosine similarity (*DINO*) and FID (Heusel et al., 2017), following Xiang et al. (2025b) and Li et al. (2025a). We also use GPTEval3D (Wu et al., 2024): a vision-language judge (GPT-4o (OpenAI, 2024) or GPT-5.6) compares pairs of unlabeled assets on six criteria, and we report its 1–10 win share, for which chance is 5.0. Data splits, registration, and the metric, judging and user-study protocols are in Appendix B; the construction of BIGDETAIL is in Appendix C.



Figure 3: **Qualitative comparison** on PAniC-3D (top) and TexVerse (bottom): texture and normal map per method (DetailGen3D is untextured).

Table 2: Human preference, TRELLIS.2 versus NOFACE3D on PAniC-3D (16 reviewers, 200 assets). *Neither* is a share of all responses; the other rows are over judged ratings.

	Joined	Genshin	VRoid
Responses	1,188	593	595
Neither	12.0% (143)	6.4% (38)	17.6% (105)
TRELLIS.2	12.2% (127)	9.0% (50)	15.7% (77)
Tie	4.5% (47)	2.9% (16)	6.3% (31)
NOFACE3D	**83.3**% (871)	**88.1**% (489)	**78.0**% (382)

Table 3: GPTEval3D (GPT-5.6) on BIGDETAIL (N=339), TRELLIS.2 versus NOFACE3D. Head-to-head win share (1–10); each pair sums to 10.

Criterion	TRELLIS.2	NOFACE3D
Fidelity	3.76	**6.24**
3D plausibility	4.60	**5.40**
Geo.–tex. alignment	3.89	**6.11**
Texture detail	4.61	**5.39**
Geometry detail	4.21	**5.79**
Overall	4.26	**5.74**

4.2 MAIN RESULTS

Refinement and editing methods. NOFACE3D is best on nearly every metric on both datasets (Table 1); TRELLIS.2 leads Chamfer on PAniC-3D by 0.001. DetailGen3D is best on SNE because it sharpens the whole surface, but this also warps the shape (TexVerse Chamfer 35.4 vs. 7.22). Easy3E and Elevate3D fall below TRELLIS.2 on most metrics.

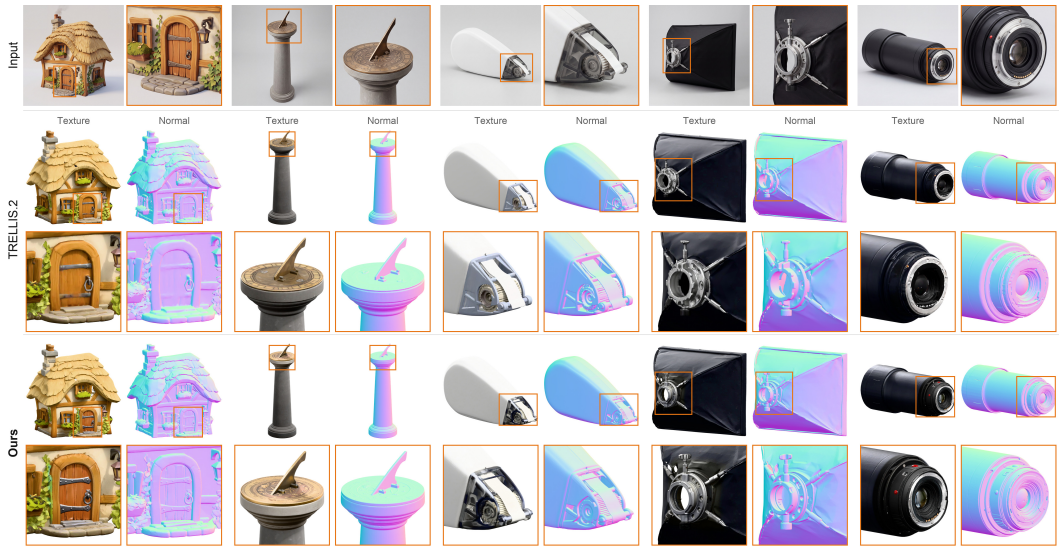

Figure 4: **Qualitative results on BIGDETAIL.** Top: input image with the labeled region and its enlargement. Below: TRELLIS.2 and NOFACE3D, texture and normal map.

Under GPT-5.6, NOFACE3D has the highest overall and geometry-detail win shares among generated methods on both datasets (Table 1). It even scores above GT on geometry detail (8.07 vs. 6.83 on PAniC-3D and 6.50 vs. 6.33 on TexVerse), since it produces denser geometric detail consistent with the image. The gain is not detail density alone: DINO similarity to GT also rises, from 0.792 to 0.812 on PAniC-3D and from 0.728 to 0.741 on TexVerse.

In Figure 3, NOFACE3D recovers the faces, hair and accessories that TRELLIS.2 smears, and object detail such as the throne's side panels, while leaving the rest of each asset unchanged. DetailGen3D warps the throne and the telephone, Elevate3D blotches textures and breaks the telephone cord, and Easy3E returns nearly blank character textures.

Human preference. Sixteen reviewers compared TRELLIS.2 and NOFACE3D blind on 200 PAniC-3D assets, giving 1,188 responses on a 1–9 scale. 83.3% of judged ratings prefer NOFACE3D and 12.2% TRELLIS.2, with a signed mean of $+1.83$ on $[-4, +4]$ (95% CI $[+1.52, +2.11]$, reviewer-cluster bootstrap), on both subsets (Table 2). 12.0% of responses were *neither*, both generations too poor to judge; counting them, 73.3% of all responses prefer NOFACE3D.

Results on BIGDETAIL. On the 339 refined BIGDETAIL assets, GPT-5.6 prefers NOFACE3D to TRELLIS.2 on all six criteria (Table 3; overall 5.74 vs. 4.26).Figure 4 shows that NOFACE3D recovers details TRELLIS.2 blurs, such as the door straps, the tape-dispenser gear and the lens-mount contacts.

4.3 NOISE SCHEDULING ABLATION

Figure 5 compares refinement with and without noise scheduling. In the ablated variant, we regenerate the selected region without boundary anchoring or the spatially varying noise schedule. Regeneration alone does not reliably produce a smooth transition to the retained asset, leaving visible discontinuities in geometry and appearance at attachment boundaries. With noise scheduling, the regenerated region connects more coherently to its surroundings.

4.4 OTHER APPLICATIONS: LOCALIZED 3D EDITING

Conditioning on an edited crop turns NOFACE3D into a localized 3D editor. We edit the input image with Gemini Flash Image and run the unchanged refinement pipeline. Regeneration away from attachment boundaries allows edits to change the region's shape, while the anchored band and the mesh-space rejoin keep the rest of the asset close to the original. Edits compose: in Figure 6 a

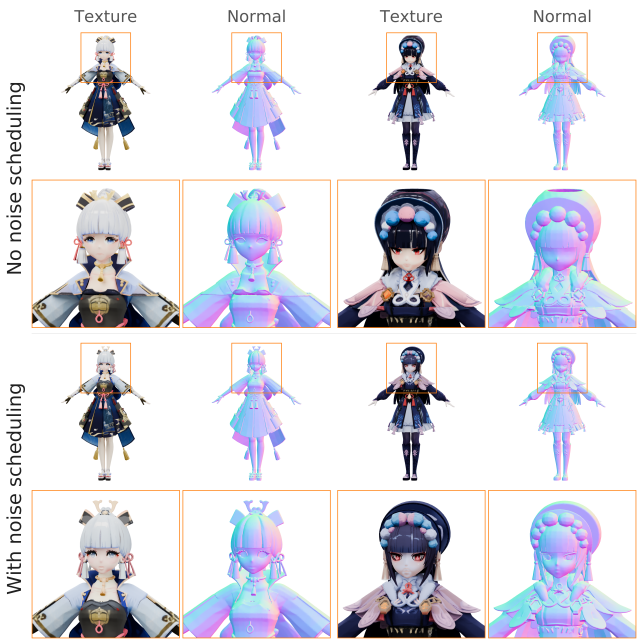

Figure 5: **Noise scheduling ablation.** Top: regeneration without anchoring or noise scheduling. Bottom: with noise scheduling. Texture and normal views use matched cameras and detail crops. Orange boxes mark the regions enlarged below each full-asset view.

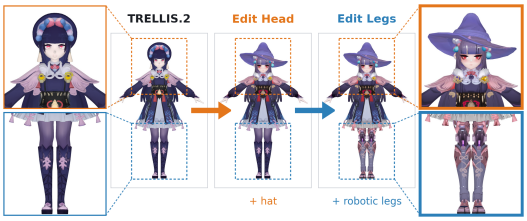

Figure 6: **Localized 3D editing.** A Genshin character gains a hat (head region), then robotic legs (leg region). Dashed boxes mark the regions.

character gains a hat, then robotic legs, and the second edit leaves the hat untouched. Please refer to our appendix for more examples.

5 Conclusion

We presented NOFACE3D, a training-free framework that refines the detail-critical regions of an asset produced by a pretrained image-to-3D generator. It reallocates the latent budget rather than enlarging it: the detailed region's 3D content is rescaled to fill the generator's latent grid, re-encoded there, and regenerated under a boundary-distance field that pins the interface latents and restores them exactly, after which the refined region is combined with the surrounding mesh. The method needs only the frozen generator, the input image and a box around the target region, so it is not tied to a category or body part, and conditioning the same process on an edited crop turns it into a localized 3D editor.

Limitations. Because NOFACE3D is training-free and runs on the frozen TRELLIS.2 generator, it inherits that generator's prior: re-encoding adds latent capacity but does not change what the generator can produce, so text such as printed labels and lettering, which TRELLIS.2 struggles to render legibly, often remains distorted after refinement. Post-training the generator on text-rich samples may alleviate this.

AI USE STATEMENT

In this work, we used generative AI tools for tasks with recommended disclosure: (i) editing the paper to improve readability, including polishing grammar and wording and revising paragraphs for clarity and concision; (ii) formatting references and LaTeX, for example, laying out tables and figures; and (iii) identifying relevant literature, searching for information, and analyzing existing work. Generative models are also part of our experimental setup, as described in the paper: Nano Banana 2 generates the BIGDETAIL input images (Section 4.1 and Appendix C), Gemini Flash Image produces the edited crops for localized editing (Section 4.4), and GPT-4o and GPT-5.6 serve as judges in the GPTEval3D evaluation, whose conclusions we check against a human preference study (Section 4.2). We have reviewed all AI-assisted work: the authors checked every AI-suggested edit to ensure that the technical content and claims were unchanged. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our human preference study asked sixteen reviewers to compare pairs of rendered 3D assets (Section 4.2 and Appendix B). Beyond this study, our evaluation uses the publicly available PAniC-3D and TexVerse datasets and the BIGDETAIL images, which we generate ourselves. PAniC-3D contains stylized characters designed by third parties, which we use only for research evaluation. As with other 3D generators and editors, NOFACE3D could be used to alter or reproduce assets without their creators' permission, or to create misleading content. Our method is training-free, operates on a frozen pretrained generator, and adds no generative capability beyond that generator: it only reallocates the generator's capacity to a user-marked region.

REPRODUCIBILITY STATEMENT

Our method is training-free and runs on the publicly released TRELLIS.2 model, described in Section 3.1 with its configuration in Appendix A. Section 3 specifies the local re-encoding frame, the boundary-distance field and its nesting constraint, the re-anchored sampling rule and the mesh-space reintegration, and Appendix A lists the settings that differ from the released generator, including the per-stage band widths. Section 4.1 and Appendix B give the data splits, metric settings, judge protocol and human-study protocol, and Appendix C describes how BIGDETAIL is built, including the subject specification, image generation and region labeling. Appendix A also gives the hardware requirements and the computational cost relative to one TRELLIS.2 generation. We will release our code and BIGDETAIL upon acceptance.

REFERENCES

Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3D: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023a.

Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. In *CVPR*, 2025.

Shuhong Chen, Kevin Zhang, Yichun Shi, Heng Wang, Yiheng Zhu, Guoxian Song, Sizhe An, Janus Kristjansson, Xiao Yang, and Matthias Zwicker. PAniC-3D: Stylized single-view 3D reconstruction from portraits of anime characters. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21068–21077, 2023b.

Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, Guosheng Lin, and Chi Zhang. MeshAnything: Artist-created mesh generation with autoregressive transformers. *arXiv preprint arXiv:2406.10163*, 2024.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Ken Deng, Yuan-Chen Guo, Jingxiang Sun, Zi-Xin Zou, Yangguang Li, Xin Cai, Yan-Pei Cao, Yebin Liu, and Ding Liang. DetailGen3D: Generative 3D geometry enhancement via data-dependent flow. *arXiv preprint arXiv:2411.16820*, 2024.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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*, 2024.

Shimin Hu, Yuanyi Wei, Fei Zha, Yudong Guo, and Juyong Zhang. Easy3E: Feed-forward 3D asset editing via rectified voxel flow. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12730–12740, 2026.

Sora Kim, Sungho Suh, and Minsik Lee. RAD: Region-aware diffusion models for image inpainting. *arXiv preprint arXiv:2412.09191*, 2024.

Klaus Krippendorff. *Content Analysis: An Introduction to Its Methodology*. SAGE Publications, 4th edition, 2018.

Hankyeol Lee, Wooyeol Baek, Seongdo Kim, and Jongyoo Kim. REVIVE 3D: Refinement via encoded voluminous inflated prior for volume enhancement. *arXiv preprint arXiv:2604.27504*, 2026.

Eran Levin and Ohad Fried. Differential diffusion: Giving each pixel its strength. *arXiv preprint arXiv:2306.00950*, 2023. Also in Computer Graphics Forum 44:e70040, 2025.

Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3D: Fast text-to-3d with sparse-view generation and large reconstruction model. In *International Conference on Learning Representations (ICLR)*, 2024.

Lin Li, Zehuan Huang, Haoran Feng, Gengxiong Zhuang, Rui Chen, Chunchao Guo, and Lu Sheng. VoxHammer: Training-free precise and coherent 3D editing in native 3D space. *arXiv preprint arXiv:2508.19247*, 2025a.

Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, and Yan-Pei Cao. TripoSG: High-fidelity 3D shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025b.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3D: High-resolution text-to-3d content creation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

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

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*, 2023.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023a.

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*, 2023b.

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. Wonder3D: Single image to 3d using cross-domain diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11461–11471, 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*, 2022.

OpenAI. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. 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 Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Geon Yeong Park, Roman Shapovalov, Rakesh Ranjan, Jong Chul Ye, Andrea Vedaldi, and Thu Nguyen-Phuoc. MeshReGen: A unified 3D geometry regeneration framework. *arXiv preprint arXiv:2604.28134*, 2026.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3d using 2d diffusion. In *International Conference on Learning Representations (ICLR)*, 2023.

Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. XCube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Nuri Ryu, Jiyun Won, Jooeun Son, Minsu Gong, Joo-Haeng Lee, and Sunghyun Cho. Elevating 3D models: High-quality texture and geometry refinement from a low-quality model. In *ACM SIGGRAPH Conference Papers*, 2025. doi: 10.1145/3721238.3730701.

Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. MVDream: Multi-view diffusion for 3d generation. In *International Conference on Learning Representations (ICLR)*, 2024.

Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. MeshGPT: Generating triangle meshes with decoder-only transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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

Wanhu Sun, Zhongjin Luo, Heliang Zheng, Jiahao Chang, Chongjie Ye, Huiang He, Shengchu Zhao, Rongfei Jia, and Xiaoguang Han. EI-Part: Explode for completion and implode for refinement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

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*, 2024.

Jiaxiang Tang, Zhaoshuo Li, Zekun Hao, Xian Liu, Gang Zeng, Ming-Yu Liu, and Qinsheng Zhang. EdgeRunner: Auto-regressive auto-encoder for artistic mesh generation. In *International Conference on Learning Representations (ICLR)*, 2025.

Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single RGB images. In *European Conference on Computer Vision (ECCV)*, volume 11215, pp. 55–71. Springer, 2018.

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. ProlificDreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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, Yikang Yang, Yajie Bao, Jiachen Qian, et al. Direct3D-S2: Gigascale 3D generation made easy with spatial sparse attention. In *Advances in Neural Information Processing Systems*, 2025.

Tong Wu, Guandao Yang, Zhibing Li, Kai Zhang, Ziwei Liu, Leonidas Guibas, Dahua Lin, and Gordon Wetzstein. GPT-4V(ision) is a human-aligned evaluator for text-to-3D generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3D generation. *arXiv preprint arXiv:2512.14692*, 2025a.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025b.

Bojun Xiong, Si-Tong Wei, Xin-Yang Zheng, Yan-Pei Cao, Zhouhui Lian, and Peng-Shuai Wang. OctFusion: Octree-based diffusion models for 3d shape generation. *Computer Graphics Forum*, 2025.

Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. InstantMesh: Efficient 3D mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024.

Yunhan Yang, Yuan-Chen Guo, Yukun Huang, Zi-Xin Zou, Zhipeng Yu, Yangguang Li, Yan-Pei Cao, and Xihui Liu. HoloPart: Generative 3d part amodal segmentation. *arXiv preprint arXiv:2504.07943*, 2025a.

Yunhan Yang, Yufan Zhou, Yuan-Chen Guo, Zi-Xin Zou, Yukun Huang, Ying-Tian Liu, Hao Xu, Ding Liang, Yan-Pei Cao, and Xihui Liu. OmniPart: Part-aware 3d generation with semantic decoupling and structural cohesion. *arXiv preprint arXiv:2507.06165*, 2025b.

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 *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025a.

Junliang Ye, Shenghao Xie, Ruowen Zhao, Zhengyi Wang, Hongyu Yan, Wenqiang Zu, Lei Ma, and Jun Zhu. NANO3D: A training-free approach for efficient 3d editing without masks. *arXiv preprint arXiv:2510.15019*, 2025b.

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*, 42(4), 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*, 43(4), 2024.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Yibo Zhang, Li Zhang, Rui Ma, and Nan Cao. Texverse: A universe of 3d objects with high-resolution textures. *arXiv preprint arXiv:2508.10868*, 2025.

Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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.

A IMPLEMENTATION DETAILS

Generator. We use the released 4B TRELLIS.2 model in its default `1024_cascade` configuration. DINOv3 features are extracted at 512 pixels for the first two stages and at 1024 pixels for the last two. The SS latent decodes to 64^3 occupancy, max-pooled to the sparse 32^3 support of the low-resolution shape SLat; the shape decoder upsamples that SLat, and its predicted occupancy is quantized to the sparse 64^3 grid of the high-resolution shape SLat. The texture SLat is sampled on the same 64^3 coordinates with the shape SLat concatenated to its input, and the texture decoder, guided by the shape decoder's intermediate subdivisions, lifts it to six PBR channels (base color, metallic, roughness, alpha). We use Euler sampling with time rescaling and classifier-free guidance; the time ladders are specified below.

Cost. With $\tau_{\max} = 1$, NOFACE3D samples each stage of the 1024^3 cascade once per region with the released 12-step ladder, so its shape stages cost the same as in one TRELLIS.2 generation. Xiang et al. (2025a) report about 17 s for a 1024^3 asset on an NVIDIA H100, 10 s for shape and 7 s for material. The only additional cost is voxelizing and encoding the crop seed, which uses the generator's own conversion routines and encoders. NOFACE3D loads the same models as TRELLIS.2 and runs on the same hardware, which Xiang et al. (2025a) specify as a GPU with at least 24 GB of memory.

Crop box and condition. For PAniC-3D and TexVerse, we use square image bounding boxes and align the base asset to the image by its frontal silhouette. Since a single image cannot constrain depth, the crop box B starts at the front-most surface under Ω and its depth equals the image box's side length, retaining the front-facing portion when the object extends farther back. This is an experimental setting, not a requirement on the shape of the refinement box. The condition crop passes through the generator's standard preprocessing (background removal and recentering). In our PAniC-3D and TexVerse experiments it is a crop of the input image and is not upscaled, so no detail enters through an image model rather than through NOFACE3D. The crop of a user image can be low-resolution; it can then be upsampled by image interpolation or by a vision-language image model such as Gemini Flash Image, which touches only the 2D condition. NOFACE3D also runs on generated inputs: the BIGDETAIL images are generated with Gemini, and the editing crops of Section 4.4 are Gemini edits.

Coordinate transforms. Let p denote parent-mesh coordinates and $q = \phi_{\mathrm{o}}(p)$ coordinates in the box-aligned frame, where ϕ_{o} is the recorded alignment transform. For labeled oriented boxes, it includes the box rotation and the inverse scale used to cache the parent mesh. Let $b_{\min}$ and $b_{\max}$ be the selected submesh's tight bounding-box corners in this frame. Its local center and uniform scale are $c_\Omega = \frac{1}{2}(b_{\min} + b_{\max})$ and $s_\Omega = 0.99999/\|b_{\max} - b_{\min}\|_\infty$, giving local coordinates $y = \phi_\Omega(q) = A\, s_\Omega(q - c_\Omega)$, where A converts Y-up coordinates to the generator's Z-up convention. The complete mapping is $\phi_\Omega \circ \phi_{\mathrm{o}}$. Local rescaling reduces the latent pitch in the box-aligned frame by a factor of s_Ω. Reintegration applies $q = \phi_\Omega^{-1}(y) = A^{-1}y/s_\Omega + c_\Omega$, then $p = \phi_{\mathrm{o}}^{-1}(q)$. Any initial parent-cache scaling is undone on final export.

Boundary-distance field. We write $\mathcal{G}$ for the nonempty set of attachment boundaries, identified once per selection as

$$\mathcal{G} = \big\{\, w : \text{a triangle of the base mesh straddles the plane of box face } w \text{ within its extent} \,\big\}. \quad (6)$$

This set can also be specified explicitly. A token at integer coordinate n on a stage's R^3 grid ($R =$ 16, 32 or 64) has normalized position $u = (n + \frac{1}{2})/R \in [0,1]^3$, corresponding to the local cube $\mathcal{C}$ shifted to the unit cube. In this frame, each face w of the crop box B lies in a plane $u_{a_w} = p_w$ perpendicular to axis a_w. Let ℓ_w denote the box extent along that axis. The normalized inward distance is

$$d(u) = \min_{w \in \mathcal{G}} d_w(u), \qquad d_w(u) = \pm\, \frac{u_{a_w} - p_w}{\ell_w}, \quad (7)$$

with $+$ for a lower face and $-$ for an upper face, so that d_w is positive inside B. Because the smoothstep is monotone, applying Eq. 2 to the smallest normalized distance is equivalent to taking the minimum release level over attachment boundaries. The band widths are $(g, r) = (0.16, 0.04)$ on the SS latent, $(0.10, 0.06)$ on the shape SLats and $(0.05, 0.05)$ on the texture SLat. They were chosen empirically and satisfy Eq. 3 with equality ($0.10 + 0.06 = 0.16$ and $0.05 + 0.05 = 0.10$); both shape SLat stages use the same band. The release maximum $\tau_{\max}$ is tunable independently at

each stage. Lowering it on SS retains more of the base structure at initialization, while the shape and texture maxima separately control regeneration of geometry and appearance.

Time ladders. Let N be the configured step count and $h_\alpha(t) = \alpha t/[1 + (\alpha - 1)t]$ the generator's time-rescaling function, with rescale factor α. For shape and texture SLats, the ladder starts at $\tau_{\max}$ and keeps entries of $\{h_\alpha(1-j/N)\}_{j=0}^{N}$ below $\tau_{\max}$, ending at zero. The SS implementation instead uses $K = \max(1, \mathrm{round}(N\tau_{\max}))$ steps with $t_k = h_\alpha(\tau_{\max}(1 - k/K))$. Thus its first model time need not equal $\tau_{\max}$. All stages initialize tokens at their individual τ_i and use Eq. 4 thereafter: release levels are not an upper bound on the noise encountered while anchored to the noised base latent.

Stage outputs. A shape token whose coordinate has no counterpart in the base latent receives a zero reference before standardization and $\tau_i = \tau_{\max}$. After de-standardizing $x(t_K)$ to $\hat{z}$, the low- and high-resolution shape SLat stages return

$$z_i^{\mathrm{out}} = z_i + \lambda_i\,(\hat{z}_i - z_i), \qquad \lambda_i = \tau_i/\tau_{\max}, \tag{8}$$

and on the final high-resolution shape SLat, tokens in the anchored band are then overwritten with their base rows, coordinates and features, so anchoring is enforced twice and independently, in the sampler and in feature space. The SS latent is not blended; it is decoded to 64^3 occupancy, every cell whose release level, upsampled from 16^3 by nearest neighbor, is at most 10^{-7} copies the base occupancy, and the result is max-pooled to the 32^3 support as in the base generator.

Texture. The texture reference is aligned to the refined shape coordinates. Within the *shape* SLat's anchored band, it copies base texture values at matching integer coordinates; elsewhere it is zero before standardization. This provides base texture values throughout both the texture anchored band and its transition band, since $g_{\mathrm{tex}} + r_{\mathrm{tex}} \le g_{\mathrm{shape}}$. The standardized shape SLat is concatenated to the texture model's input as in the base generator. The texture sampler uses the default settings of the released TRELLIS.2 model (Xiang et al., 2025a): 12 Euler steps with time-rescale factor 3.0, and guidance strength 1.0 with guidance rescale 0, which disables classifier-free guidance. The blend of Eq. 8 is omitted, so every released token keeps its sampled value, and tokens in the anchored band are overwritten with their base rows as for the shape SLat. The result decodes, guided by the refined shape decoder's subdivisions, to the six PBR channels.

Reintegration. Base vertices take PBR values queried from the base asset's PBR voxels, and refined vertices take those of the refined decode. The full refined submesh is concatenated with the retained parent faces; it is not clipped again to the crop box.

B Evaluation details

Data. We use the Genshin (PMX) and VRoid (VRM) subsets of PAniC-3D. On these characters the head and upper body carry much of the perceived identity while occupying little of the latent grid, which is why the marked region is a crop of that part. Of the 5,631 PAniC-3D assets with completed original TRELLIS.2 results and refinement (132 Genshin, 5,499 VRoid), the 4,892 that complete every evaluator form the large-scale set, and the 329 on which every baseline also completed (120 Genshin, 209 VRoid) form the comparison set. The original TRELLIS.2 results are generated once and cached, so they differ from ours only by the refinement. GT, original and refined meshes are registered and rendered with a shared camera and normalization.

Baselines. On PAniC-3D and TexVerse every method sees the same single image. DetailGen3D outputs untextured meshes, so its appearance metrics are not reported. Easy3E receives the image crop as its target view and the crop box as its 3D mask.

Metrics. Chamfer distance and F-score use 250,000 points per mesh. SNE uses 22 views on a Fibonacci sphere at 512^2, and the appearance metrics use four 1024^2 views at azimuths 30°, 120°, 210°, 300° and pitch 30°. DINO and FID measure semantic and distributional similarity rather than pixel-wise agreement. FID is computed over the set of rendered views and has no per-asset deviation.

Judged evaluation. Following GPTEval3D (Wu et al., 2024), the judge is GPT-4o or GPT-5.6, which we use as a scalable proxy for human preference; the user study of Section 4.2 agrees with it. The six criteria are fidelity and completeness, 3D plausibility, geometry–texture alignment, texture detail, geometry detail, and overall quality; the judge is not told which asset is refined. The

geometry–texture alignment criterion asks whether the geometric detail is consistent with the image and texture, that is, how realistic the geometry looks, not whether it matches GT. The judge sees four RGB views and four normal maps of each asset at azimuths $0°$, $90°$, $180°$, $270°$. In the *pairwise* protocol it picks the better of two assets or a tie; in the *monadic* protocol it rates each asset from 1 to 10 alone, and pairwise outcomes are derived by comparing the independent ratings. Both report the win share $10\,(\text{wins} + 0.5\,\text{ties})/\text{games}$, for which chance is 5.0. Side-by-side judging is position-biased: in a pilot, GPT-4o chose the left panel 68.5% of the time. The comparison with baselines therefore counts both orders of every pair, and the large-scale tournament ($1{,}000$ assets, $8{,}000$ judge calls) keeps only decisions on which both orders agree. The labeled trial shows the artist mesh as a named reference and asks which of the two assets matches it better; as in the tournament, only assets on which both orders agree are counted.

Human study. The study in Section 4.2 pairs TRELLIS.2 and NOFACE3D on 200 PAniC-3D assets from the Genshin and VRoid subsets. Each trial shows rotating textured and normal-map views of both assets and the conditioning image of the full object and of the marked region; the options are unlabeled, with left/right assignment and orientation randomized. Reviewers rate 1–9 (left to right preference) or answer *neither* when the base generation is too poor to judge. Ratings are mapped so that 9 favours NOFACE3D, and the signed score is the mapped rating minus 5. Sixteen reviewers gave $1{,}188$ responses (7–181 judged ratings each, median 35); no demographic information was collected.

Human study statistics. Signed mean $+1.83$ (s.d. 1.76) over judged ratings, with 95% CIs $[+1.52, +2.11]$ (reviewer-cluster bootstrap), $[+1.64, +2.00]$ (asset-cluster bootstrap) and $[+1.32, +1.87]$ around $+1.59$ (crossed reviewer–asset random-effects model; variance 48% asset, 4% reviewer, 48% residual). Krippendorff's α (Krippendorff, 2018) is 0.44 on signed scores and 0.42 on the four labels. *Neither* is 12.0% of responses (6.4% Genshin, 17.6% VRoid); counting it, shares are TRELLIS.2 10.7%, tie 4.0%, NOFACE3D 73.3%, and coding it as a tie gives a mean of $+1.61$.

C BIGDETAIL CONSTRUCTION

TRELLIS.2 already reconstructs Objaverse-style assets well, such assets may overlap its training data, and few of them concentrate their detail in a small volume that a uniform latent grid under-resolves. Every BIGDETAIL subject is designed around that case. The images are generated, so no artist mesh exists and none of the reference-based metrics can be computed; BIGDETAIL is evaluated with pairwise GPTEval3D on the whole asset, the original TRELLIS.2 result against the refined asset for the same input.

Subjects and images. Each subject is specified by an object, its *smooth mass* (e.g. "red cylindrical steel tank"), and a compact *detail cluster* with named parts (e.g. "valve and trigger assembly: squeeze lever, carry handle, pressure gauge, safety pin, tamper seal"). Subjects span everyday and industrial objects: stationery and office equipment, hand and power tools, kitchen and household appliances, furniture and lighting, musical instruments, sports gear, cameras and laboratory instruments, vehicles, boats and aircraft, railway, firefighting, utility and space hardware, and buildings. For each subject we generate one 4096×4096 image with Nano Banana 2 (`gemini-3.1-flash-image`) and reconstruct it with the original TRELLIS.2 model.

Region labels. Annotators mark each detail-critical region as a pair of boxes, a rectangle on the image and an oriented 3D box on the mesh, with no assumed camera; an asset may carry several regions, or be marked as needing no refinement and excluded. The evaluation set is the 339 refined assets.

Refinement. We use the annotated bounding boxes directly: the 3D box specifies the selection and its orientation, whole triangles whose centroids fall inside it are kept, and the attachment boundaries $\mathcal{G}$ are the box faces crossed by the base mesh, as in Eq. 6. The per-stage band widths are those of Appendix A, with independently configurable release maxima. The condition is the exact image crop, resized with Lanczos interpolation and padded to a white 1024^2 square before the generator's background removal, without generative image sharpening. Each labeled region is refined in its own run.

D LARGE-SCALE COMPARISON WITH TRELLIS.2

Table 4 compares NOFACE3D with the original TRELLIS.2 results on all 4,892 PAniC-3D assets. Every appearance metric improves, most clearly FID (13.97 → 7.43) and DINO (0.812 → 0.830), the fine F-score and surface IoU rise (F1 at 0.25% from 0.141 to 0.144, IoU from 0.373 to 0.381), and pointwise geometry otherwise changes little: Chamfer differs by 0.33%, F1 at 1% by less than 0.1% and SNE by 0.2%. This is consistent with the rest of the asset being preserved. The refinement shows in appearance against GT and in judged and human preference.

Under the monadic protocol, both judges prefer NOFACE3D to TRELLIS.2 on all six criteria, with overall quality 4.81 vs. 4.22 on GPT-4o and 5.02 vs. 3.53 on GPT-5.6 (Table 5); the margins are moderate because most of each asset is the preserved remainder. A pairwise tournament on a 1,000-asset subset, keeping only decisions on which both orders agree, gives the same ordering. Finally, a labeled trial shows the artist mesh as a named reference and asks which of the two matches it better: of the 871 assets on which both orders agree, the judge chose NOFACE3D on 681 and TRELLIS.2 on 188, 78.4% of the 869 decided assets, and judged the remaining 2 a tie.

Table 4: TRELLIS.2 versus NOFACE3D on 4,892 PAniC-3D assets against artist GT. Mean ± population s.d.; Chamfer and SNE ×100; FID is set-level.

Metric	TRELLIS.2	NOFACE3D
Chamfer ↓	$\mathbf{3.359 \pm 2.764}$	3.370 ± 2.773
F1 @ 0.25% ↑	0.141 ± 0.070	$\mathbf{0.144 \pm 0.070}$
F1 @ 1% ↑	0.37219 ± 0.15536	0.37222 ± 0.15548
Surface IoU 128^3 ↑	0.373 ± 0.140	$\mathbf{0.381 \pm 0.142}$
SNE ↓	$\mathbf{1.312 \pm 0.610}$	1.314 ± 0.620
PSNR ↑	15.70 ± 2.17	$\mathbf{16.53 \pm 2.76}$
SSIM ↑	0.8884 ± 0.0290	$\mathbf{0.8906 \pm 0.0299}$
LPIPS ↓	0.1163 ± 0.0265	$\mathbf{0.1123 \pm 0.0268}$
DINO ↑	0.812 ± 0.114	$\mathbf{0.830 \pm 0.119}$
FID ↓	13.97	**7.43**

Table 5: Monadic judged comparison on 4,892 PAniC-3D assets: each mesh is rated alone, and win shares (1–10) are derived from the ratings.

	GPT-4o			GPT-5.6		
Criterion	GT	TRELLIS.2	NOFACE3D	GT	TRELLIS.2	NOFACE3D
Fidelity / completeness	5.68	4.41	**4.91**	6.38	3.61	**5.01**
3D plausibility	5.63	4.64	**4.73**	5.87	4.33	**4.80**
Geo.–tex. alignment	5.74	4.44	**4.82**	4.94	4.67	**5.40**
Texture details	6.05	4.08	**4.86**	7.46	2.42	**5.12**
Geometry details	5.64	4.63	**4.73**	5.29	4.75	**4.96**
Overall quality	5.97	4.22	**4.81**	6.45	3.53	**5.02**

E ADDITIONAL RESULTS

Tables 6 and 7 repeat the comparison with refinement and editing methods with standard deviations across assets, and Tables 8 and 9 give its judged results on all six criteria. Table 10 gives both judges' scores on BIGDETAIL. Figure 7 shows two more editing sequences.

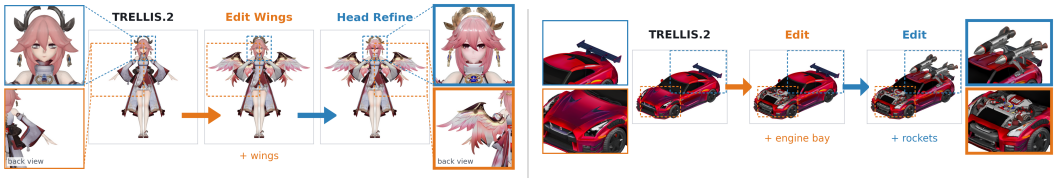

Figure 7: **More editing sequences.** Left: a Genshin character gains wings, then its head is refined. Right: a car gains an engine bay (front), then rockets (rear).

Table 6: Objective comparison on PAniC-3D (N=329). Mean ± s.d.; Chamfer and SNE ×100; FID is set-level.

Metric	TRELLIS.2	Easy3E	Elevate3D	DetailGen3D	NOFACE3D
Chamfer ↓	**2.938 ± 1.884**	3.955 ± 2.071	3.059 ± 1.896	2.939 ± 1.900	2.939 ± 1.878
F1 @ 0.25% ↑	0.138 ± 0.073	0.074 ± 0.034	0.115 ± 0.051	0.127 ± 0.056	**0.142 ± 0.074**
F1 @ 1% ↑	0.389 ± 0.166	0.265 ± 0.111	0.372 ± 0.146	**0.393 ± 0.153**	0.389 ± 0.165
Surface IoU 128^3 ↑	0.383 ± 0.144	0.294 ± 0.113	0.377 ± 0.135	0.386 ± 0.141	**0.391 ± 0.147**
SNE ↓	1.12 ± 0.40	1.24 ± 0.37	1.26 ± 0.38	**1.04 ± 0.40**	1.12 ± 0.40
PSNR ↑	15.80 ± 1.90	13.17 ± 1.08	16.59 ± 1.76	–	**16.78 ± 2.44**
SSIM ↑	0.882 ± 0.023	0.878 ± 0.023	0.876 ± 0.021	–	**0.883 ± 0.025**
DINO ↑	0.792 ± 0.097	0.588 ± 0.094	0.671 ± 0.083	–	**0.812 ± 0.103**
LPIPS ↓	0.118 ± 0.021	0.128 ± 0.019	0.122 ± 0.018	–	**0.115 ± 0.021**
FID ↓	43.17	98.74	68.25	–	**33.02**

Table 7: Objective comparison on TexVerse (N=151). Mean ± s.d.; Chamfer and SNE ×100.

Metric	TRELLIS.2	Easy3E	Elevate3D	DetailGen3D	NOFACE3D
Chamfer ↓	7.27 ± 14.9	11.7 ± 19.3	13.7 ± 18.7	35.4 ± 31.0	**7.22 ± 14.8**
F1 @ 0.25% ↑	0.123 ± 0.147	0.060 ± 0.059	0.058 ± 0.064	0.009 ± 0.010	**0.140 ± 0.150**
F1 @ 1% ↑	0.306 ± 0.230	0.219 ± 0.188	0.194 ± 0.178	0.032 ± 0.031	**0.320 ± 0.234**
Surface IoU 128^3 ↑	0.334 ± 0.247	0.250 ± 0.193	0.198 ± 0.169	0.055 ± 0.044	**0.337 ± 0.244**
SNE ↓	1.97 ± 1.33	1.81 ± 1.37	2.16 ± 1.40	**1.23 ± 0.84**	1.61 ± 1.18
PSNR ↑	18.02 ± 9.24	17.25 ± 9.10	16.97 ± 9.15	–	**19.57 ± 9.29**
SSIM ↑	0.860 ± 0.066	0.851 ± 0.068	0.836 ± 0.071	–	**0.863 ± 0.065**
DINO ↑	0.728 ± 0.134	0.663 ± 0.139	0.620 ± 0.133	–	**0.741 ± 0.130**
LPIPS ↓	0.135 ± 0.061	0.145 ± 0.063	0.158 ± 0.066	–	**0.134 ± 0.061**
FID ↓	66.56	74.87	73.94	–	**64.92**

Table 8: Pairwise judged comparison on PAniC-3D (N=329). Win share (1–10, chance 5.0).

Criterion	GT	TRELLIS.2	Easy3E	Elevate3D	DetailGen3D	NOFACE3D
GPT-4o						
Fidelity / completeness	7.84	6.41	2.17	4.73	–	**7.57**
3D plausibility	7.17	5.77	2.76	2.96	4.54	**6.81**
Geo.–tex. alignment	7.90	6.45	2.41	4.39	–	**7.60**
Texture details	7.87	6.46	1.95	4.93	–	**7.61**
Geometry details	7.36	5.88	2.23	3.04	4.50	**6.98**
Overall quality	7.87	6.43	2.32	4.49	–	**7.59**
GPT-5.6						
Fidelity / completeness	8.52	5.97	1.80	5.14	–	**8.20**
3D plausibility	7.74	6.56	3.38	0.69	4.27	**7.36**
Geo.–tex. alignment	7.49	7.00	3.37	3.33	–	**8.53**
Texture details	9.05	5.91	2.03	4.83	–	**8.11**
Geometry details	6.83	6.82	3.02	0.71	4.55	**8.07**
Overall quality	8.52	6.48	2.44	3.81	–	**8.36**

Table 9: Pairwise judged comparison on TexVerse (N=151). Win share (1–10, chance 5.0).

Criterion	GT	TRELLIS.2	Easy3E	Elevate3D	DetailGen3D	NOFACE3D
GPT-4o						
Fidelity / completeness	5.58	5.24	5.34	5.07	–	**5.72**
3D plausibility	5.61	5.21	5.41	5.01	3.12	**5.64**
Geo.–tex. alignment	5.59	5.23	5.38	5.05	–	**5.66**
Texture details	5.62	5.25	5.34	5.03	–	**5.71**
Geometry details	5.58	5.22	5.39	5.04	3.12	**5.65**
Overall quality	5.59	5.22	5.37	5.04	–	**5.66**
GPT-5.6						
Fidelity / completeness	7.28	4.05	5.75	3.82	–	**7.08**
3D plausibility	7.01	3.10	**6.26**	2.89	4.53	6.22
Geo.–tex. alignment	7.05	4.14	6.25	4.45	–	**7.31**
Texture details	7.92	4.37	5.67	4.73	–	**7.14**
Geometry details	6.33	3.09	5.61	3.14	5.32	**6.50**
Overall quality	7.30	3.84	6.13	3.89	–	**7.22**

Table 10: Pairwise judged comparison on BIGDETAIL (N=339), TRELLIS.2 versus NOFACE3D. Head-to-head win share (1–10); each pair sums to 10.

	GPT-4o		GPT-5.6	
Criterion	TRELLIS.2	NOFACE3D	TRELLIS.2	NOFACE3D
Fidelity / completeness	4.40	**5.60**	3.76	**6.24**
3D plausibility	4.42	**5.58**	4.60	**5.40**
Geo.–tex. alignment	4.40	**5.60**	3.89	**6.11**
Texture details	4.57	**5.43**	4.61	**5.39**
Geometry details	4.60	**5.40**	4.21	**5.79**
Overall quality	4.58	**5.42**	4.26	**5.74**