CVE3D: CONSISTENT MULTI-VIEW EXPANSION AND ADAPTIVE FUSION FOR COHERENT AND RELIABLE 3D GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Insufficient observational constraints remain a major challenge for image-conditioned 3D generation, particularly in the single-image setting. A high-quality 3D generator can preserve visible evidence but must hallucinate the occluded side of an object. Novel-view diffusion provides complementary evidence, but independently generated views often drift in appearance, geometry, or pose. These cross-view inconsistencies impede the integration of view generation and multi-view-conditioned 3D generation into a unified pipeline. We present CVE3D, a consistent view-expansion framework that bridges this gap. A provisional 3D representation generated from the source image is rendered at each target camera pose to provide view-aligned RGB, depth, and surface-normal cues. Conditioned on these cues and the detached correction from the preceding view, our View-by-View Consistency Guidance Adapter corrects a selectively fine-tuned Zero-1-to-3 backbone to sequentially synthesize consistent target views, thereby reducing cross-view inconsistencies before 3D generation. We generate and rank complete trajectories based on source-referenced semantic consistency and adjacent-view continuity, rather than selecting each view independently. The selected views are subsequently integrated into TRELLIS.2 using our Consistency-Guided Adaptive Flow-Prediction Fusion, which guides cross-attention entropy fusion with view-consistency scores to limit the influence of residual unreliable observations during 3D generation. Experiments demonstrate that CVE3D achieves excellent overall generation quality, with consistently strong performance in semantic alignment and rendered-view fidelity. Qualitative comparisons further show detailed geometry and coherent appearance across viewpoints, confirming the effectiveness of our consistency-guided view expansion and fusion design. These results establish CVE3D as a practical and effective framework for integrating novel-view synthesis with multi-view-conditioned 3D generation. Code is available at `https://anonymous.4open.science/r/CVE3D-AD3B`.

1 INTRODUCTION

Recent image-to-3D systems, including Hunyuan3D 2.1 (Hunyuan3D et al., 2025), Pixal3D (Li et al., 2026b), Unique3D (Wu et al., 2024a), and Fancy123 (Yu et al., 2025), have improved the fidelity and efficiency of generating textured assets from a single image. Despite these advances, image-conditioned generation remains limited by insufficient observational constraints. A single image constrains only visible surfaces, leaving occluded geometry and appearance dependent on learned priors. Generation is consequently sensitive to the input viewpoint, particularly for non-canonical views. Increasing model capacity cannot replace the missing observations, and plausible completions need not match the underlying object.

Novel-view generation offers complementary conditioning views (Liu et al., 2023; Long et al., 2024; Liu et al., 2024; Huang et al., 2025). However, high-quality generated views may still exhibit inconsistencies in geometry, appearance, or camera pose. Treating these views as compatible observations can propagate such discrepancies into the generated 3D asset. Integrating the two stages therefore requires agreement with the source image and continuity between generated views, beyond image quality at individual viewpoints.

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

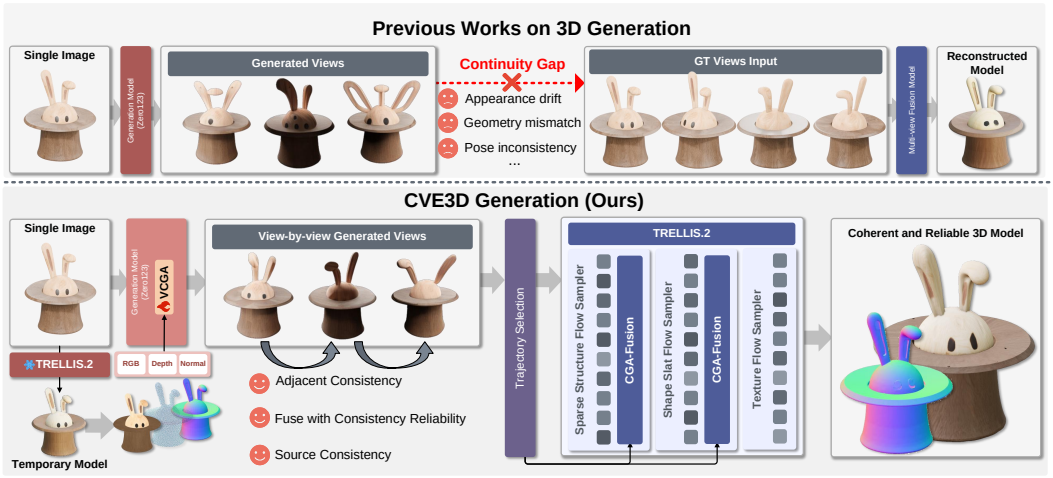


Figure 1: Overview of CVE3D. Existing two-stage pipelines face a continuity gap between generated views, which prevents seamless integration of novel-view generation and multi-view 3D generation. CVE3D instead constrains view expansion along an ordered camera trajectory, evaluates trajectory consistency, and uses the resulting scores to guide adaptive fusion during 3D generation.

This requirement differs from reconstruction with captured multi-view observations. ReconViaGen (Chang et al., 2026) uses generative priors to complete regions unobserved by multiple inputs, while MV-SAM3D (Li et al., 2026a) adapts fusion through attention entropy and optional visibility cues. Both begin with multiple observations. Our setting instead starts from one image and must establish which generated observations should condition a pretrained 3D model, and how strongly.

We present CVE3D, a consistent view-expansion framework illustrated in Figures 1 and 2. A provisional source-view generation supplies RGB, depth, and normal cues at target cameras. Conditioned on these cues and the detached preceding-view correction, the trainable View-by-View Consistency Guidance Adapter (VCGA) corrects the selectively fine-tuned Zero-1-to-3 backbone view by view. Complete trajectories are ranked by source-referenced semantic consistency and adjacent-view continuity. The selected views enter TRELLIS.2 through Consistency-Guided Adaptive Flow-Prediction Fusion (CGA-Fusion), where their consistency scores guide cross-attention entropy fusion of per-view flow predictions during 3D generation.

Our contributions are summarized below:

- We formulate *consistent view expansion* as an interface between novel-view generation from one image and multi-view-conditioned 3D generation, with source-referenced consistency and adjacent-view continuity as explicit requirements.
- We introduce a trainable VCGA that uses rendered appearance and geometric cues together with the detached correction from the preceding view to guide generation view by view with selective fine-tuning of the view-generation backbone.
- We combine trajectory-level selection with CGA-Fusion, using view-consistency scores to guide adaptive cross-attention entropy fusion during TRELLIS.2 generation.
- We conduct extensive experiments on Toys4K and GSO, demonstrating the leading performance of CVE3D in semantic alignment and rendered-view fidelity. Ablation studies further confirm the effectiveness of both proposed consistency mechanisms.

2 Related Work

Image-Conditioned 3D Generation. Image-conditioned 3D modeling includes direct prediction of meshes and Gaussians (Pan et al., 2019; Szymanowicz et al., 2024) and generative modeling of geometry and appearance with learned representations (Hunyuan3D et al., 2025; Xiang et al., 2026; Yan et al., 2026; Shim et al., 2026). Complementary approaches improve image alignment and

geometric detail through pixel-aligned conditioning or deformation-based refinement (Li et al., 2026b; Yu et al., 2025), while multi-view RGB and normal generation provides intermediate cues for mesh reconstruction (Wu et al., 2024a). These directions advance representation quality and reconstruction fidelity. We focus on the consistency of generated observations supplied to a 3D generator.

Generative Novel-View Synthesis. Camera-conditioned diffusion extends a source image to novel viewpoints (Liu et al., 2023). Joint multi-view and cross-domain modeling coordinates the generation of images and geometric cues (Liu et al., 2024; Long et al., 2024), while parameter-efficient adaptation and flexible view conditioning extend pretrained diffusion models to multi-view generation (Huang et al., 2025; Zhou et al., 2025). Geometry-aware approaches incorporate rendering or depth-based warping to strengthen cross-view correspondence (Höllein et al., 2024; Cao et al., 2025). Related evaluation studies explicitly assess consistency between generated views, complementing conventional image-quality metrics (Asim et al., 2025; Xie et al., 2025). Building on these directions, we use renderings of a fixed source-derived model together with preceding corrective context to guide a selectively fine-tuned view generator sequentially.

Coupled View Generation and Adaptive Multi-View 3D Generation. Prior studies have explored coupling view generation with reconstruction for objects, scenes, and individual parts (Wang et al., 2024; Gao et al., 2024; Chen et al., 2025). These efforts investigate embedding reconstruction into multi-view denoising or using diffusion priors to constrain unobserved regions (Xu et al., 2024; Wu et al., 2024b). Other approaches explore reconstruction feedback by returning rendered 3D cues to the diffusion process (Wen et al., 2025; Chen et al., 2024), while scene-level methods combine generated observations with iterative reconstruction or consistency-constrained sampling (Wu et al., 2025; Wang et al., 2025). For multi-view object generation, reconstruction priors and adaptive fusion have been explored to improve agreement with input observations (Chang et al., 2026; Li et al., 2026a). In particular, MV-SAM3D uses cross-attention entropy for adaptive fusion, inspiring our local aggregation mechanism. Our focus is on controlling the consistency of generated observations and their contributions to subsequent 3D generation. Our framework jointly addresses source consistency and adjacent-view continuity through sequential corrections, trajectory selection, and consistency-guided flow-prediction fusion.

3 METHOD

3.1 OVERVIEW

CVE3D couples view generation and multi-view-conditioned 3D generation through two complementary modules, illustrated in Figure 2. Geometry-Conditioned Sequential View Expansion uses the trainable VCGA to correct a selectively fine-tuned view generator with rendered guidance and preceding corrective context. CGA-Fusion then combines trajectory selection with adaptive fusion inside TRELLIS.2, using source-referenced consistency and adjacent-view continuity to guide the aggregation of per-view flow predictions.

3.2 GEOMETRY-CONDITIONED SEQUENTIAL VIEW EXPANSION

Provisional Source-View Generation. The frozen TRELLIS.2 generator $G_{3\mathrm{D}}$ first constructs a provisional model $M_0 = G_{3\mathrm{D}}(I_0)$ from the source image. At target camera C_i, the rasterization-based renderer R produces:

$$F_i = R(M_0, C_i) = [R_i^{\mathrm{rgb}}, R_i^{\mathrm{depth}}, R_i^{\mathrm{normal}}] \in \mathbb{R}^{7\times H\times W}, \tag{1}$$

where H and W denote the rendering height and width. The seven channels comprise three RGB, one depth, and three normal components. Holding M_0 fixed provides a shared source-anchored reference for the generated trajectory. These features serve as conditioning signals rather than definitive view constraints, promoting agreement with the source observation while the diffusion prior completes unobserved content.

View-by-View Consistency Guidance Adapter. Let $x_{i,t}$ denote the noisy diffusion latent of target view i at timestep t, and let C_i be its camera-to-world transformation. We denote source-to-target

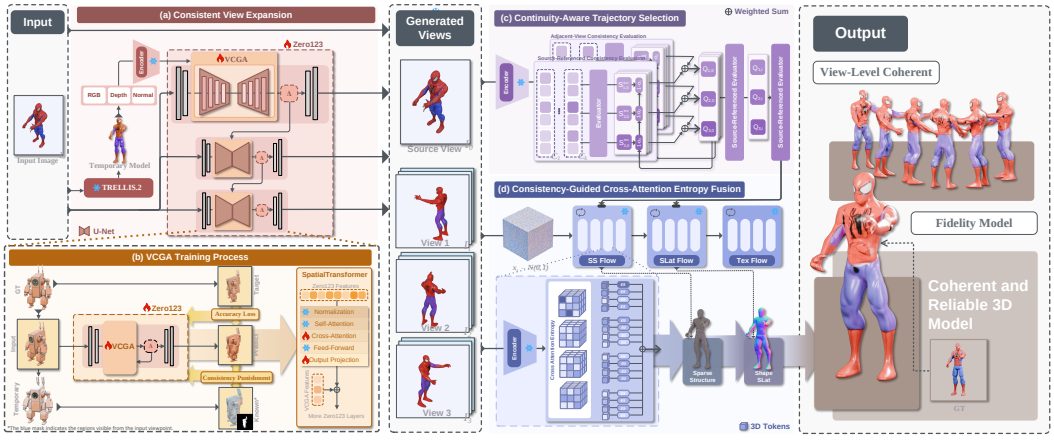


Figure 2: Architecture of CVE3D. (a) A provisional TRELLIS.2 model generated from the source view is rendered at the target cameras. VCGA uses the resulting RGB, depth, and normal features to correct Zero-1-to-3 sequentially along each candidate trajectory. (b) VCGA and selected Zero-1-to-3 layers are jointly optimized with visibility-weighted diffusion supervision. (c) Candidate trajectories are evaluated using source-referenced consistency and adjacent-view continuity. (d) The selected trajectory is processed through CGA-Fusion, where consistency scores guide adaptive weighting and cross-attention entropy implements token-wise aggregation of per-view flow predictions in the sparse-structure and shape stages.

camera motion by $\Delta C_{0\to i}$ and adjacent-camera motion by $\Delta C_{i-1\to i} = C_i^{-1} C_{i-1}$. Before adapter correction, the Zero-1-to-3 denoiser predicts:

$$\epsilon_{i,t}^{\text{base}} = \epsilon_\theta(x_{i,t}, I_0, \Delta C_{0\to i}, t). \tag{2}$$

The adapter learns structured residual corrections to this backbone prediction. Conditioned on rendered features and trajectory state H_i, the adapter A_ϕ prepares multi-scale feature conditions $\{g_{i,t}^s\}_s$ and a noise-level condition $g_{i,t}^\epsilon$. At each matching-resolution SpatialTransformer block, a feedback module B_ϕ^s combines the current output $h_{i,t}^s$ with the corresponding adapter-generated condition to produce an additive correction. A noise-level feedback module B_ϕ^ϵ similarly corrects the feature-conditioned noise prediction:

$$(g_{i,t}^\epsilon, \{g_{i,t}^s\}_s) = A_\phi(x_{i,t}, t, F_i, C_i, \Delta C_{i-1\to i}, H_i), \tag{3}$$

$$\Delta h_{i,t}^s = B_\phi^s(h_{i,t}^s, g_{i,t}^s), \tag{4}$$

$$\epsilon_{i,t}^{\text{feat}} = \epsilon_\theta\big(x_{i,t}, I_0, \Delta C_{0\to i}, t; \{\Delta h_{i,t}^s\}_s\big), \tag{5}$$

$$\hat{\epsilon}_{i,t} = \epsilon_{i,t}^{\text{feat}} + B_\phi^\epsilon(\epsilon_{i,t}^{\text{feat}}, g_{i,t}^\epsilon). \tag{6}$$

The projection heads for both pathways are zero-initialized to preserve the backbone's initial predictions, with s indexing the spatial scales at which feature corrections are applied. The corresponding resolutions and projection dimensions are detailed in Appendix A.1.

The preceding view's total noise-prediction correction provides context for the next view at the same diffusion timestep. We define:

$$H_i = \text{sg}(\delta_{i-1,t}), \qquad \delta_{i,t} = \hat{\epsilon}_{i,t} - \epsilon_{i,t}^{\text{base}}, \tag{7}$$

where $\delta_{i,t}$ includes the effects of both feature and output corrections. The stop-gradient operator sg transfers corrective context without backpropagating through earlier views.

Adapter Training. We jointly optimize the adapter parameters ϕ and the cross-attention and SpatialTransformer output projections of Zero-1-to-3, keeping its remaining parameters frozen. TRELLIS.2 remains frozen, with no gradients through M_0, rendered guidance, or H_i. A visibility-weighted diffusion objective places greater emphasis on regions visible in the source image, encouraging source-view consistency for subsequent fusion while retaining standard supervision elsewhere to

learn completions of unseen content. Let V_i be the geometric source-visibility map, computed in image space and resized to the noise-prediction resolution. We minimize:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{i,t,\epsilon}\left[\frac{\sum_p (1+\lambda_v V_{i,p})\|\epsilon_p - \hat{\epsilon}_{i,t,p}\|_2^2}{C_\epsilon H_\epsilon W_\epsilon}\right], \tag{8}$$

where $\lambda_v > 0$ controls visibility weighting; C_ϵ, H_ϵ, and W_ϵ are the noise tensor's channel count, height, and width; and p indexes spatial locations. The expectation $\mathbb{E}_{i,t,\epsilon}$ covers target views, the prescribed feedback timesteps, and Gaussian noise. Training tuples, timestep selection, and visibility alignment are detailed in Appendix A.2.

3.3 Consistency-Guided Adaptive Flow-Prediction Fusion

Continuity-Aware Trajectory Selection. Independent best-of-K view selection can combine incompatible generation hypotheses. We instead sample K trajectories $\mathcal{T}_k = \{\hat{I}_i^{(k)}\}_{i=1}^N$, preserving camera order and sequential corrective context. A frozen image encoder measures cosine similarity to the source image ($s_{i,k}^{\text{src}}$) and the preceding generated view ($s_{i,k}^{\text{adj}}$). Their consistency score is

$$q_{i,k} = (1-\alpha_{i,k}) s_{i,k}^{\text{src}} + \alpha_{i,k} s_{i,k}^{\text{adj}}. \tag{9}$$

The first target uses source consistency alone, with $\alpha_{1,k} = 0$. For $i > 1$, the preceding view's consistency score $q_{i-1,k}$ within trajectory k determines the adjacent-view weight:

$$\alpha_{i,k} = \text{clip}\left(\frac{\exp(q_{i-1,k}/\tau_a)}{\exp(1/\tau_a) + \exp(q_{i-1,k}/\tau_a)}, \alpha_{\min}, \alpha_{\max}\right). \tag{10}$$

Here, τ_a controls the sensitivity to the preceding view's score, and $[\alpha_{\min}, \alpha_{\max}]$ bounds the adjacent-view contribution. This weighting limits the influence of a low-consistency predecessor on subsequent assessment, reducing the risk of trajectory-wide deviations. We select the trajectory maximizing $Q_k = N^{-1}\sum_{i=1}^N q_{i,k}$. Scores below a consistency threshold trigger additional sampling within the budget specified in Appendix A.3.

Consistency-Guided Cross-Attention Entropy. Inspired by MV-SAM3D (Li et al., 2026a), we use cross-attention entropy for multi-view fusion in TRELLIS.2 and introduce consistency multipliers tailored to generated views. Let q_i denote the score of selected view i. A fixed-anchor exponential mapping expands closely spaced scores, assigning larger multiplier changes to improvements at higher consistency levels before clipping:

$$m_i = \text{clip}\left(m_{\text{ref}} \exp\frac{q_i - q_{\text{ref}}}{\tau_q}, m_{\min}, m_{\max}\right). \tag{11}$$

The anchor pair $(q_{\text{ref}}, m_{\text{ref}})$ fixes the reference scale, τ_q controls expansion, and the bounds limit excessive amplification.

During sparse structure and geometry generation, we collect each view's cross-attention at the initial timestep. For 3D token p, let $H_{i,p}$ be the normalized entropy over image patches and $P_{i,p}$ the fraction of attention assigned to those patches. With $u_{i,p} = H_{i,p}P_{i,p}$, fusion weights are:

$$w_{i,p}^{\text{ent}} = \text{Norm}_i\left[m_i \exp(-\gamma_e u_{i,p})\right], \tag{12}$$

where Norm_i normalizes across views and $\gamma_e > 0$ scales the entropy response. The entropy term provides an implicit token-level reliability signal, while m_i adjusts a view's contribution according to source consistency and adjacent-view continuity. Weights are computed once per stage and reused, requiring no additional trainable parameters.

Stage-Wise Flow-Prediction Fusion. For the sparse-structure and shape stages $r \in \{\text{ss}, \text{shape}\}$, let $z_{p,t}^r$ be the shared state and c_i the image condition of view i, with $i = 0$ denoting the source. At every Euler step, the stage-specific weights $w_{i,p}^r$ from Equation 12 aggregate the frozen model's per-view flow predictions:

$$v_{p,t}^r = \sum_{i=0}^N w_{i,p}^r f_\theta^r(z_{p,t}^r, t, c_i). \tag{13}$$

The fused flow advances the shared state without directly averaging latents. TRELLIS.2 then generates texture from the resulting shape latent and source image and decodes the asset. Algorithm 1 in Appendix A.3 summarizes weighting and fusion.

4 EXPERIMENTS

4.1 IMPLEMENTATION DETAILS

We train VCGA and fine-tune selected Zero-1-to-3 layers on four NVIDIA L20 GPUs, sampling 1,200 objects per epoch from Objaverse-XL (Deitke et al., 2023). For evaluation, we generate assets at a resolution of 512^3 on Toys4K (Stojanov et al., 2021) and Google Scanned Objects (GSO) (Downs et al., 2022) using a single NVIDIA L20 GPU. Additional training, data preparation, and inference details are provided in Appendices A and B.

4.2 IMAGE-TO-3D GENERATION

We evaluate the ability of CVE3D to generate 3D assets from a single input image. In Figure 3, we present representative results across diverse objects, showcasing detailed geometry and faithful appearance. The generated assets preserve fine structural details and characteristic surface patterns, while their material representations support rendering under different lighting conditions. These results illustrate the ability of consistent view expansion and adaptive fusion to support high-fidelity 3D generation.

Qualitative Comparison. We compare our approach with TRELLIS.2, Hunyuan3D-2.1, Fancy123, and Pixal3D. In Figure 4, we present results from both front and rear viewpoints, highlighting the ability of CVE3D to generate detailed geometry and faithful appearance, including regions not visible in the input image. The rear-view comparisons further demonstrate strong generation quality in these unobserved regions, supporting the effectiveness of consistency-guided view expansion and adaptive fusion for 3D generation.

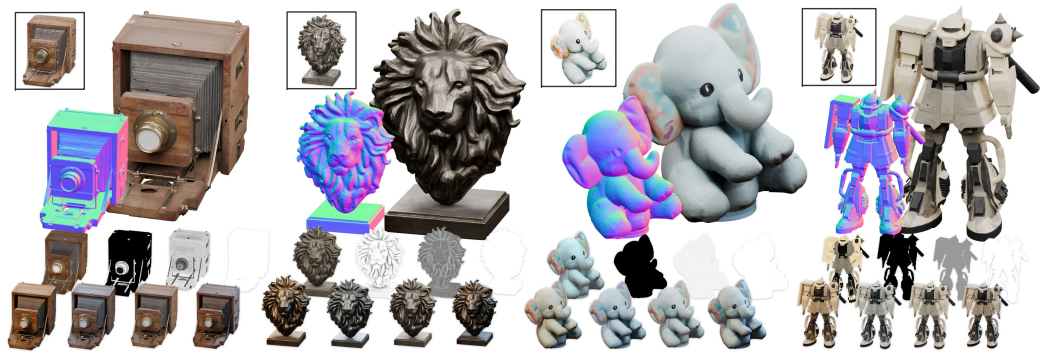


Figure 3: Assets generated by CVE3D from single images, showing geometry, material components, and renderings under different lighting conditions. Insets show the inputs.

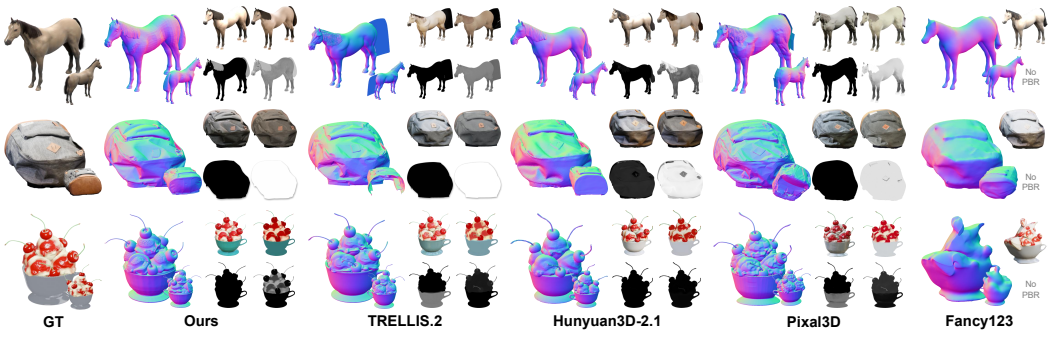


Figure 4: Qualitative comparison with baseline methods, showing front- and rear-view normals, rendered appearance, and available PBR components. GT denotes ground truth.

Quantitative Comparison. We conduct quantitative evaluation on Toys4K and GSO. Semantic alignment is assessed using CLIP and CLIP-N (Radford et al., 2021), and ULIP-2 (Xue et al., 2024), while RGB and normal rendering fidelity is measured with PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018). Table 1 shows that CVE3D leads across all reported metrics on GSO and achieves the best performance on most metrics on Toys4K. These results demonstrate the effectiveness of our approach in improving semantic alignment and rendered-image fidelity. To further validate the reliability of these findings, we conduct additional experiments, with the corresponding results and analyses presented in Appendix D.2.

Results reported in the original papers are provided solely for reference and do not support direct comparisons, owing to differences in datasets, evaluation metrics, and protocols, as well as the unavailability of some evaluation code. TRELLIS.2 results are taken from its Table 2. Results for Pixal3D, Fancy123, and Ouroboros3D are taken from Table 1 of each paper. For Ouroboros3D, we report the 3D reconstruction metrics.

Table 1: Quantitative image-to-3D evaluation on Toys4K and GSO, with original-paper results provided separately for reference. Values in the Toys4K and GSO blocks are reported as mean ± standard deviation. Cell colors indicate per-column relative performance from green (best) to red (worst); best values are bold and second-best values are underlined.

Method	Semantic			RGB			Normal		
	CLIP↑	CLIP-N↑	ULIP-2↑	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Toys4K									
TRELLIS.2 (Xiang et al., 2026)	0.816 ± 0.076	**0.722 ± 0.062**	0.394 ± 0.077	22.207 ± 4.033	0.902 ± 0.053	0.091 ± 0.050	25.990 ± 4.077	**0.936 ± 0.035**	0.077 ± 0.050
Hunyuan3D-2.1 (Hunyuan3D et al., 2025)	0.801 ± 0.075	0.705 ± 0.066	**0.407 ± 0.076**	19.758 ± 4.103	0.889 ± 0.059	0.108 ± 0.058	24.341 ± 4.762	0.927 ± 0.039	0.104 ± 0.064
Pixal3D (Li et al., 2026b)	0.857 ± 0.069	0.709 ± 0.064	0.354 ± 0.064	20.231 ± 3.833	0.895 ± 0.048	0.101 ± 0.047	23.228 ± 2.915	0.923 ± 0.031	0.110 ± 0.050
Fancy123 (Yu et al., 2025)	0.782 ± 0.075	0.652 ± 0.068	0.345 ± 0.075	19.490 ± 3.409	0.885 ± 0.052	0.117 ± 0.051	22.997 ± 3.389	0.922 ± 0.034	0.126 ± 0.053
Ouroboros3D (Wen et al., 2025)	0.770 ± 0.078	0.639 ± 0.069	0.297 ± 0.062	19.629 ± 3.493	0.881 ± 0.047	0.115 ± 0.044	21.274 ± 2.714	0.886 ± 0.048	0.140 ± 0.052
CVE3D (Ours)	**0.909 ± 0.048**	0.720 ± 0.047	0.364 ± 0.067	**23.226 ± 4.019**	**0.927 ± 0.023**	**0.070 ± 0.024**	**26.363 ± 2.397**	0.935 ± 0.020	**0.064 ± 0.023**
GSO									
TRELLIS.2 (Xiang et al., 2026)	0.793 ± 0.087	0.547 ± 0.092	0.276 ± 0.068	18.267 ± 3.400	0.867 ± 0.049	0.141 ± 0.051	21.373 ± 4.393	0.856 ± 0.065	0.181 ± 0.078
Hunyuan3D-2.1 (Hunyuan3D et al., 2025)	0.778 ± 0.095	0.535 ± 0.086	0.300 ± 0.063	16.867 ± 3.608	0.856 ± 0.061	0.154 ± 0.065	19.344 ± 3.978	0.842 ± 0.061	0.219 ± 0.086
Pixal3D (Li et al., 2026b)	0.821 ± 0.074	0.558 ± 0.086	0.277 ± 0.068	18.146 ± 2.089	0.868 ± 0.047	0.131 ± 0.044	16.321 ± 2.417	0.823 ± 0.054	0.174 ± 0.051
Fancy123 (Yu et al., 2025)	0.789 ± 0.067	0.488 ± 0.070	0.268 ± 0.068	18.812 ± 2.461	**0.877 ± 0.045**	0.123 ± 0.043	16.736 ± 2.522	0.847 ± 0.060	0.180 ± 0.056
Ouroboros3D (Wen et al., 2025)	0.689 ± 0.068	0.428 ± 0.073	0.202 ± 0.056	15.654 ± 2.568	0.858 ± 0.042	0.166 ± 0.049	17.779 ± 1.702	0.823 ± 0.040	0.224 ± 0.062
CVE3D (Ours)	**0.905 ± 0.044**	**0.643 ± 0.047**	**0.324 ± 0.059**	**21.307 ± 3.302**	**0.877 ± 0.028**	**0.107 ± 0.023**	**25.448 ± 2.746**	**0.908 ± 0.028**	**0.090 ± 0.025**
Paper Results (reference only)									
TRELLIS.2 (Xiang et al., 2026)	0.894	0.758	0.477	—	—	—	—	—	—
Hunyuan3D-2.1 (Hunyuan3D et al., 2025)	—	—	—	—	—	—	—	—	—
Pixal3D (Li et al., 2026b)	—	—	—	—	—	—	24.21	0.897	0.108
Fancy123 (Yu et al., 2025)	0.835	—	—	14.37	0.651	0.330	—	—	—
Ouroboros3D (Wen et al., 2025)	—	—	—	21.761	0.8894	0.1091	—	—	—

4.3 Ablation and Design Analysis

We conduct ablation studies to examine the contributions of VCGA and consistency-guided adaptive weighting. The four view-expansion configurations in Table 2 are evaluated on 100 randomly selected objects from GSO at a resolution of 512^3, using the same source images and evaluation protocol. The qualitative comparisons in Figure 5 complement the quantitative results, showing that the full model better preserves object structure and appearance in both examples, whereas the ablated configurations exhibit fragmented geometry or distorted shapes.

Table 2: Ablation study on GSO. Results are reported as mean ± standard deviation. Cell colors indicate relative performance within each column; the best and second-best results are marked in bold and with underlining, respectively.

Configuration	Semantic			RGB			Normal		
	CLIP↑	CLIP-N↑	ULIP-2↑	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
CVE3D (full)	**0.874 ± 0.055**	**0.613 ± 0.059**	**0.307 ± 0.070**	**19.614 ± 4.010**	**0.847 ± 0.050**	**0.131 ± 0.036**	23.989 ± 2.829	0.879 ± 0.045	0.113 ± 0.034
w/o Adapter	0.802 ± 0.112	0.547 ± 0.093	0.258 ± 0.071	17.639 ± 3.777	0.809 ± 0.059	0.178 ± 0.069	23.217 ± 3.369	0.873 ± 0.051	0.139 ± 0.064
w/o adaptive weighting	0.746 ± 0.105	0.511 ± 0.084	0.241 ± 0.069	16.650 ± 4.381	0.796 ± 0.065	0.202 ± 0.070	21.876 ± 3.500	0.860 ± 0.053	0.169 ± 0.067
w/o both	0.751 ± 0.112	0.512 ± 0.083	0.240 ± 0.067	16.743 ± 4.318	0.795 ± 0.066	0.203 ± 0.069	21.856 ± 3.459	0.859 ± 0.053	0.169 ± 0.067
Native TRELLIS.2	0.852 ± 0.084	0.584 ± 0.082	0.271 ± 0.072	18.908 ± 4.208	0.815 ± 0.059	0.159 ± 0.052	**24.559 ± 3.176**	**0.884 ± 0.048**	**0.110 ± 0.035**

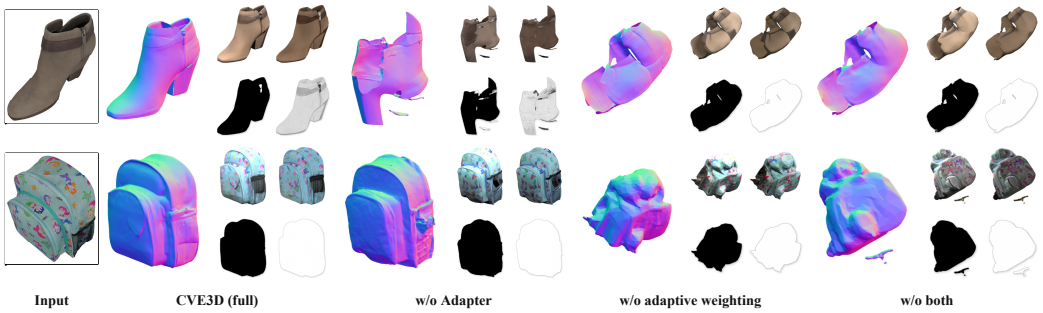


Figure 5: Qualitative ablation of VCGA and adaptive weighting. For each input, we compare normal renderings, appearance renderings, and material maps from the full model and three ablated configurations. The full model better preserves object silhouettes and structures, while removing components introduces distortions or missing parts.

View-by-View Consistency Adapter. We remove adapter corrections from the Zero-1-to-3 backbone while retaining trajectory selection and the 3D generation procedure. As shown in Table 2, removing VCGA consistently degrades semantic alignment as well as RGB and normal fidelity. This result confirms that adapter-guided view generation provides more reliable observations for subsequent 3D generation.

Consistency-Guided Adaptive Weighting. We replace both consistency multipliers and entropy-based weighting with uniform weights across views at every 3D token. The adapter and trajectory selection remain unchanged, isolating adaptive fusion from view generation. Uniform weighting consistently degrades semantic, RGB, and normal metrics, showing that generated views should not contribute equally during 3D generation. These results demonstrate that consistency-guided adaptive fusion improves the effective integration of generated views, leading to better semantic alignment and geometric and appearance fidelity.

Joint Contribution and View Expansion. Removing both components yields a direct view-expansion baseline in which generated observations are supplied to the 3D generator without consistency-aware refinement or adaptive fusion. Its consistently inferior performance relative to native TRELLIS.2 indicates that increasing the number of conditioning views is not inherently beneficial, as cross-view discrepancies can introduce conflicting evidence and degrade 3D generation. The recovery achieved by the complete model demonstrates that consistency must be regulated during both view generation and multi-view fusion, supporting the complementary roles of VCGA and consistency-guided adaptive weighting.

4.4 Hyperparameter Sensitivity Analysis

We analyze feedback timing, consistency-mapping temperature, and the maximum view multiplier on 100 randomly selected GSO objects (GSO100). Each sweep varies one hyperparameter while keeping all other settings fixed. Tables report object-level mean evaluation metrics.

Feedback Timing. The results in Table 3 show that feedback at iterations $(24, 36)$ achieves the highest CLIP, CLIP-N, and RGB PSNR. Feedback at $(32, 44)$ yields the best ULIP-2, RGB SSIM and LPIPS, and all normal-render metrics. These results reveal a trade-off between CLIP-based alignment and rendered-view fidelity across feedback schedules. These results suggest that, when prioritizing semantic alignment, earlier intervention may be preferable, while later intervention may better support appearance and surface-normal fidelity.

Consistency-mapping temperature. As shown in Table 4, $\tau_q = 0.05/\ln 2$ achieves the best semantic and normal-render metrics, as well as RGB LPIPS, whereas $\tau_q = 0.1/\ln 2$ yields the highest RGB PSNR and SSIM. Increasing τ_q to $0.2/\ln 2$ lowers performance across all reported metrics. The results favor smaller temperatures for semantic alignment and normal fidelity, with an intermediate value better preserving RGB PSNR and SSIM. These observations suggest choosing a

Table 3: Sensitivity to feedback iterations on GSO100 evaluation set. Entries are object-level means rounded to three decimal places; best values are bold and second-best values are underlined based on unrounded means.

Iterations	Semantic			RGB			Normal		
	CLIP↑	CLIP-N↑	ULIP-2↑	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
$(24, 36)$	**0.842**	**0.579**	0.261	**17.970**	<u>0.810</u>	<u>0.165</u>	<u>24.266</u>	<u>0.880</u>	<u>0.117</u>
$(28, 40)$	<u>0.841</u>	<u>0.578</u>	<u>0.261</u>	17.880	0.809	0.167	24.243	0.879	0.117
$(32, 44)$	0.841	0.576	**0.262**	<u>17.907</u>	**0.811**	**0.165**	**24.380**	**0.880**	**0.117**

lower temperature when prioritizing semantic alignment and surface-normal fidelity, while a moderate value may be more suitable when appearance accuracy is the primary objective.

Table 4: Sensitivity to consistency-mapping temperature on GSO100 with $m_{\max} = 20$. Entries are object-level means rounded to three decimal places; best values are bold and second-best values are underlined based on unrounded means.

τ_q	Semantic			RGB			Normal		
	CLIP↑	CLIP-N↑	ULIP-2↑	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
0.07213	**0.840**	**0.579**	**0.264**	<u>17.906</u>	<u>0.810</u>	**0.165**	**24.498**	**0.881**	**0.115**
0.14427	<u>0.837</u>	<u>0.578</u>	<u>0.261</u>	**17.972**	**0.812**	<u>0.165</u>	<u>24.319</u>	<u>0.880</u>	<u>0.116</u>
0.28854	0.835	0.576	0.259	17.796	0.807	0.171	23.969	0.877	0.122

Maximum View Multiplier. The results in Table 5 show that $m_{\max} = 20$ achieves the best CLIP-N, RGB metrics, normal PSNR, and normal LPIPS. A bound of 10 yields the highest CLIP and ULIP-2, while 40 gives the highest normal SSIM. The evaluated settings indicate that a tighter upper bound may be worth considering when semantic alignment is the main objective, whereas a moderate bound may offer a better balance for appearance and geometric fidelity.

Table 5: Sensitivity to the maximum view multiplier on GSO100 with $\tau_q = 0.1/\ln 2$. Entries are object-level means rounded to three decimal places; best values are bold and second-best values are underlined based on unrounded means. For $m_{\max} = 40$, semantic and RGB metrics use 99 valid results; all other entries use 100.

$m_{\max}$	Semantic			RGB			Normal		
	CLIP↑	CLIP-N↑	ULIP-2↑	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
5	<u>0.840</u>	0.574	0.258	17.830	0.808	0.169	24.105	0.878	0.121
10	**0.841**	0.577	**0.262**	17.811	<u>0.811</u>	<u>0.167</u>	24.274	<u>0.880</u>	<u>0.117</u>
20	0.837	**0.578**	0.261	**17.972**	**0.812**	**0.165**	**24.319**	0.880	**0.116**
40	0.838	<u>0.578</u>	<u>0.261</u>	<u>17.835</u>	0.810	0.167	<u>24.309</u>	**0.880**	0.117

5 Conclusion

We presented CVE3D, a framework that connects novel-view generation from a single image with multi-view-conditioned 3D generation through explicit consistency control. VCGA and trajectory selection produce coherent observations, while CGA-Fusion incorporates their consistency into adaptive aggregation of per-view flow predictions in TRELLIS.2. Experiments on Toys4K and GSO demonstrate strong semantic alignment and rendered-view fidelity, with qualitative and ablation results supporting both components. The qualitative comparisons further highlight its ability to preserve object structure and appearance in regions unobserved from the input view. CVE3D offers an effective framework for high-fidelity 3D generation from a single image, unifying consistent view expansion and adaptive fusion to produce coherent and reliable 3D assets.

AI USE STATEMENT

We did not use generative AI tools at any stage of this research or in the preparation of this manuscript. All authors have reviewed the manuscript and take responsibility for its final content, including the text, claims, results, and accompanying materials.

REFERENCES

Asim M, Wewer C, Wimmer T, et al. Met3r: Measuring multi-view consistency in generated images[C]//2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2025: 6034–6044.

Cao C, Yu C, Liu S, et al. Mvgenmaster: Scaling multi-view generation from any image via 3d priors enhanced diffusion model[C]//2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2025: 6045–6056.

Chang J, Ye C, Wu Y, et al. Reconviagen: Towards accurate multi-view 3d object reconstruction via generation[C]//International Conference on Learning Representations. 2026: 90387–90401.

Chen H, Shen B, Liu Y, et al. 3d-adapter: Geometry-consistent multi-view diffusion for high-quality 3d generation[J]. arXiv preprint arXiv:2410.18974, 2024.

Chen M, Shapovalov R, Laina I, et al. Partgen: Part-level 3d generation and reconstruction with multi-view diffusion models[C]//2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2025: 5881–5892.

Deitke M, Liu R, Wallingford M, et al. Objaverse-xl: A universe of 10M+ 3d objects[C]//Advances in Neural Information Processing Systems. 2023: 35799–35813.

Downs L, Francis A, Koenig N, et al. Google scanned objects: A high-quality dataset of 3d scanned household items[C]//2022 international conference on robotics and automation (ICRA). IEEE, 2022: 2553–2560.

Gao R, Hoł yński A, Henzler P, et al. Cat3d: Create anything in 3d with multi-view diffusion models[C]//Advances in Neural Information Processing Systems. Curran Associates, Inc., 2024: 75468–75494.

Höllein L, Božič A, Müller N, et al. Viewdiff: 3d-consistent image generation with text-to-image models[C]//2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2024: 5043–5052.

Huang Z, Guo Y C, Wang H, et al. Mv-adapter: Multi-view consistent image generation made easy[C]//2025 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2025: 16377–16387.

Hunyuan3D T, Yang S, Yang M, et al. Hunyuan3D 2.1: From images to high-fidelity 3D assets with production-ready PBR material[J]. arXiv preprint arXiv:2506.15442, 2025.

Li B, Wu D, Li J, et al. Mv-sam3d: Adaptive multi-view fusion for layout-aware 3d generation[J]. arXiv preprint arXiv:2603.11633, 2026a.

Li D Y, Zhao W, Chen Y, et al. Pixal3D: Pixel-aligned 3D generation from images[C]//Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers. 2026b: 1–12.

Liu R, Wu R, Van Hoorick B, et al. Zero-1-to-3: Zero-shot one image to 3d object[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2023: 9298–9309.

Liu Y, Lin C, Zeng Z, et al. Syncdreamer: Generating multiview-consistent images from a single-view image[C]//International Conference on Learning Representations. 2024: 27676–27697.

Long X, Guo Y C, Lin C, et al. Wonder3D: Single image to 3d using cross-domain diffusion[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2024: 9970–9980.

Pan J, Han X, Chen W, et al. Deep mesh reconstruction from single RGB images via topology modification networks[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2019: 9963–9972.

Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision[C]//Proceedings of the 38th International Conference on Machine Learning. 2021: 8748–8763.

Shim E, Lee H, Ju E. DiffGI: Differentiable geometry images for high-fidelity thin-shell 3d generation[C]//Computer Vision – ECCV 2026. Springer Nature Switzerland, 2026: 298–315.

Stojanov S, Thai A, Rehg J M. Using shape to categorize: low-shot learning with an explicit shape bias[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2021: 1798–1808.

Szymanowicz S, Rupprecht C, Vedaldi A. Splatter image: Ultra-fast single-View 3d reconstruction[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2024: 10208–10217.

Wang H, Liu Y, Liu Z, et al. VistaDream: Sampling multiview consistent images for single-view scene reconstruction[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2025: 26772–26782.

Wang X, Xie L, Dong C, et al. Real-esrgan: Training real-world blind super-resolution with pure synthetic data[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops. IEEE, 2021: 1905–1914.

Wang Z, Wang Y, Chen Y, et al. Crm: Single image to 3d textured mesh with convolutional reconstruction model[C]//European conference on computer vision. Springer, 2024: 57–74.

Wang Z, Bovik A, Sheikh H, et al. Image quality assessment: From error Visibility to structural similarity[J]. IEEE Transactions on Image Processing, 2004, 13(4): 600–612.

Wen H, Huang Z, Wang Y, et al. Ouroboros3d: Image-to-3d generation via 3d-aware recursive diffusion[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2025: 21631–21641.

Wu K, Liu F, Cai Z, et al. Unique3D: High-quality and efficient 3d mesh generation from a single image[C]//Advances in Neural Information Processing Systems. Curran Associates, Inc., 2024a: 125116–125141.

Wu R, Mildenhall B, Henzler P, et al. Reconfusion: 3d reconstruction with diffusion priors[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2024b: 21551–21561.

Wu S, Xu C, Huang B, et al. Genfusion: Closing the loop between reconstruction and generation via videos[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2025: 6078–6088.

Xiang J, Chen X, Xu S, et al. Native and compact structured latents for 3d generation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2026: 14419–14429.

Xie X, Lessen J E, Pons-Moll G. Mvgbench: A comprehensive benchmark for multi-view generation models[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2025: 8207–8218.

Xu Y, Tan H, Luan F, et al. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model[C]//International Conference on Learning Representations. 2024: 33168–33184.

Xue L, Yu N, Zhang S, et al. Ulip-2: Towards scalable multimodal pre-training for 3d understanding[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2024: 27091–27101.

Yan R, Cao Y P, Wang P, et al. Generative 3d gaussians with learned density control[C]//In Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26). Association for Computing Machinery, 2026.

Yu Q, Li X, Tang Y, et al. Fancy123: One image to high-quality 3d mesh generation via plug-and-play deformation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2025: 595–604.

Zhang R, Isola P, Efros A A, et al. The unreasonable effectiveness of deep features as a perceptual metric[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2018: 586–595.

Zhou J, Gao H, Voleti V, et al. Stable virtual camera: Generative view synthesis with diffusion models[C]//2025 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2025: 12405–12414.

A Additional Implementation Details

A.1 VCGA Architecture

The rendered guidance and the backbone prediction are defined in Equations 1 and 2, respectively. VCGA operates at 256^2 and has a base width of 64. Its feature residuals are projected to 320, 640, 1280, and 1280 channels at resolutions of 32^2, 16^2, 8^2, and 4^2, respectively, to match the corresponding backbone features. The index $s \in \{32, 16, 8, 4\}$ denotes these spatial scales.

A.2 Sequential Conditioning and Training Details

Each training tuple uses the four ground-truth renderings described in Appendix B.2. One provides the source observation, while the others supply ordered target supervision. Rendering the provisional model at the corresponding target cameras provides the guidance in Equation 1. The noisy input $x_{i,t}$ is obtained through the standard forward diffusion process at the prescribed feedback timestep. The preceding correction is initialized to zero for the first target. Each subsequent view receives only the detached total correction from the immediately preceding view at the same timestep through Equation 7.

Source visibility is computed geometrically in image space and resized to the spatial dimensions of $\hat{\epsilon}_{i,t}$ before loss evaluation, yielding $V_i \in [0, 1]^{1\times H_\epsilon \times W_\epsilon}$. In Equation 8, the squared norm sums errors over the C_ϵ channels at location p, while the outer sum covers all $H_\epsilon W_\epsilon$ spatial locations, producing a mean visibility-weighted noise-prediction error.

Optimization and Sampling. The fine-tuned Zero-1-to-3 layers comprise the cross-attention modules (`attn2`) and SpatialTransformer output projections (`proj_out`). All other backbone parameters and TRELLIS.2-4B remain frozen. We use AdamW with learning rates of 10^{-4} and 10^{-6} for the adapter and backbone layers, respectively, and weight decay 0.01. Training uses BF16 precision and gradient-norm clipping at 1.0. Each sample uses two diffusion timesteps corresponding to the configured inference feedback iterations, shared across its target views. Losses are averaged over target views and timesteps, and preceding-view corrections are detached.

Each device processes one object per micro-batch and accumulates gradients for 300 steps, yielding an effective global batch size of $1{,}200$ objects. Validation is performed every 10 optimizer updates on 20 objects from an object-disjoint validation partition. Early stopping uses a training patience of 20 steps, a validation patience of 5 checks, and a minimum improvement of 10^{-5}.

Residual Supervision. The training entry point explicitly disables recursive consistency supervision. An uncorrected classifier-free-guided noise prediction is computed without gradients. The corrected prediction is evaluated with gradients enabled, and its difference from the detached baseline is matched to the difference between the sampled noise and that same baseline. Subtracting the common baseline leaves the noise-prediction error in Equation 8 unchanged. The preceding-view state stores

this total predicted residual, rather than only the output correction head's contribution. Source visibility uses voxel-DDA with a 64^3 grid and area resizing to 32×32 for diffusion-loss weighting, with $\lambda_v = 2$.

A.3 Trajectory Selection and Fusion Implementation

With the frozen CLIP image encoder E_{sem} of Zero-1-to-3, we compute source-referenced and adjacent-view similarities as

$$s_{i,k}^{\text{src}} = \text{clip}_{[0,1]}\left[\cos\left(E_{\text{sem}}(\hat{I}_i^{(k)}), E_{\text{sem}}(I_0)\right)\right], \tag{14}$$

$$s_{i,k}^{\text{adj}} = \text{clip}_{[0,1]}\left[\cos\left(E_{\text{sem}}(\hat{I}_i^{(k)}), E_{\text{sem}}(\hat{I}_{i-1}^{(k)})\right)\right]. \tag{15}$$

The first target is evaluated using source-referenced consistency alone, with $\alpha_{1,k} = 0$. For subsequent targets, $\alpha_{i,k}$ is determined by the preceding view's consistency score $q_{i-1,k}$ within the same candidate trajectory. Each sampling round generates four complete trajectories. If the highest mean consistency score remains below 0.42, up to two additional rounds are performed.

For view i and 3D token p, let $a_{i,p,j}$ denote cross-attention renormalized over the J image patches. Its normalized entropy is

$$H_{i,p} = -\frac{1}{\log J}\sum_{j=1}^{J} a_{i,p,j}\log a_{i,p,j}. \tag{16}$$

The patch-attention mass $P_{i,p}$ measures the fraction of attention assigned to image patches within the conditioning token set, yielding $u_{i,p} = H_{i,p}P_{i,p}$.

To compute $w_{i,p}^{\text{ent}}$, we normalize the exponentiated entropy responses across views, multiply them by m_i, apply a positive floor η, and renormalize across views. The source view is assigned $q_0 = 1$, and all view multipliers follow the same consistency mapping. We use $\gamma_e = 30$ and $\eta = 0.01$. The floor retains a nonzero contribution from every selected view. Small numerical offsets ensure numerical stability. Algorithm 1 summarizes the weighting and flow-prediction fusion procedure. Weights are computed independently at the initial timestep of sparse-structure and shape generation, then reused throughout the corresponding stage.

Algorithm 1 Consistency-guided cross-attention entropy weighting and flow-prediction fusion.

Require: Stage r, image conditions $\{c_i\}_{i=0}^{N}$, scores $\{q_i\}_{i=1}^{N}$, source multiplier m_0
Require: Initial state $z_{t_0}^r$, schedule $\{t_k\}_{k=0}^{T}$, entropy scale γ_e, floor $\eta > 0$
Ensure: Fused stage output $z_{t_T}^r$

for $i = 1, \ldots, N$ **do**
  $m_i \leftarrow \text{clip}(m_{\text{ref}} \exp((q_i - q_{\text{ref}})/\tau_q), m_{\min}, m_{\max})$
end for
for $k = 0, \ldots, T-1$ **do**
  for $i = 0, \ldots, N$ **do**
    $v_i^r \leftarrow f_\theta^r(z_{t_k}^r, t_k, c_i)$
    if $k = 0$ **then**
      Collect patch-normalized attention $a_{i,p,j}^r$ and patch mass $P_{i,p}^r$
      $H_{i,p}^r \leftarrow -\frac{1}{\log J}\sum_{j=1}^{J} a_{i,p,j}^r \log a_{i,p,j}^r$ for every token p
      $e_{i,p}^r \leftarrow \exp(-\gamma_e H_{i,p}^r P_{i,p}^r)$
    end if
  end for
  if $k = 0$ **then**
    $b_{i,p}^r \leftarrow e_{i,p}^r / \sum_{\ell=0}^{N} e_{\ell,p}^r$ for every view and token
    $\widetilde{w}_{i,p}^r \leftarrow \max(m_i b_{i,p}^r, \eta)$
    $w_{i,p}^r \leftarrow \widetilde{w}_{i,p}^r / \sum_{\ell=0}^{N} \widetilde{w}_{\ell,p}^r$
    Cache $w_{i,p}^r$ for all remaining steps of this stage
  end if
  $v_p^r \leftarrow \sum_{i=0}^{N} w_{i,p}^r v_{i,p}^r$ for every token p
  $z_{p,t_{k+1}}^r \leftarrow z_{p,t_k}^r + (t_{k+1} - t_k)v_p^r$
end for
return $z_{t_T}^r$

A.4 Inference Configuration

At inference, the Zero-1-to-3 backbone is loaded from `105000.ckpt` and generates 256^2 views using 50 DDIM sampling steps with a classifier-free guidance scale of 3.0. Each generated view is upsampled by $4\times$ with Real-ESRGAN (Wang et al., 2021) to 1024^2, followed by Lanczos resizing to 576×576 for trajectory evaluation and 3D generation. TRELLIS.2-4B uses 12 sampling steps for sparse-structure generation and 12 sampling steps for shape generation. Texture generation is conditioned on the resulting shape latent and the source image. Its decoder produces O-Voxel geometry with base-color, metallic, roughness, and opacity attributes.

B Data Preparation

B.1 Dataset Splits and Asset Selection

We reserve 2% of the valid Objaverse-XL object pool as validation candidates and keep the resulting training and validation partitions disjoint at the object level. For the primary comparison, we select 400 high-quality Toys4K objects and 100 GSO objects. Every evaluated method uses exactly the same Toys4K and GSO object sets. The GSO ablation subset is sampled separately at the object level and is shared by all ablation configurations.

B.2 Rendering and Image Preprocessing

For each training object, we render four views using the nvdiffrast-based TRELLIS.2 PBR renderer at relative azimuths of $\{0°, 90°, 180°, 270°\}$, with a camera radius of 10 and a $54°$ field of view. During preprocessing, the starting azimuth is sampled uniformly from $[0°, 360°)$, the shared pitch from $[-15°, 15°]$, and the illumination from the available HDRI environment maps. The four views share the sampled illumination and a fixed white background. Source and reference renderings are stored at 512^2.

C Additional Experimental Details

C.1 Evaluation Protocol and Metrics

We apply the same input-anchored four-view evaluation protocol to every method. The prediction and ground truth assets are rendered at 512^2 from four shared cameras uniformly spaced in azimuth and anchored at the recorded input view.

For semantic evaluation, CLIP is the mean cosine similarity between ground-truth RGB and predicted RGB features rendered from matching cameras. ULIP-2 (Xue et al., 2024) measures the cosine similarity between the input-image representation and a $10{,}000$-point colored surface sample of the aligned prediction, without an additional orientation search. For rendered fidelity, PSNR, SSIM, and VGG-based LPIPS are computed over the full frame for both RGB and normal maps. We first average view-level measurements for each object and then aggregate over objects. All entries are reported as the sample mean and sample standard deviation, with the latter computed using the $n-1$ denominator.

Let $V = 4$, let G_v and P_v denote the ground-truth and predicted RGB renderings from matched camera v, and let ϕ_{C} be the normalized CLIP image encoder (Radford et al., 2021). The reported CLIP score is:

$$S_{\mathrm{CLIP}} = \frac{1}{V}\sum_{v=1}^{V} \frac{\phi_{\mathrm{C}}(G_v)^{\mathsf{T}}\phi_{\mathrm{C}}(P_v)}{\|\phi_{\mathrm{C}}(G_v)\|_2\|\phi_{\mathrm{C}}(P_v)\|_2}. \tag{17}$$

CLIP-N uses the same CLIP image encoder (Radford et al., 2021) to compute the mean cosine similarity between ground-truth RGB renderings and predicted normal maps at matched viewpoints. For a colored surface point cloud $\mathcal{P}$ sampled from the aligned prediction, the ULIP-2 score is:

$$S_{\mathrm{ULIP2}} = \cos\big(\phi_I^{\mathrm{U}}(I_0), \phi_P^{\mathrm{U}}(\mathcal{P})\big), \tag{18}$$

where ϕ_I^{U} and ϕ_P^{U} are its image and point-cloud encoders.

For either RGB or normal maps, let $X_v, Y_v \in [0,1]^{3\times H\times W}$ denote a matched prediction-reference pair. Pixel fidelity is measured by:

$$\mathrm{MSE}_v = \frac{1}{3HW}\|X_v - Y_v\|_2^2, \tag{19}$$

$$\mathrm{PSNR}_v = 10\log_{10}\left(\frac{1}{\mathrm{MSE}_v}\right). \tag{20}$$

For local means μ_X, μ_Y, variances σ_X^2, σ_Y^2, and covariance σ_{XY}, structural similarity is:

$$\mathrm{SSIM}_v = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)}, \tag{21}$$

averaged over spatial windows and channels. With normalized VGG features $\hat{f}_l$ and learned channel weights w_l, perceptual distance is:

$$\mathrm{LPIPS}_v = \sum_l \frac{1}{H_l W_l}\sum_{h,w}\left\| w_l \odot \left(\hat{f}_l(X_v)_{h,w} - \hat{f}_l(Y_v)_{h,w}\right)\right\|_2^2. \tag{22}$$

For any view-level metric m_v, its object score is $\bar{m}_o = V^{-1}\sum_{v=1}^{V} m_v$. Across n evaluated objects, we report:

$$\mu_m = \frac{1}{n}\sum_{o=1}^{n}\bar{m}_o, \qquad s_m = \sqrt{\frac{1}{n-1}\sum_{o=1}^{n}(\bar{m}_o - \mu_m)^2}. \tag{23}$$

C.2 Baseline Configurations

For each comparison baseline, we use the latest official model release available at the time of evaluation and its official default inference configuration, without baseline-specific hyperparameter tuning. All methods receive identically preprocessed inputs, and their generated assets are evaluated using the protocol described in Section C.1.

C.3 Ablation Configurations

All ablations are performed only at inference time and reuse the trained components and inference settings of the full model. Every configuration that retains the adapter loads the same checkpoint. In *w/o Adapter*, adapter corrections are disabled while trajectory selection and adaptive fusion remain active. In *w/o adaptive weighting*, the adapter is retained, but both the consistency multipliers and token-wise entropy weighting are replaced by uniform weights of $1/(N+1)$ across the source view and the N generated views in the sparse-structure and shape stages. The *w/o both* variant applies both changes while retaining view expansion and trajectory selection. Native TRELLIS.2 bypasses view expansion and generates directly from the source image. All other settings are held fixed.

D Additional Results and Analysis

D.1 Additional Qualitative Comparisons

Figures 6–8 provide additional generated assets with material components and renderings under different lighting. Figures 9–12 show enlarged textured and normal renderings. Boxed insets indicate the input images.



Figure 6: Additional assets generated by CVE3D, with material components and renderings under different lighting conditions. Insets show the input images.

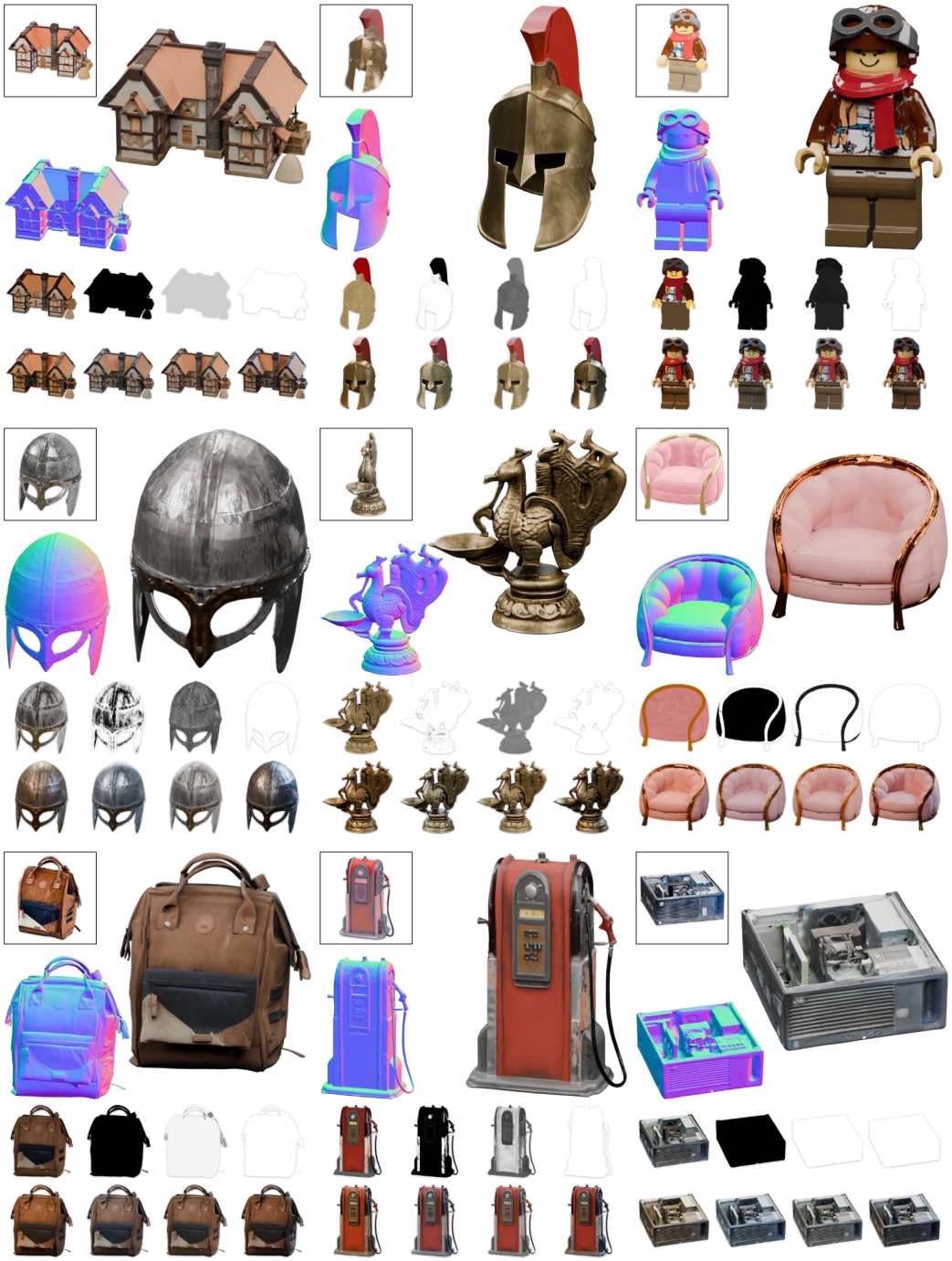


Figure 7: Additional assets generated by CVE3D, with material components and renderings under different lighting conditions. Insets show the input images.



Figure 8: Additional assets generated by CVE3D, with material components and renderings under different lighting conditions. Insets show the input images.



Figure 9: Additional assets generated by CVE3D, shown with textured and normal renderings. Insets show the input images.

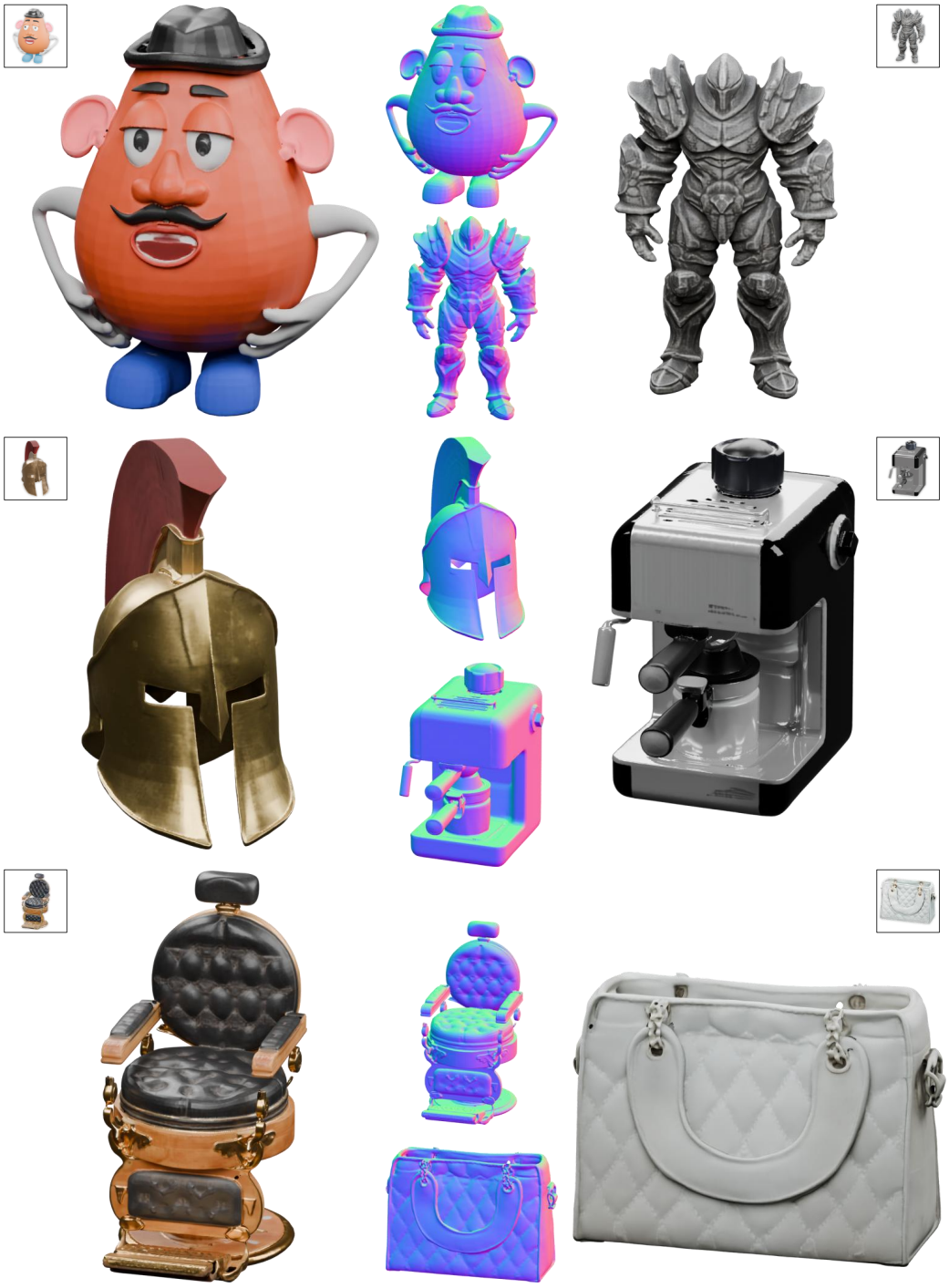


Figure 10: Additional assets generated by CVE3D, shown with textured and normal renderings. Insets show the input images.



Figure 11: Additional assets generated by CVE3D, shown with textured and normal renderings. Insets show the input images.



Figure 12: Additional assets generated by CVE3D, shown with textured and normal renderings. Insets show the input images.

D.2 Additional Quantitative Comparisons

Toys4K Protocol. We conduct three additional runs, each randomly sampling 400 Toys4K objects for asset generation and evaluation, yielding 1,200 records covering 965 distinct objects. Excluding the primary evaluation, we report object-level statistics and equally weighted averages of the three run means. Two-sided Student-t confidence intervals are computed as $\bar{x} \pm t_{1-\alpha/2,n-1} s/\sqrt{n}$, where s is the sample standard deviation and n counts the statistical units at each aggregation level ($n = 3$ for run-level estimates).

Table 6: Toys4K supplementary evaluation: object-level statistics over 965 distinct objects. Missing values are omitted per metric; Min–Max denotes the range of distinct objects. Green indicates a mean that outperforms the best baseline in Table 1; lighter green indicates a mean that outperforms the second-best but not the best baseline.

Metric	Mean	95% CI	99% CI	Min–Max
CLIP	0.9015	[0.8983, 0.9046]	[0.8973, 0.9056]	[0.6369, 0.9854]
CLIP-N	0.6921	[0.6883, 0.6959]	[0.6871, 0.6972]	[0.4348, 0.8662]
ULIP-2	0.3438	[0.3395, 0.3481]	[0.3381, 0.3494]	[0.0982, 0.5133]
RGB PSNR	22.2254	[21.9457, 22.5050]	[21.8576, 22.5932]	[12.1834, 36.5501]
RGB SSIM	0.9116	[0.9094, 0.9137]	[0.9087, 0.9144]	[0.7469, 0.9914]
RGB LPIPS	0.0872	[0.0851, 0.0893]	[0.0845, 0.0900]	[0.0122, 0.2851]
Normal PSNR	25.8644	[25.6907, 26.0380]	[25.6360, 26.0928]	[21.4651, 44.4481]
Normal SSIM	0.9257	[0.9242, 0.9273]	[0.9237, 0.9278]	[0.8233, 0.9949]
Normal LPIPS	0.0762	[0.0745, 0.0779]	[0.0739, 0.0785]	[0.0094, 0.1748]

Table 7: Toys4K supplementary evaluation across three evaluation runs, each involving newly sampled objects and repeated asset generation and evaluation. Min–Max denotes the range of run means. Green indicates a mean that outperforms the best baseline in Table 1; lighter green indicates a mean that outperforms the second-best but not the best baseline.

Metric	Mean	95% CI	99% CI	Min–Max
CLIP	0.9006	[0.8964, 0.9048]	[0.8909, 0.9103]	[0.8986, 0.9016]
CLIP-N	0.6919	[0.6830, 0.7007]	[0.6715, 0.7123]	[0.6886, 0.6957]
ULIP-2	0.3439	[0.3314, 0.3563]	[0.3152, 0.3726]	[0.3403, 0.3496]
RGB PSNR	22.3266	[21.1671, 23.4861]	[19.6520, 25.0012]	[21.8984, 22.8242]
RGB SSIM	0.9127	[0.9062, 0.9192]	[0.8978, 0.9276]	[0.9098, 0.9149]
RGB LPIPS	0.0862	[0.0796, 0.0929]	[0.0709, 0.1016]	[0.0842, 0.0893]
Normal PSNR	25.8672	[25.3807, 26.3537]	[24.7451, 26.9894]	[25.6508, 26.0321]
Normal SSIM	0.9261	[0.9217, 0.9304]	[0.9160, 0.9361]	[0.9240, 0.9272]
Normal LPIPS	0.0759	[0.0712, 0.0807]	[0.0650, 0.0868]	[0.0743, 0.0780]

Across the three evaluation runs, the run means remain close, suggesting that the observed performance is not specific to a single sampled subset or generation run. The object-level and run-level summaries also yield similar average performance. Both summaries remain favorable in semantic alignment and rendered-view fidelity relative to the baseline means in Table 1. These comparisons provide additional context for the performance of CVE3D, although the different evaluation subsets preclude a controlled comparison with those baselines. Taken together, the repeated evaluations support the stability of the observed results across object sampling and asset generation.

GSO Protocol. We conduct three additional runs, each randomly sampling 100 GSO objects for asset generation and evaluation, yielding 300 records covering 249 distinct objects. Excluding the primary evaluation, we report object-level statistics and equally weighted averages of the three run means. Confidence intervals follow the same Student-t procedure as for Toys4K, with $n = 3$ for run-level estimates.

Table 8: Object-level statistics for the 249 distinct GSO objects in the supplementary evaluation. Missing values are omitted per metric. Green indicates a mean that outperforms the best baseline in Table 1; lighter green indicates a mean that outperforms the second-best but not the best baseline.

Metric	Mean	95% CI	99% CI	Min–Max
CLIP	0.8870	[0.8812, 0.8928]	[0.8794, 0.8946]	[0.7164, 0.9783]
CLIP-N	0.6179	[0.6101, 0.6257]	[0.6077, 0.6281]	[0.4458, 0.7969]
ULIP-2	0.2917	[0.2830, 0.3003]	[0.2803, 0.3030]	[0.1288, 0.4951]
RGB PSNR	19.0031	[18.6293, 19.3769]	[18.5105, 19.4957]	[12.4132, 30.3443]
RGB SSIM	0.8406	[0.8348, 0.8464]	[0.8330, 0.8483]	[0.6768, 0.9485]
RGB LPIPS	0.1377	[0.1325, 0.1428]	[0.1309, 0.1445]	[0.0388, 0.2651]
Normal PSNR	24.0368	[23.7213, 24.3523]	[23.6210, 24.4526]	[19.1123, 31.7959]
Normal SSIM	0.8820	[0.8768, 0.8872]	[0.8751, 0.8888]	[0.7360, 0.9655]
Normal LPIPS	0.1099	[0.1064, 0.1135]	[0.1052, 0.1146]	[0.0276, 0.2118]

Table 9: GSO supplementary evaluation across three evaluation runs, each involving newly sampled objects and repeated asset generation and evaluation. Min–Max denotes the range of run means. Green indicates a mean that outperforms the best baseline in Table 1; lighter green indicates a mean that outperforms the second-best but not the best baseline.

Metric	Mean	95% CI	99% CI	Min–Max
CLIP	0.8897	[0.8847, 0.8946]	[0.8783, 0.9011]	[0.8877, 0.8917]
CLIP-N	0.6187	[0.6110, 0.6264]	[0.6009, 0.6364]	[0.6154, 0.6215]
ULIP-2	0.2957	[0.2934, 0.2980]	[0.2904, 0.3010]	[0.2947, 0.2966]
RGB PSNR	19.0910	[18.6410, 19.5411]	[18.0529, 20.1292]	[18.8840, 19.2206]
RGB SSIM	0.8410	[0.8320, 0.8501]	[0.8201, 0.8619]	[0.8374, 0.8447]
RGB LPIPS	0.1359	[0.1332, 0.1386]	[0.1297, 0.1421]	[0.1347, 0.1367]
Normal PSNR	24.0726	[23.7496, 24.3956]	[23.3276, 24.8176]	[23.9244, 24.1673]
Normal SSIM	0.8822	[0.8757, 0.8886]	[0.8673, 0.8971]	[0.8797, 0.8849]
Normal LPIPS	0.1097	[0.1047, 0.1146]	[0.0982, 0.1212]	[0.1077, 0.1117]

The three GSO evaluation runs exhibit limited variation in their mean scores, providing further evidence that the observed results are not confined to one favorable subset or generation run. Both aggregation levels show favorable semantic alignment and rendering fidelity relative to Table 1. These cross-subset comparisons are descriptive rather than controlled, but the agreement across repeated evaluations supports the reliability of the reported performance under repeated object sampling and generation.

E Discussion and Limitations

CVE3D expands conditioning evidence but cannot eliminate the ambiguity of single-image 3D generation. Errors in provisional geometry or preceding views may propagate through view expansion, while semantic similarity does not guarantee geometric accuracy. CGA-Fusion assigns relative weights rather than calibrated reliability, so poor or out-of-distribution views may still influence generation. Provisional generation and best-of-K trajectory sampling also increase latency over a native single-view pipeline.