

OREO: Fidelity Alignment in 3D Generation via On-the-fly Rendering-Editing Optimization

Zhiyuan Ma^{1,2}, Wenbo Hu^{2†}, Wang Zhao²,
Pengfei Wang¹, Ying Shan², and Lei Zhang^{1†}

¹ The Hong Kong Polytechnic University

² Tencent ARC Lab

Abstract. Despite recent advancements in 3D generation, models often struggle to produce assets with high visual fidelity. To bridge this gap, we propose OREO, an alignment framework that enhances the realism of 3D generators by leveraging rich 2D diffusion priors. Instead of relying on static datasets, OREO establishes a dynamic optimization loop that produces on-the-fly pseudo ground truths. At its core, OREO combines two components: (i) Reinforced Editing, which uses an image editor to refine rendered views of the 3D output, improving the visual fidelity of rendered views while preserving geometric structure; and (ii) Contrastive Distillation, which treats the edited views as positive and the original renderings as negative anchors, distilling the fidelity gap into the 3D generator via a latent contrastive objective. Experiments demonstrate that OREO effectively improves upon pre-trained baselines, producing 3D assets with enhanced visual realism. Our project page is at <https://theericma.github.io/oreo/>.

Keywords: 3D Generation · Alignment · Image Diffusion Prior

1 Introduction

Recent 3D generators, such as Trellis [34] and the Hunyuan3D series [13, 15, 39, 49], have greatly simplified the 3D content creation process. However, despite their strong performance in geometry generation, achieving high visual fidelity remains challenging. Generated assets often lack intricate textures and fine-grained details observed in real-world subjects. This limitation largely stems from the scarcity of high-quality 3D data with diverse appearances, which pales in comparison to the vast visual knowledge encapsulated in billion-scale 2D imagery. Consequently, 3D generative models may not fully capture the nuanced visual characteristics required for photorealistic rendering.

To bridge this gap, we propose to distill the rich visual priors of pre-trained 2D diffusion models into 3D generators. These models encapsulate a deep understanding of visual fidelity and fine-grained details, far surpassing the scope of existing 3D datasets. To this end, we introduce the On-the-fly Rendering Editing Optimization (OREO) framework, as illustrated in Fig. 1. The core idea

[†]Corresponding authors

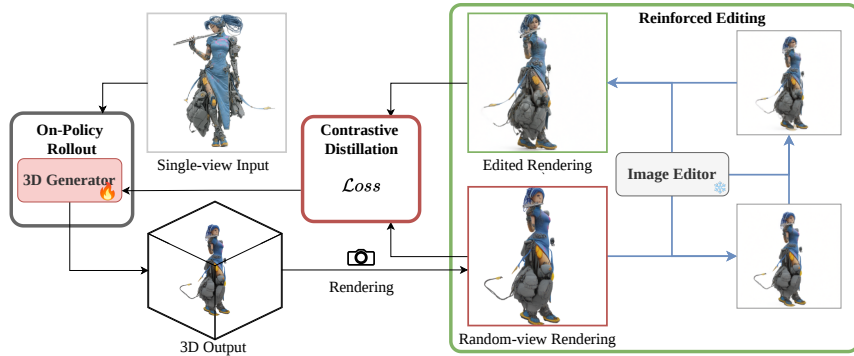


Fig. 1: Teaser. To enhance visual fidelity of 3D generation, our framework first **re-inforces** rendered views via image editing, then distills the gap back to the training generator with a **latent contrastive** objective. Detailed workflow in Fig. 2.

is to establish a dynamic alignment mechanism: instead of learning from static datasets, the generator improves by learning from pseudo ground truths generated on-the-fly. Specifically, OREO implements a Render-Edit-Optimize loop, where the model renders views from its current output, receives corrective edits to enhance realism, and uses these refined views as supervision targets.

At the core of this loop is the Reinforced Editing algorithm, which generates high-quality supervision signals. To ensure reliable guidance, Reinforced Editing prioritizes structure preservation, focusing on improving visual fidelity while maintaining the original spatial layout. Inspired by recent inversion-free editing techniques, this approach mitigates the unwanted geometric deviations common in standard editing methods, ensuring that the generated pseudo-GTs remain geometrically aligned with the underlying 3D mesh. To turn these refined targets into effective supervision, we propose Contrastive Distillation, which treats the edited view as a positive and the original rendering as a negative, and distills the fidelity gap into the 3D generator through a latent-space contrastive objective. Experiments on the state-of-the-art Trellis model demonstrate that OREO effectively improves its performance, producing 3D assets with superior visual realism. In summary, our contributions are as follows:

1. We introduce OREO, a novel alignment framework that leverages pre-trained 2D diffusion priors to enhance the fidelity of 3D generators through an on-the-fly optimization loop with dynamic pseudo ground truths.
2. At its core, we propose Reinforced Editing, an inversion-free algorithm that improves the visual fidelity of rendered views while preserving structure, coupled with Contrastive Distillation that turns the edited/original views into latent-space positive/negative anchors for effective generator post-training.
3. We validate the effectiveness of OREO on state-of-the-art 3D generators. Results show that our method effectively improves visual fidelity while preserving geometric integrity.

2 Related Work

Learning-based 3D Generation and Generator Post-training. Recent learning-based 3D generators synthesize assets feed-forward in latent spaces [5, 9, 16–18, 20, 21, 31, 34, 36, 37, 40, 44, 46, 49], but their reliance on synthetic data such as Objaverse [7] caps visual fidelity and texture detail. OREO addresses this by transferring appearance priors from pretrained 2D diffusion models: it uses 2D image editing to lift rendered views into higher-fidelity, structure-preserving edited views, and treats the discrepancy as online supervision for post-training.

2D Image Editing for Fidelity-Enhancing Supervision. Effective supervision requires edits that enhance fidelity while preserving viewpoint and structure. Training-based image editors [1, 28, 45, 48] inject appearance details but do not constrain viewpoint or layout, and inversion-based training-free methods [11, 24] still suffer from inversion errors. Inversion-free approaches [6, 47] instead couple source and target trajectories, avoiding inversion altogether; we build our Reinforced Editing on this line.

2D Priors for 3D Editing and Generation. Although 2D priors have been explored for 3D editing and generation, existing methods do not directly meet the goal of post-training a feed-forward 3D generator. On the one hand, 3D editing and geometry-conditioned editing methods, such as Instruct-NeRF2NeRF [10], DreamEditor [50], GaussianEditor [30], DFFSplat [14], Image Sculpting [41], MvDrag3D [2], GeoDiffusion [3], and Omni-3DEdit [4], mainly target individual instances or scenes and typically rely on per-scene optimization, making it difficult to amortize the fidelity-enhancing prior into a generalizable 3D generator. On the other hand, score-distillation methods such as DreamFusion [25], Magic3D [19], ProlificDreamer [32], ScaleDreamer [23], Hybrid Fourier Score Distillation [38], and Progressive Rendering Distillation [22] are optimization-based text-to-3D methods whose input setting and inference pipeline differ from OREO’s learning-based image-to-3D generator post-training, making them unsuitable for direct quantitative comparison. DMD [42] is closer to a generator-learning paradigm and is therefore retained as a reasonable score-distillation ablation baseline; however, it still relies on implicit score-gradient supervision. In contrast, OREO explicitly constructs high-fidelity, structure-aligned edited views as pseudo-GT and uses stable online supervision to post-train the 3D generator.

3 Method

We present OREO, an alignment framework that enhances 3D visual fidelity via on-the-fly optimization. An overview of the full pipeline is illustrated in Fig. 2. We leverage a pre-trained 2D editing model to construct pseudo ground truths on the fly, overcoming the inability to train without 3D ground-truth supervision. Specifically, the generator performs on-policy sampling to obtain a denoised 3D asset at an intermediate step, which is then rendered into a 2D view from a sampled camera pose, as detailed in Sec. 3.2. We then apply the image

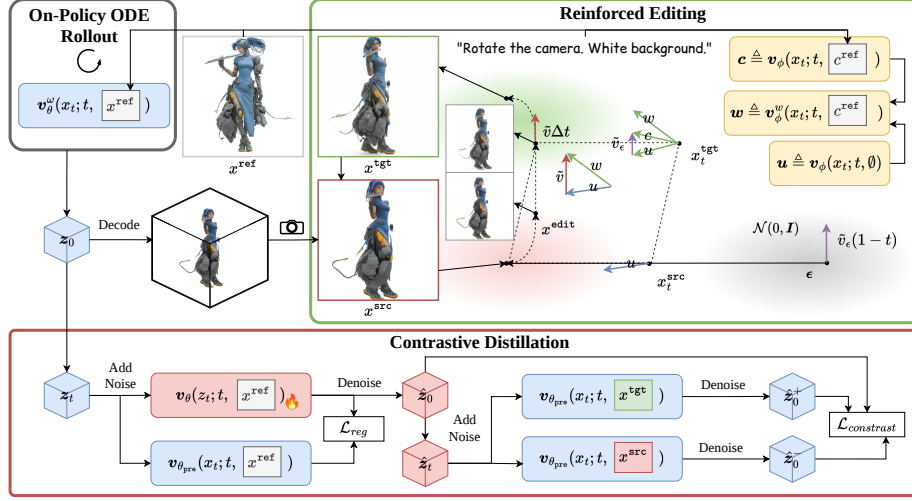


Fig. 2: Overview of OREO. The training generator produces a 3D asset via on-policy ODE rollout that is rendered into a source view x^{src} . **Reinforced Editing** turns x^{src} into a high-fidelity target x^{tgt} , and **Contrastive Distillation** uses $(x^{\text{src}}, x^{\text{tgt}})$ as a negative/positive pair to supervise the generator in the latent space.

editing prior with our proposed Reinforced Editing in Sec. 3.3 to enhance the view, improving visual fidelity while preserving structural consistency. Finally, using the enhanced view as a dynamic supervision signal, the training method in Sec. 3.4 distills the fidelity improvement back into the 3D generator.

3.1 Problem Formulation

Given a reference image x^{ref} , the 3D latent generator G_θ followed by a decoder $\mathcal{D}$ produces a 3D asset $\mathcal{A} = \mathcal{D}(G_\theta(x^{\text{ref}}))$. A renderer $\mathcal{P}$ then projects $\mathcal{A}$ into a 2D view $x^{\text{src}} = \mathcal{P}(\mathcal{A}, \pi)$ under camera pose π . In our training framework, a pre-trained 2D editor $\mathcal{E}_\phi$ then enhances this view into a higher-fidelity target $x^{\text{tgt}} = \mathcal{E}_\phi(x^{\text{src}}, x^{\text{ref}})$. The generator is optimized by minimizing the discrepancy between the two:

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{x^{\text{ref}}, \pi} [\mathcal{L}(x^{\text{src}}, x^{\text{tgt}})] \quad (1)$$

where $\mathcal{L}$ is a supervision loss that encourages improved visual fidelity.

3.2 Preliminaries

Flow Matching Framework and Notation. Flow Matching models generation as an ODE process that transports Gaussian noise to the data distribution:

$$dx_t = v(x_t; t, c)dt, \quad t \in [1, 0] \quad (2)$$

where x_t is the latent state at time t , and c is the condition (e.g., text or image). For better alignment to the condition c , Classifier-Free Guidance (CFG) is applied to the velocity field, using a null condition $\emptyset$ and a guidance scale w :

$$v^w(x_t; t, c) = v(x_t; t, \emptyset) + w \cdot (v(x_t; t, c) - v(x_t; t, \emptyset)) \quad (3)$$

A key property used in our method is single-step estimation of the clean sample $\hat{x}_0$ from the intermediate state x_t :

$$\hat{x}_0 = x_t - t \cdot v(x_t; t, c) \quad (4)$$

To distinguish modalities, we use z and θ for the 3D generator v_θ , where G_θ denotes its ODE sampling process and the pre-trained frozen weights are denoted $v_{\theta_{pre}}$. Similarly, we use x and ϕ for the 2D editor v_ϕ , where $\mathcal{E}_\phi$ denotes the multi-step editing process as described below.

Inversion-Free Image Editing. We leverage FlowEdit [6], an inversion-free method that modifies images by coupling source and target flow trajectories. Starting from x^{src} , it iteratively updates the editing variable via $x^{\text{edit}} \leftarrow x^{\text{edit}} - \Delta t \cdot \tilde{v}$, where the differential velocity is defined as:

$$\tilde{v} = v^{\text{tgt}}(x_t^{\text{tgt}}; t, c^{\text{tgt}}) - v^{\text{src}}(x_t^{\text{src}}; t, c^{\text{src}}) \quad (5)$$

where the **source branch** and the **target branch** is constructed as

$$x_t^{\text{src}} = (1 - t)x^{\text{src}} + t\epsilon, \quad x_t^{\text{tgt}} = x^{\text{edit}} + (x_t^{\text{src}} - x^{\text{src}}), \quad \epsilon \sim \mathcal{N}(0, I). \quad (6)$$

The editing process is performed over the last N_e steps of the full N -step ODE schedule. With integration step size $\Delta t = t_{i-1} - t_i$, x^{edit} evolves into x^{tgt} , which transfers the image from being consistent with c^{src} to following c^{tgt} without the need for costly inversion that maps the image back to noise, thereby preserving structural fidelity.

3.3 Constructing On-the-fly Pseudo-GTs via Reinforced Editing

A critical step in our pipeline is transforming the rendered view x^{src} into a high-fidelity pseudo ground truth x^{tgt} . Existing image editing models [33] can leverage their learned priors to make rendered views more realistic, but often alter the camera perspective or object pose, as analyzed in Sec. 4.2. We propose Reinforced Editing to enable structure-preserving enhancement on top of pre-trained editing models. Different from the original FlowEdit [6] that transfers between separate conditions c^{src} and c^{tgt} in Eq. 5, our goal is to enhance the rendered view to better resemble a novel view of x^{ref} , where no independent c^{src} exists. We thus guide the target branch toward $c^{\text{ref}} = c^{\text{tgt}}$ and replace the source branch with an unconditional prediction:

$$\tilde{v} = v_\phi^w(x_t^{\text{tgt}}; t, c^{\text{ref}}) - v_\phi(x_t^{\text{src}}; t, \emptyset), \quad (7)$$

Algorithm 1 Reinforced Editing

Require: Rendered view x^{src} , Reference condition c^{ref} , Editor $\mathcal{E}_\phi$, Guidance scale w , Editing steps N_e , Total steps N

Ensure: Pseudo-GT x^{tgt}

- 1: Initialize $x^{\text{edit}} \leftarrow x^{\text{src}}$, Sample noise $\epsilon \sim \mathcal{N}(0, I)$
- 2: Select time steps $\{t_i\}_{i=0}^{N_e}$ as the last N_e steps from the full schedule of N steps
- 3: **for** $i = 0$ to $N_e - 1$ **do** ▷ FlowEdit: Backward Euler Integration
- 4: $t \leftarrow t_i, \Delta t \leftarrow t_{i+1} - t_i$
- 5: $x_t^{\text{src}} \leftarrow (1 - t)x^{\text{src}} + t\epsilon$
- 6: $x_t^{\text{tgt}} \leftarrow x^{\text{edit}} + (x_t^{\text{src}} - x^{\text{src}})$ ▷ Eq. 6
- // Reinforced Editing (Sec. 3.3)
- 7: $\tilde{v} \leftarrow v_\phi^w(x_t^{\text{tgt}}; t, c^{\text{ref}}) - v_\phi(x_t^{\text{src}}; t, \emptyset)$ ▷ Eq. 7
- 8: $x^{\text{edit}} \leftarrow x^{\text{edit}} - \tilde{v}\Delta t$ ▷ FlowEdit: Euler Step
- // Noise Update
- 9: $\epsilon \leftarrow \epsilon - (v_\phi(x_t^{\text{tgt}}; t, c^{\text{ref}}) - v_\phi(x_t^{\text{tgt}}; t, \emptyset)) \cdot (1 - t)$ ▷ Eq. 8
- 10: **end for**
- 11: **return** $x^{\text{tgt}} \leftarrow x^{\text{edit}}$

where v_ϕ is the velocity field of the pre-trained 2D editor, w is the classifier-free guidance scale [12], and c^{ref} encodes both the reference image x^{ref} and a text prompt indicating a viewpoint change, as illustrated in Fig. 2.

Noise Update. The editing effect may be insufficient because the shared noise ϵ in Eq. 6 is randomly sampled in each iteration, whose stochasticity can weaken the editing strength [35]. To address this, we dynamically update ϵ at each step by injecting the classifier-free guidance signal into the noise:

$$\epsilon \leftarrow \epsilon - (v_\phi(x_t^{\text{tgt}}; t, c^{\text{ref}}) - v_\phi(x_t^{\text{tgt}}; t, \emptyset)) \cdot (1 - t) \quad (8)$$

This update aligns the noise with the conditional direction, strengthening the editing effect while preserving fine-grained structural details. The complete procedure is summarized in Algorithm 1.

3.4 Contrastive Distillation with Pseudo-GTs

To instantiate the abstract loss $\mathcal{L}$ in Eq. 1 for the 3D latent generator G_θ , we lift the supervision from the pixel space of $x^{\text{src}}, x^{\text{tgt}}$ into the latent space of the pretrained generator $v_{\theta_{pre}}$, and cast it as a contrastive objective: we treat the generator’s predicted clean latent $\hat{z}_0$ as an *anchor*, with x^{tgt} inducing a positive and x^{src} inducing a negative, pulling the anchor toward the positive while pushing it away from the negative. This closes the Render-Edit-Optimize loop.

Per-Iteration Setup. Concretely, in each training iteration, given a reference image x^{ref} , we roll out a clean latent $z_0 = G_\theta(x^{\text{ref}})$ via ODE sampling. We then sample a camera pose π and render $x^{\text{src}} = \mathcal{P}(\mathcal{D}(z_0), \pi)$. From it we obtain a pseudo-GT x^{tgt} via Reinforced Editing in Algorithm 1. Finally, we sample a

Algorithm 2 Closed-loop Optimization via Dynamic Pseudo-GTs**Require:** Generator v_θ , frozen pretrained velocity $v_{\theta_{pre}}$, 2D editor $\mathcal{E}_\phi$, learning rate η **Ensure:** Optimized parameters θ^*

```

1: while not converged do
2:   Given reference image  $x^{\text{ref}}$ 
3:    $z_0 = G_\theta(x^{\text{ref}})$  ▷ Rollout: Generate clean latent
4:   Sample camera pose  $\pi$ 
5:    $x^{\text{src}} = \mathcal{P}(\mathcal{D}(z_0), \pi)$  ▷ Render: Decode and project
6:   Obtain  $x^{\text{tgt}}$  via Algorithm 1 from  $x^{\text{src}}$  and  $x^{\text{ref}}$  ▷ Reinforced Editing
7:   Sample  $t \sim \mathcal{U}(0, 1)$ 
8:    $z_t = (1 - t)z_0 + t\epsilon$ ,  $\epsilon \sim \mathcal{N}(0, I)$  ▷ Noise Injection
9:   Predict  $\hat{z}_0$  from  $z_t$  using  $v_\theta$  ▷ Prediction: Denoise latent, Eq. 4
10:   $\hat{z}_t = (1 - t)\hat{z}_0 + t\epsilon'$ ,  $\epsilon' \sim \mathcal{N}(0, I)$  ▷ Re-noise anchor prediction
11:   $z_0^+ = \hat{z}_t - t \cdot v_{\theta_{pre}}(\hat{z}_t; t, x^{\text{tgt}})$  ▷ Positive, Eq. 10
12:   $z_0^- = \hat{z}_t - t \cdot v_{\theta_{pre}}(\hat{z}_t; t, x^{\text{src}})$  ▷ Negative, Eq. 10
13:   $\mathcal{L}_{\text{contrast}} = \|\hat{z}_0 - z_0^+\|^2 - \|\hat{z}_0 - z_0^-\|^2$  ▷ Latent contrastive loss, Eq. 11
14:   $\mathcal{L}_{\text{reg}} = \|v_\theta(z_t) - v_{\theta_{pre}}(z_t)\|^2$  ▷ Regularization
15:   $\theta \leftarrow \theta - \eta \nabla_\theta (\mathcal{L}_{\text{contrast}} + \lambda \mathcal{L}_{\text{reg}})$  ▷ Update generator
16: end while

```

timestep $t \sim \mathcal{U}(0, 1)$, perturb the latent as $z_t = (1 - t)z_0 + t\epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$, and recover the single-step prediction

$$\hat{z}_0 = z_t - t \cdot v_\theta(z_t; t, x^{\text{ref}}), \quad (9)$$

which serves as the anchor in our contrastive objective below.

Latent-Space Contrastive Supervision. Inspired by [43], we re-noise the anchor with a fresh draw $\epsilon' \sim \mathcal{N}(0, I)$ as $\hat{z}_t = (1 - t)\hat{z}_0 + t\epsilon'$, and run the frozen pretrained generator in two passes that share $\hat{z}_t$ but differ in their visual condition,

$$z_0^+ = \hat{z}_t - t \cdot v_{\theta_{pre}}(\hat{z}_t; t, x^{\text{tgt}}), \quad z_0^- = \hat{z}_t - t \cdot v_{\theta_{pre}}(\hat{z}_t; t, x^{\text{src}}). \quad (10)$$

Both z_0^+ and z_0^- are detached from the computational graph and serve as the positive and the negative, respectively. Under x^{tgt} the pretrained generator denoises toward the edited pseudo-GT, while under x^{src} it denoises toward the un-edited rendering. The contrastive objective then pulls the anchor $\hat{z}_0$ toward z_0^+ and pushes it away from z_0^- ,

$$\mathcal{L}_{\text{contrast}} = \|\hat{z}_0 - z_0^+\|^2 - \|\hat{z}_0 - z_0^-\|^2. \quad (11)$$

To preserve geometric stability, we further regularize the velocity field deviation from the pre-trained prior,

$$\mathcal{L}_{\text{reg}} = \|v_\theta(z_t) - v_{\theta_{pre}}(z_t)\|^2, \quad (12)$$

giving the total objective $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{contrast}} + \lambda \mathcal{L}_{\text{reg}}$. The overall training procedure is summarized in Algorithm 2.

Table 1: Quantitative comparison of 2D feedback sources. We report CLIP and DINO similarity to the reference x^{ref} for fidelity and to the source view x^{src} for content preservation, together with Mask IoU for viewpoint consistency. Details are in Sec. 4.2.

Method	w.r.t. x^{ref}		w.r.t. x^{src}		IoU $\uparrow$
	CLIP Sim. $\uparrow$	DINO Sim. $\uparrow$	CLIP Sim. $\uparrow$	DINO Sim. $\uparrow$	
Unedited	0.7613	0.7915	N/A	N/A	N/A
NanoBanana Pro	0.8803	0.9001	0.8199	0.8404	0.6830
Qwen-Edit (Orig.)	0.7753	0.8269	0.7882	0.8084	0.6374
Ours	0.7992(+0.0379)	0.8214(+0.0299)	0.8833	0.9099	0.9520

4 Experiments

4.1 Experimental Setup

Datasets. Existing 3D benchmarks such as Objaverse [7] subsets focus on everyday objects, where pre-trained baselines already do well and fidelity gaps stay hidden. To cover the more complex needs of image-to-3D users, especially imaginative and diverse concept-design tasks, we curate a *Conceptual Design Dataset* of 2,396 in-the-wild reference images under-served by existing 3D priors. We hold out 100 references for evaluation and use the remaining 2,296 for training. OREO uses these images alone and requires no paired 3D ground truth. Curation and filtering details are in the *Supp. Material*. For a general-distribution reference, we additionally evaluate on Google Scanned Objects (GSO) [8].

Implementation Details. We use Trellis [34] as the 3D backbone and Qwen-Image-Edit [33] for Reinforced Editing, whose hyper-parameters are given in Sec. 4.2. Training details are provided in the *Supp. Material*.

4.2 Validating the Editing Feedback

The 3D generator is supervised by pseudo-GTs from a 2D editor, so the quality of these edits directly bounds what the 3D loop can achieve. We therefore first evaluate the editor’s feedback at the 2D level, before 3D training.

Evaluation Protocol. For each reference in the Conceptual Design Dataset, we generate a 3D asset with Trellis and render 5 views at azimuths -90° , -45° , 0° , 45° , and 90° as source views x^{src} . To evaluate the editing performance, we adopt three metrics. To measure the multi-view fidelity improvement, we compute the CLIP-ViT-L/14 [27] and DINOv3-ViT-L/16 [29] embedding similarity of each rendered view to the reference x^{ref} , and report the mean over the 5 views before and after editing. To check content preservation, we additionally report the similarity to x^{src} . Besides, to quantify the consistency of object pose, we use Mask IoU, with foreground masks extracted by U²-Net [26] as in Trellis.

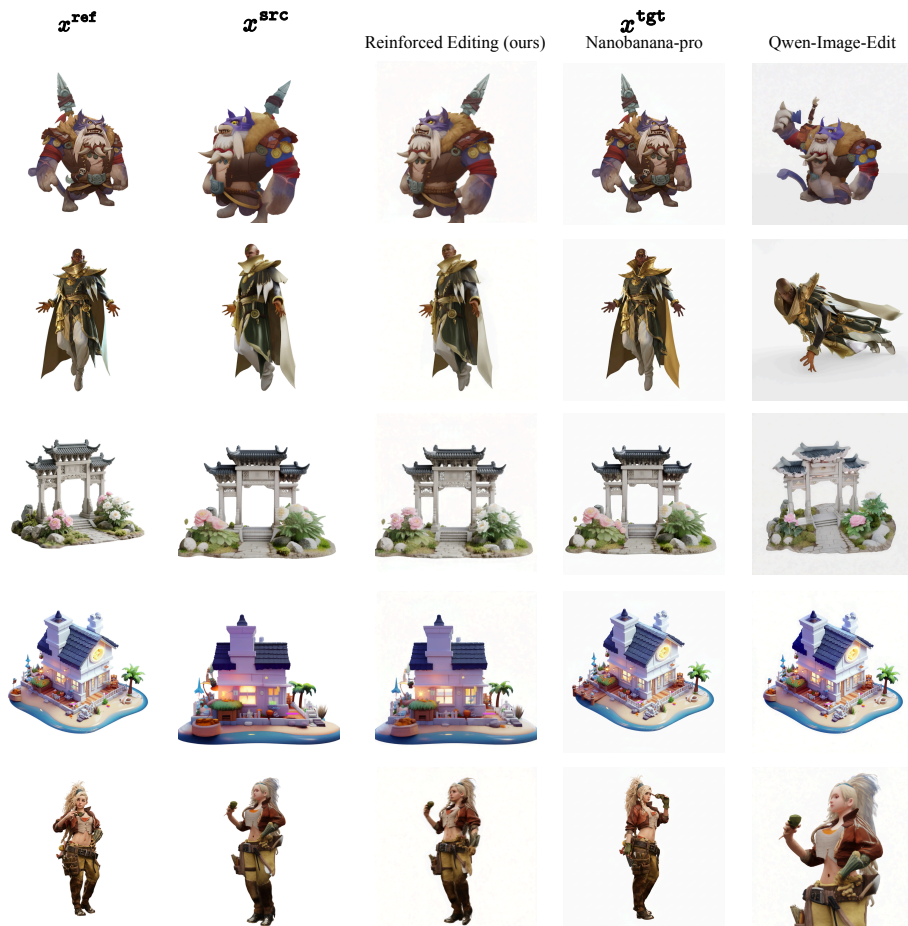


Fig. 3: Qualitative comparison of 2D feedback sources. An ideal x^{tgt} keeps the viewpoint of x^{src} but lifts its fidelity toward x^{ref} ; we compare Reinforced Editing (Ours), NanoBanana Pro, and Qwen-Image-Edit. Details are discussed in Sec. 4.2.

Compared Methods. We compare our Reinforced Editing (RE) against the original Qwen-Image-Edit [33] pipeline and NanoBanana Pro. The setup for the compared methods is provided in the *Supp. Material*.

Qualitative Analysis. Fig. 3 visualizes the feedback quality. A critical challenge is maintaining both the camera perspective and content fidelity of the source view x^{src} . NanoBanana Pro primarily fails at controlling the viewpoint: when x^{src} differs substantially in viewpoint from x^{ref} , it tends to abandon the x^{src} viewpoint and revert to x^{ref} . For instance, in row 1, x^{src} shows the orc warrior from a side view, yet NanoBanana Pro’s output snaps back to a frontal view nearly identical to x^{ref} . Even when the viewpoint is roughly preserved, it is not faithful to the local content of the source view, e.g., the mage’s cape in

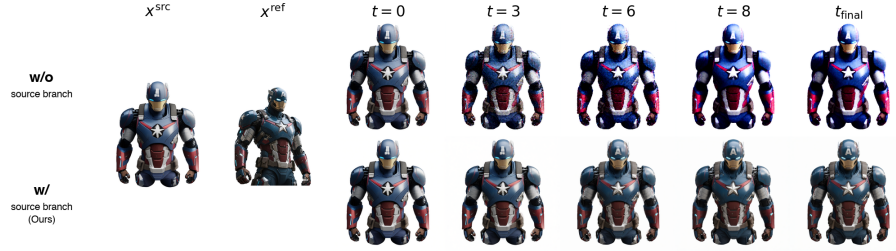


Fig. 4: Editing trajectories with and without the source branch. Removing the source branch (top) leads to progressive color over-saturation and structural drift; keeping it (bottom, Ours) stabilizes the trajectory. Details in Sec. 4.2.

row 2 is altered. In contrast, Reinforced Editing injects high-fidelity textures while strictly adhering to the source perspective and content layout, ensuring geometrically aligned supervision. Qwen-Image-Edit fails at the pose level in an even more pronounced way: the orc warrior’s hand pose is altered in row 1, the mage collapses into a tilted, falling posture in row 2, and the cowgirl in row 5 is cropped from a full-body figure into a half-body portrait. It also drifts on the background, which is occasionally washed to a plain gray tone.

Quantitative Results. As shown in Table 1, the *Unedited* row represents the raw rendered view x^{src} , quantifying its initial alignment with the reference x^{ref} . Baseline editors can achieve high similarity to x^{ref} but deviate substantially from x^{src} , which is evidenced by their low Mask IoU. In contrast, RE maintains high similarity to x^{src} while still improving semantic fidelity.

Ablation Study on RE Parameters. We further study two key hyper-parameters of RE: the editing ratio N_e/N and the absolute editing steps N_e , as discussed in Sec. 3.3. As shown in Fig. 5(a) and Fig. 6(a), a low ratio of 0.25 barely edits the source view: the mecha warrior in row 1 and the king in row 2 remain almost identical to x^{src} , with limited detail enhancement. Conversely, a full ratio of 1.0 causes severe deviations from the source view; the mecha warrior is rendered at a shrunk scale with an altered pose, and the king loses part of its cape and body, leading to identity mismatch. In contrast, a ratio of 0.75 improves texture fidelity while preserving the original silhouette and viewpoint. This is corroborated by Fig. 5(a): CLIP and DINO similarity rise sharply from ratio 0.25 to 0.75, and then degrade markedly at ratio 1.0. Fig. 5(b) shows that the primary gain concentrates between 6 and 9 steps, where CLIP and DINO similarity rise most steeply. Beyond 9 steps the curves quickly flatten: the semantic gains from 15 and 30 steps are marginal, while Mask IoU keeps dropping as extra editing further deviates from x^{src} . Fig. 6(b) visually confirms this saturation on facial details: at 9 steps the two female characters already exhibit sharp facial features, such as clearly resolved eyes and skin shading; increasing steps to 15 or 30 yields fewer perceptible improvements in the face regions. Therefore, we adopt $N_e = 9$, $N = 12$ in all subsequent 3D experiments for a better quality-efficiency balance.

Table 2: Ablation on key components of Reinforced Editing. We report ΔCLIP and ΔDINO , defined as the change in CLIP/DINO similarity to x^{ref} before and after editing; higher is better. Details are in Sec. 4.2.

Variant	ΔCLIP Sim. $\uparrow$	ΔDINO Sim. $\uparrow$
Full (Ours)	+0.0379	+0.0299
w/o Source Branch	-0.0523	-0.0497
w/o Noise Update	-0.0001	+0.0089

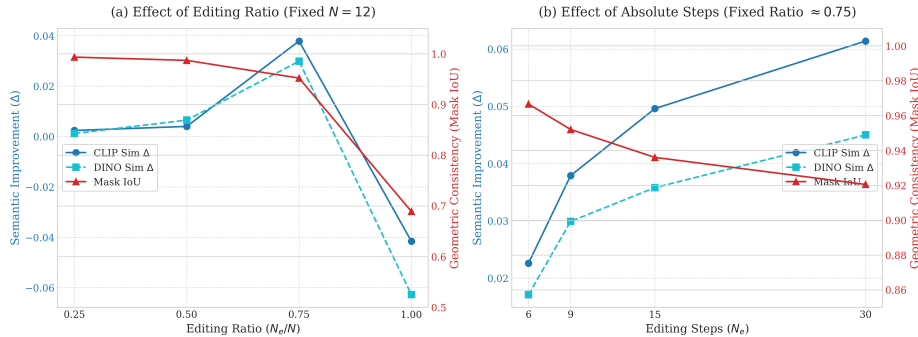


Fig. 5: Sensitivity Analysis of RE Parameters. (a) **Effect of Editing Ratio (N_e/N):** A ratio of 0.75 yields the best trade-off between semantic improvement (CLIP/DINO Δ) and structural consistency (Mask IoU). (b) **Effect of Editing Steps (N_e):** Under a fixed ratio (≈ 0.75), fewer steps (e.g., 9) achieve comparable performance to more expensive settings (e.g., 30) with significantly lower cost.

Ablation Study on RE Components. RE builds on two key components, the source branch in Eq. 7 and the dynamic noise update in Eq. 8, and we ablate each in turn to see how much it contributes. Without the source branch, only the target-conditioned velocity is used, and the trajectory drifts away from x^{src} : Fig. 4 shows the resulting color over-saturation and structural degradation, and Table 2 reports negative ΔCLIP and ΔDINO , meaning the edited output is even less faithful to x^{ref} than the unedited source. Without the noise update, the per-step ϵ is resampled at random and no longer aligned with the conditional CFG direction, diluting the edit signal in the target-unconditional delta. Table 2 shows ΔCLIP and ΔDINO dropping to nearly zero, so the edit is applied but barely takes effect. We evaluate whether these 2D-level benefits transfer to the downstream 3D generator in the next section.

4.3 Main Results

Experimental Setup. We evaluate the 3D generation quality of Trellis distilled with OREO. As baselines, we compare against the pre-trained Trellis and Photo3D, a related post-training work built on the same Trellis backbone. Following the protocol in Sec. 4.2, we render 5 views at azimuths $-90^\circ, -45^\circ, 0^\circ, 45^\circ, 90^\circ$



Fig. 6: Qualitative ablation study on RE parameters. (a) **Editing Ratio:** A low ratio (e.g., 0.25) limits editing capability, while a full ratio (1.0) destroys the original structure. Our choice of 0.75 strikes a balance. (b) **Editing Steps:** Increasing steps from 9 to 30 yields negligible visual improvement, confirming the efficiency of our setting.

for each reference and report CLIP-ViT-L/14 [27] and DINOv3-ViT-L/16 [29] embedding similarity to the reference x^{ref} , averaged over the 5 views. In addition to our proposed Conceptual Design Dataset, which targets imaginative and out-of-distribution references, we further validate on GSO [8], an in-distribution benchmark of common real-world objects, to verify that OREO does not compromise generation quality on the pre-trained model’s original distribution.

Quantitative and Qualitative Evaluation. Table 3 summarizes the quantitative comparison. OREO achieves clear gains in CLIP and DINO similarity on the Conceptual Design Dataset, our target imaginative distribution. In contrast, the improvement on the in-distribution GSO benchmark is marginal, indicating that OREO’s fidelity gain is targeted: on distributions that are not part of the training references, OREO preserves the pre-trained model’s original quality rather than compromising it. Fig. 7 shows the visual comparison. Compared with the pre-trained Trellis and Photo3D, OREO recovers finer appearance details while preserving the geometric structure and identity cues of x^{ref} . For exam-

Table 3: Main quantitative comparison of 3D generation. We report CLIP and DINO similarity to the reference x^{ref} on the Conceptual Design Dataset and GSO. Details are in Sec. 4.3.

Method	Conceptual Design		GSO	
	CLIP Sim. $\uparrow$	DINO Sim. $\uparrow$	CLIP Sim. $\uparrow$	DINO Sim. $\uparrow$
Trellis	0.7613	0.7916	0.7722	0.7022
Photo3D	0.7380	0.7837	0.7512	0.7034
OREO (Ours)	0.7834	0.8065	0.7764	0.7069

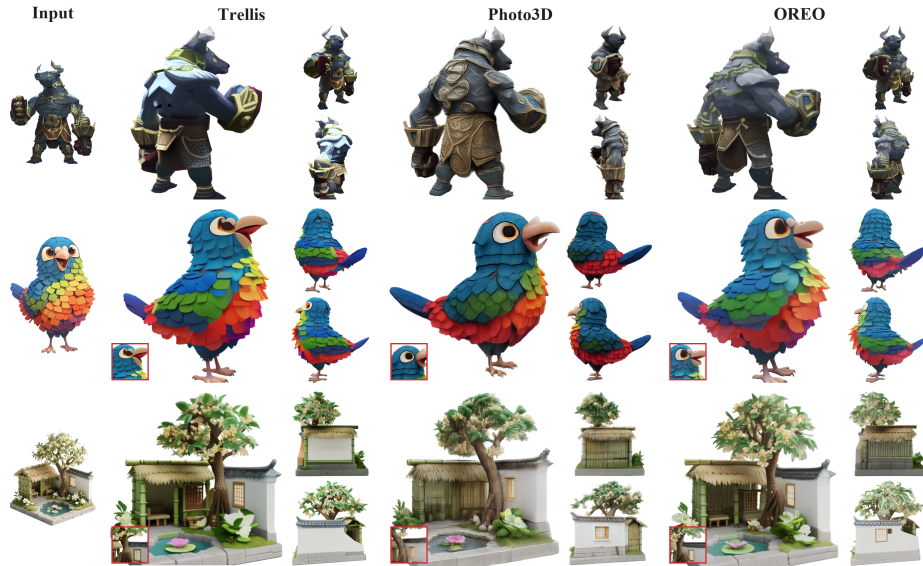


Fig. 7: Main qualitative comparison of 3D generation. OREO delivers higher multi-view visual fidelity than Trellis and Photo3D. Details are in Sec. 4.3.

ple, OREO reconstructs the eye and spotted feather patterns on the parrot, the twisted tree trunk shape in the courtyard, and more consistent back-view colors on the minotaur.

User Preference Study. To complement the automatic metrics, we further conduct a user preference study on the same held-out Conceptual Design Dataset, in which participants compare multi-view renderings from Pretrained Trellis, Photo3D, and OREO under blinded conditions. OREO obtains the highest preference ratio, indicating that human evaluators more frequently favor its balance between reference fidelity, fine-grained detail, and multi-view consistency. Full protocols and results are in the *Supp. Material*.

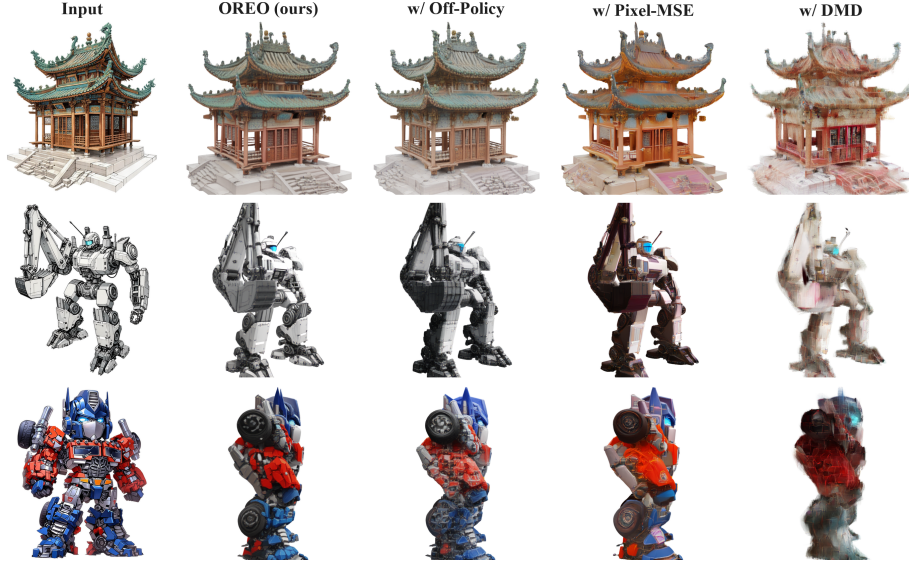


Fig. 8: Qualitative ablation of the 3D generator design choices. We compare Full OREO against other training variants. Details are in Sec. 4.4.

Table 4: Ablation on the design choices of the 3D generator. We report CLIP and DINO similarity to the reference x^{ref} on the Conceptual Design Dataset. Details are in Sec. 4.4.

Method / Variant	CLIP Sim. $\uparrow$	DINO Sim. $\uparrow$
Pretrained Trellis	0.7613	0.7916
Full OREO (Ours)	0.7834	0.8065
w/ Pixel-MSE Supervision	0.6617	0.6982
w/ Off-policy Rollout	0.7279	0.7613
w/ DMD (Score Distillation)	0.5393	0.5486

4.4 Ablation Study and Analysis

Full OREO’s training scheme rests on three designs. First, the rollout is *on-policy*: the anchor $\hat{z}_0$ is derived from the training generator’s own current output $G_\theta(x^{\text{ref}})$. Second, supervision is applied in the latent space via a contrastive objective, bypassing the differentiable renderer. Third, the contrastive target is built from an explicit multi-step pseudo-GT produced by Reinforced Editing. We ablate them with three variants: *w/ Off-policy Rollout* freezes the rollout to $G_{\theta_{\text{pre}}}(x^{\text{ref}})$; *w/ Pixel-MSE Supervision* regresses in pixel space; *w/ DMD* drops the explicit target and instead uses a single-step score-distillation gradient. Table 4 shows all three fall below the pretrained Trellis baseline while Full OREO improves over it.

Degradation under Off-policy Rollout. Freezing the rollout to $z_0 = G_{\theta_{pre}}(x^{\text{ref}})$ decouples the anchor from the training generator’s own trajectory. Training itself remains stable, and the geometric structure is largely preserved as shown in Fig. 8, but colors desaturate and fine-grained textures are progressively smoothed out. For example, the multi-color glazed roof in row 1 loses its saturation, and the mecha in row 3 fades into a muted red-blue palette. Quantitatively, CLIP and DINO similarity drop by 0.0555 and 0.0452 relative to Full OREO, and both metrics fall below the pretrained baseline.

Attenuation of Gradient through Differentiable Rendering. Replacing the latent contrastive loss with a pixel-space regression $\|x^{\text{src}} - x^{\text{tgt}}\|^2$ forces the gradient back through the frozen decoder $\mathcal{D}$ and renderer $\mathcal{P}$ before reaching v_θ . Training does not diverge, but the attenuated gradient can no longer drive fidelity improvement, and the outputs drift toward a texture-averaged, color-shifted solution. The roof in row 1 turns uniformly beige and the excavator robot in row 2 fades into washed-out white. CLIP and DINO similarity fall to 0.6617/0.6982, well below the pretrained baseline, as shown in Fig. 8.

Necessity of Reinforced Editing. Rather than construct an explicit pseudo-GT, DMD supplies the training generator with a single-step score-distillation gradient obtained from the frozen 2D editor. Without a multi-step edited target, the gradient captures only a local view of the 2D prior at one noise scale, and the accumulated update over training is high-variance and directionally inconsistent. Fig. 8 shows the resulting outputs suffer from severe geometric distortion and color collapse, e.g., the mecha in row 3 loses its structural coherence entirely. CLIP and DINO similarity collapse to 0.5393/0.5486, the worst among all variants, confirming that a coherent multi-step target from RE cannot be replaced by an implicit gradient signal.

5 Conclusion

This paper proposes **OREO**, a fidelity alignment framework for 3D generation. Addressing the lack of visual realism in existing 3D models, we introduce an on-the-fly optimization loop that leverages 2D diffusion priors. At its core, we propose Reinforced Editing to generate structure-preserving pseudo-GTs, which serve as dynamic supervision targets. Experimental results demonstrate that OREO effectively enhances the visual fidelity and texture details of 3D generators while maintaining geometric integrity, offering a scalable solution to bridge the gap between 3D geometry and 2D visual realism. Limitations are provided in the *Supp. Material*.

Future Work. Although OREO performs well, there is still room for further exploration. First, the current FlowEdit process has high inference costs; future work could explore more efficient distillation strategies (e.g., one-step editing). Second, we will attempt to extend OREO to more complex scene generation tasks, utilizing panoramic editing models to optimize large-scale 3D environments. Finally, combining with Large Multimodal Models (LMM) for more fine-grained interactive editing is also an exciting direction.

References

1. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: CVPR (2023)
2. Chen, H., Lan, Y., Chen, Y., Zhou, Y., Pan, X.: Mvdrag3d: Drag-based creative 3d editing via multi-view generation-reconstruction priors. arXiv preprint arXiv:2410.16272 (2024)
3. Chen, K., Xie, E., Chen, Z., Wang, Y., Hong, L., Li, Z., Yeung, D.Y.: Geodiffusion: Text-prompted geometric control for object detection data generation. In: ICLR. vol. 2024, pp. 8979–9001 (2024)
4. Chen, L., Wang, P., Zhang, G., Ma, Z., Zhang, L.: Omni-3dedit: Generalized versatile 3d editing in one-pass. In: CVPR. pp. 12640–12650 (2026)
5. Chen, Z., Tang, J., Dong, Y., Cao, Z., Hong, F., Lan, Y., Wang, T., Xie, H., Wu, T., Saito, S., et al.: 3dtopia-xl: Scaling high-quality 3d asset generation via primitive diffusion. In: CVPR. pp. 26576–26586 (2025)
6. Couairon, P., Aberman, K., Rabbat, M., Douze, M., Cord, M.: Flowedit: Inversion-free text-based editing using pre-trained flow models. In: ECCV (2024)
7. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: CVPR. pp. 13142–13153 (2023)
8. Downs, L., Francis, A., Koenig, N., Kinman, B., Hickman, R., Reymann, K., McHugh, T.B., Vanhoucke, V.: Google scanned objects: A high-quality dataset of 3d scanned household items. In: ICRA. pp. 2553–2560. IEEE (2022)
9. Guo, J., Gao, S., Bian, J.W., Sun, W., Zheng, H., Jia, R., Gong, M.: Hyper3d: Efficient 3d representation via hybrid triplane and octree feature for enhanced 3d shape variational auto-encoders. arXiv preprint arXiv:2503.10403 (2025)
10. Haque, A., Tancik, M., Efros, A.A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. In: ICCV. pp. 19740–19750 (2023)
11. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. In: ICLR (2023)
12. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: NeurIPS Workshop (2021)
13. Hunyuan3D, T., Yang, S., Yang, M., Feng, Y., Huang, X., Zhang, S., He, Z., Luo, D., Liu, H., Zhao, Y., et al.: Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. arXiv preprint arXiv:2506.15442 (2025)
14. Koh, E., Hyun, S., Lee, M., Chung, J., Seo, K., Heo, J.P.: Diffusion feature field for text-based 3d editing with gaussian splatting. NeurIPS **38**, 159–178 (2026)
15. Lai, Z., Zhao, Y., Liu, H., Zhao, Z., Lin, Q., Shi, H., Yang, X., Yang, M., Yang, S., Feng, Y., et al.: Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details. arXiv preprint arXiv:2506.16504 (2025)
16. Li, W., Zhang, X., Sun, Z., Qi, D., Li, H., Cheng, W., Cai, W., Wu, S., Liu, J., Wang, Z., et al.: Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. arXiv preprint arXiv:2505.07747 (2025)
17. Li, Y., Zou, Z.X., Liu, Z., Wang, D., Liang, Y., Yu, Z., Liu, X., Guo, Y.C., Liang, D., Ouyang, W., et al.: Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. arXiv preprint arXiv:2502.06608 (2025)
18. Liang, X., Ma, Z., Sun, L., Guo, Y., Zhang, L.: Aligncvc: Aligning cross-view consistency for single-image-to-3d generation. In: AAAI. vol. 40, pp. 6889–6897 (2026)

19. Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y., Lin, T.Y.: Magic3d: High-resolution text-to-3d content creation. In: CVPR. pp. 300–309 (2023)
20. Lin, C., Pan, P., Yang, B., Li, Z., Mu, Y.: Diffsplat: Repurposing image diffusion models for scalable gaussian splat generation. arXiv preprint arXiv:2501.16764 (2025)
21. Liu, X., Zhang, X., Ma, Z., Zhu, X., Lei, Z.: Mvboost: Boost 3d reconstruction with multi-view refinement. In: CVPR. pp. 21664–21673 (2025)
22. Ma, Z., Liang, X., Wu, R., Zhu, X., Lei, Z., Zhang, L.: Progressive rendering distillation: Adapting stable diffusion for instant text-to-mesh generation without 3d data. In: CVPR. pp. 11036–11050 (2025)
23. Ma, Z., Wei, Y., Zhang, Y., Zhu, X., Lei, Z., Zhang, L.: Scaledreamer: Scalable text-to-3d synthesis with asynchronous score distillation. In: ECCV. pp. 1–19. Springer (2024)
24. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. In: CVPR (2023)
25. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988 (2022)
26. Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O.R., Jagersand, M.: U2-net: Going deeper with nested u-structure for salient object detection. Pattern recognition **106**, 107404 (2020)
27. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML. pp. 8748–8763. PmLR (2021)
28. Sheynin, S., Polyak, A., Singer, U., Kirstain, Y., Zohar, A., Ashual, O., Parikh, D., Taigman, Y.: Emu edit: Precise image editing via recognition and generation tasks. arXiv preprint arXiv:2311.10089 (2023)
29. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khali-dov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., et al.: Dinov3. arXiv preprint arXiv:2508.10104 (2025)
30. Wang, J., Fang, J., Zhang, X., Xie, L., Tian, Q.: Gaussianeditor: Editing 3d gaussians delicately with text instructions. In: CVPR. pp. 20902–20911 (2024)
31. Wang, P., Chen, L., Ma, Z., Guo, Y., Zhang, G., Zhang, L.: One2scene: Geometric consistent explorable 3d scene generation from a single image. arXiv preprint arXiv:2602.19766 (2026)
32. Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J.: Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. NeurIPS **36** (2024)
33. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.m., Bai, S., Xu, X., Chen, Y., et al.: Qwen-image technical report. arXiv preprint arXiv:2508.02324 (2025)
34. Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., Yang, J.: Structured 3d latents for scalable and versatile 3d generation. In: CVPR. pp. 21469–21480 (2025)
35. Xie, C., Li, M., Li, S., Wu, Y., Yi, Q., Zhang, L.: Dnaedit: Direct noise alignment for text-guided rectified flow editing. NeurIPS **38**, 124456–124474 (2026)
36. Yang, S., Cun, X., Li, X., Li, Y., Zhang, J.: 4dvd: cascaded dense-view video diffusion model for high-quality 4d content generation. International Journal of Computer Vision **134**(5), 233 (2026)
37. Yang, S., Li, X., Cun, X., Wang, G., Li, L., Shan, Y., Zhang, J.: Gencompositor: Generative video compositing with diffusion transformer. In: The Fourteenth In-

- ternational Conference on Learning Representations (2026), <https://openreview.net/forum?id=ynim5u2N4i>
38. Yang, S., Wang, Y., Li, H., Meng, J., Wu, Y., Meng, X., Zhang, J.: Hybrid fourier score distillation for efficient one image to 3d object generation. *Visual Intelligence* **3**(1), 17 (2025)
 39. Yang, X., Shi, H., Zhang, B., Yang, F., Wang, J., Zhao, H., Liu, X., Wang, X., Lin, Q., Yu, J., et al.: Hunyuan3d-1.0: A unified framework for text-to-3d and image-to-3d generation. *arXiv preprint arXiv:2411.02293* (2024)
 40. Ye, C., Wu, Y., Lu, Z., Chang, J., Guo, X., Zhou, J., Zhao, H., Han, X.: Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. *arXiv preprint arXiv:2503.22236* **3**, 2 (2025)
 41. Yenphraphai, J., Pan, X., Liu, S., Panozzo, D., Xie, S.: Image sculpting: Precise object editing with 3d geometry control. In: *CVPR*. pp. 4241–4251 (2024)
 42. Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W.T., Park, T.: One-step diffusion with distribution matching distillation. In: *CVPR* (2024)
 43. Yu, X., Qi, X., Li, Z., Zhang, K., Zhang, R., Lin, Z., Shechtman, E., Wang, T., Nitzan, Y.: Self-evaluation unlocks any-step text-to-image generation. In: *CVPR*. pp. 7816–7826 (2026)
 44. Yushi, L., Zhou, S., Lyu, Z., Hong, F., Yang, S., Dai, B., Pan, X., Loy, C.C.: Gaussiananything: Interactive point cloud flow matching for 3d generation. In: *ICLR* (2025)
 45. Zhang, K., Mo, L., Chen, W., Sun, H., Su, Y.: Magicbrush: A manually annotated dataset for instruction-guided image editing. *NeurIPS* **36**, 31428–31449 (2023)
 46. Zhang, L., Zhang, Q., Jiang, H., Bai, Y., Yang, W., Xu, L., Yu, J.: Bang: Dividing 3d assets via generative exploded dynamics. *ACM Transactions on Graphics (TOG)* **44**(4), 1–21 (2025)
 47. Zhang, Y., Chen, N., Li, Z.: Rf-inversion: Inversion-free image editing with rectified flow. In: *CVPR* (2024)
 48. Zhao, H., Ma, X., Chen, L., Si, S., Wu, R., An, K., Yu, P., Zhang, M., Li, Q., Chang, B.: Ultraedit: Instruction-based fine-grained image editing at scale. *NeurIPS* **37**, 3058–3093 (2024)
 49. Zhao, Z., Lai, Z., Lin, Q., Zhao, Y., Liu, H., Yang, S., Feng, Y., Yang, M., Zhang, S., Yang, X., et al.: Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202* (2025)
 50. Zhuang, J., Wang, C., Lin, L., Liu, L., Li, G.: Dreameditor: Text-driven 3d scene editing with neural fields. In: *SIGGRAPH Asia 2023 conference papers*. pp. 1–10 (2023)