AnchorFlow: Training-Free 3D Editing via Latent Anchor-Aligned Flows

Zhenglin Zhou[1] Fan Ma[1] Chengzhuo Gui[1] Xiaobo Xia[2]
Hehe Fan[1] Yi Yang[1] Tat-Seng Chua[2]
[1]Zhejiang University [2]National University of Singapore
https://zhenglinzhou.github.io/AnchorFlow/

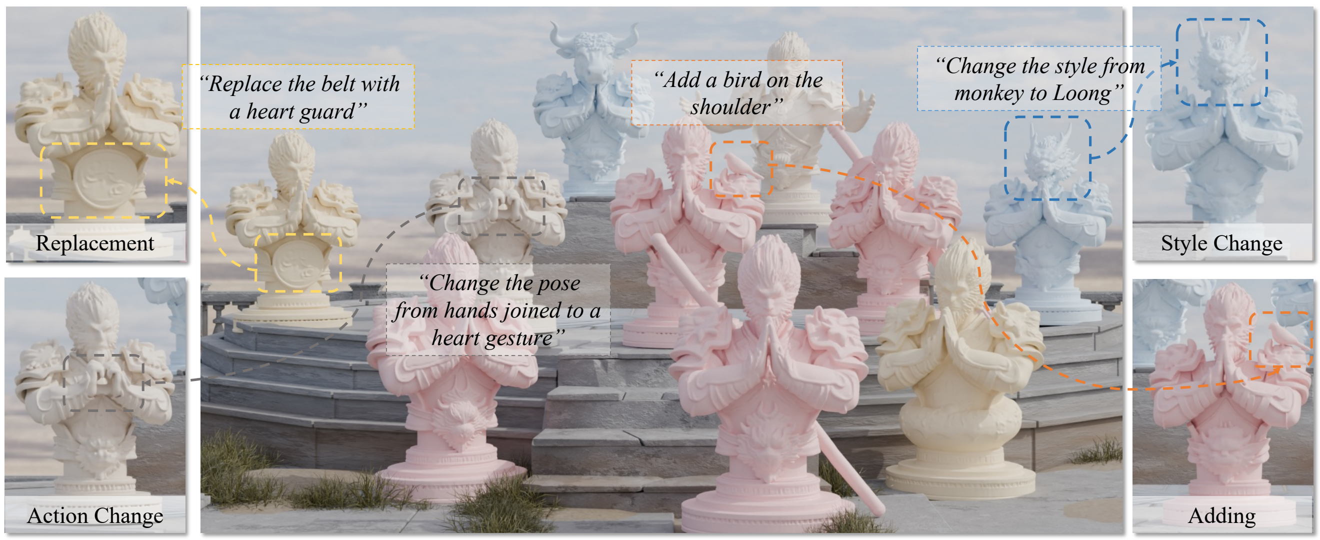

"Replace the belt with a heart guard"
"Add a bird on the shoulder"
"Change the style from monkey to Loong"
"Change the pose from hands joined to a heart gesture"
Replacement
Style Change
Action Change
Adding

Figure 1. **Overview of 3D Editing Results from AnchorFlow Across Diverse Editing Tasks.** We present four major types of edits supported by AnchorFlow: (1) **Action Change**: altering the pose or articulation of the 3D shape; (2) **Object Addition**: introducing new geometric elements; (3) **Object Replacement**: substituting existing components with new ones; (4) **Style Change**: modifying the shape style while preserving the overall.

Abstract

Training-free 3D editing aims to modify 3D shapes based on human instructions without model finetuning. It plays a crucial role in 3D content creation. However, existing approaches often struggle to produce strong or geometrically stable edits, largely due to inconsistent latent anchors introduced by timestep-dependent noise during diffusion sampling. To address these limitations, we introduce AnchorFlow, which is built upon the principle of latent anchor consistency. Specifically, AnchorFlow establishes a global latent anchor shared between the source and target trajectories, and enforces coherence using a relaxed anchor-alignment loss together with an anchor-aligned update rule. This design ensures that transformations remain stable and semantically faithful throughout the editing process. By stabilizing the latent reference space, AnchorFlow enables more pronounced semantic modifications. Moreover, AnchorFlow is mask-free. Without mask supervision, it effectively preserves geometric fidelity. Experiments on the Eval3DEdit benchmark show that AnchorFlow consistently delivers semantically aligned and structurally robust edits across diverse editing types.

1. Introduction

3D content editing is essential for applications such as games, films, and AR/VR. However, producing high-quality edits remains difficult, often requiring extensive manual work and geometric expertise. This has motivated growing interest in training-free 3D editing, where users specify semantic editing instructions and generative models apply the modification automatically. Recent progress in 3D flow-based generative models [51, 61] provides strong priors that make such editing feasible. Yet, despite these advancements, current training-free editing methods often fail to produce sufficiently strong or structurally stable edits, revealing fundamental limitations in existing formulations.

A key challenge lies in how these methods handle inversion within the editing process. Current inversion-free strategies [26] implicitly rely on stochastic Gaussian noise as latent anchors at every timestep. Because 3D flow models [51] are highly sensitive to noise perturbations, these timestep-wise anchors drift unpredictably, leading to inconsistent flow directions. As a result, semantic changes introduced during editing tend to cancel out, producing insufficient modifications or distorted geometry (see Fig. 2(a)). The reliance on unstable latent anchors restricts the reliability and controllability of training-free 3D editing.

To overcome these limitations, we propose AnchorFlow, a mask-free and training-free 3D editing framework centered on latent anchor consistency. Specifically, AnchorFlow introduces a global latent anchor shared by both source and target trajectories, and enforces its consistency through an anchor-alignment loss. In practice, we optimize a relaxed anchor-alignment loss that encourages the single-step inversions of the two trajectories to remain close in latent space. Building on this principle, AnchorFlow derives an anchor-aligned update rule that progressively transforms the source shape while preserving its structural identity, without model finetuning or mask annotation.

AnchorFlow can be justified as follows. Effective 3D editing demands stable latent references instead of timestep-dependent noise anchors. By enforcing pairwise consistency at each timestep and leveraging the continuity of the flow, our method naturally propagates these local constraints into a globally consistent latent anchor. As a result, AnchorFlow enables sufficient semantic modifications with improved geometric fidelity. To further evaluate the method, we introduce Eval3DEdit, a benchmark dataset that categorizes 3D edits into five types: addition, removal, replacement, style change, and action change. Extensive experiments on this benchmark demonstrate that AnchorFlow produces robust, structure-preserving edits and achieves performance comparable to state-of-the-art training-free and mask-free 3D editing methods [1, 5, 13, 20, 26, 51].

In summary, our contributions are as follows:

- We introduce an implicit anchor-alignment mechanism that resolves the latent anchor inconsistency problem in inversion-free editing.
- We develop AnchorFlow, a mask-free and training-free 3D editing framework that also enables scalable and low-cost curation of pairwise 3D editing data.
- We build Eval3DEdit, a benchmark dataset covering representative rigid and non-rigid edits for comprehensive 3D editing evaluation.

2. Related Work

Shape Generation. Early progress in 3D generative modeling was constrained by the lack of large-scale 3D data, leading researchers to lift 2D supervision into 3D. Building on text-to-image diffusion priors [44], methods such as DreamFields [19], DreamFusion [41], SJC [53], and their follow-ups [7, 23, 56, 60, 65, 73, 74] enabled text-guided 3D generation. With the emergence of large-scale 3D datasets [10, 11] and multi-view diffusion models [33–36, 47], 3D generation became more scalable and efficient. These advances further enabled large reconstruction models (LRMs) [18, 27, 50, 57, 62, 75] that recover 3D assets from multi-view inputs. Parallel to these developments, native 3D diffusion models have also been explored. Some encode 3D shapes into latent spaces using VAEs before diffusion training [17, 21, 58, 67, 70], while others first obtain neural 3D representations [24, 66] and then train diffusion models to generate them [21, 37, 39, 40, 48, 55, 68, 72]. Building on these advancements, large 3D foundation models (LFMs) [30, 51, 52, 59, 61, 64, 69] have recently emerged, showing substantial improvements in quality, efficiency, and robustness. Among them, vecset-based LFMs [30, 51, 52, 69] stand out for their scalability and practicality, providing strong base capabilities for downstream tasks.

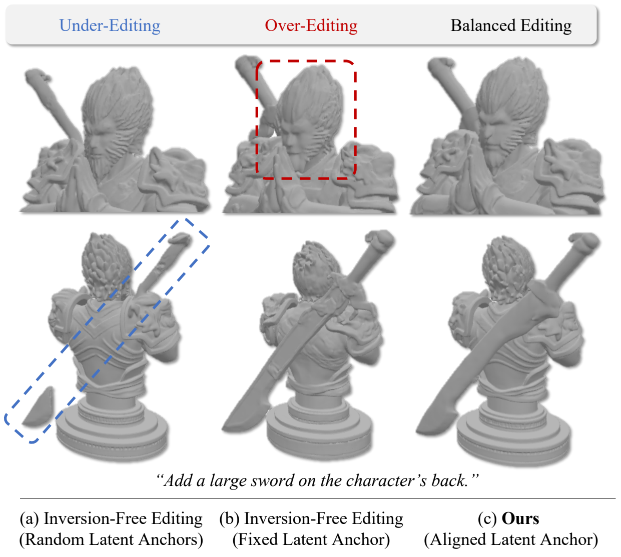

Under-Editing
Over-Editing
Balanced Editing
"Add a large sword on the character's back."
(a) Inversion-Free Editing (Random Latent Anchors)
(b) Inversion-Free Editing (Fixed Latent Anchor)
(c) Ours (Aligned Latent Anchor)

Figure 2. **Effect of Latent Anchor Selection on 3D Editing.** (a) Random timestep-wise anchors cause inconsistent flows, resulting in under-editing and geometric breakage. (b) Fixed anchors over-constrain trajectories and push the model away from the source manifold, causing over-editing. (c) Our aligned anchors maintain consistent latent references, enabling balanced editing.

Shape Editing. Editing 3D shapes has been a fundamental challenge in computer graphics and vision. In the early stage, shape editing focuses on geometry-based deformation methods [9, 45, 49], which enabled smooth local manipulations. Other classical techniques, including cut-and-paste editing [4] and sketch-based interfaces [22, 38], allowed interactive mesh fusion and shape manipulation.

While these methods provided geometric control, they required extensive manual intervention, making large-scale edits infeasible. With the advent of deep learning, the field transitioned toward 2D diffusion-driven 3D editing via score distillation sampling (SDS). These methods typically harness pretrained 2D diffusion models to guide 3D shape optimization, enabling semantic and text-driven control. Representative works [2, 6, 13, 15, 25, 31, 46, 54, 63] optimize implicit or explicit 3D representations under CLIP or SDS-based guidance. While these approaches opened a path to generative 3D editing, they suffer from unstable gradients and high computational cost.

More recently, research has moved toward multi-view diffusion and LRMs to overcome the efficiency and consistency limitations of SDS-based optimization. These methods [1, 3, 12, 14, 29, 42] generate multi-view images from edited inputs and reconstruct the edited 3D object via a LRM, achieving higher semantic fidelity and efficiency. However, these pipelines still rely on external diffusion models for view synthesis, which often leads to view-inconsistent artifacts and inaccurate geometry. Recently, LFMs [51, 61, 69] have achieved significant advances in quality and robustness, offering a new paradigm for 3D creation. Early attempts [28] rely on precise 3D masks, limiting their practicality and scalability. Our work takes a step forward by enabling mask-free and balanced 3D shape editing tailored to foundation models [51, 69].

3. Method

In this section, we formally introduce AnchorFlow, a training-free and mask-free 3D editing framework.

3.1. Preliminaries

We first review the formulation of flow matching and its training-free editing extension.

Flow Matching. Flow matching formulates generative modeling as learning a continuous-time probability flow from a simple prior distribution to a complex data distribution [32]. Let $\{\boldsymbol{X}_t\}_{t\in[0,1]}$ denote the latent trajectory, where $\boldsymbol{X}_1$ is commonly initialed from a Gaussian noise and $\boldsymbol{X}_0$ corresponds to a data sample (*e.g.*, meshes [51]). The dynamics are governed by an ODE:

$$\mathrm{d}\boldsymbol{X}_t = \boldsymbol{v}_\theta(\boldsymbol{X}_t, t, c)\mathrm{d}t, \tag{1}$$

where $\boldsymbol{v}_\theta$ is a velocity field conditioned on an optional input c. At inference, samples are obtained by integrating this ODE backward from noise $\boldsymbol{X}_1$ to data $\boldsymbol{X}_0$.

Inversion-Free Editing. Let $\boldsymbol{X}^{\mathrm{src}}$ denote a source asset, and c_{src} and c_{tar} denote source and target conditions, respectively. Editing in flow-based generative models aims to obtain an edited asset $\boldsymbol{X}^{\mathrm{tar}}$ that satisfies the semantic modification of c_{tar}, while faithfully preserving the identity of $\boldsymbol{X}^{\mathrm{src}}$. FlowEdit [26] introduces an inversion-free formulation. It constructs an editing trajectory $\{\boldsymbol{X}_t^{\mathrm{FE}}\}_{t\in[0,1]}$ that connects the source to the target:

$$\boldsymbol{X}_t^{\mathrm{FE}} = \boldsymbol{X}_t^{\mathrm{tar}} - \boldsymbol{X}_t^{\mathrm{src}} + \boldsymbol{X}_0^{\mathrm{src}}, \tag{2}$$

with boundary conditions $\boldsymbol{X}_1^{\mathrm{FE}} = \boldsymbol{X}_0^{\mathrm{src}}$ and $\boldsymbol{X}_0^{\mathrm{FE}} = \boldsymbol{X}_0^{\mathrm{tar}}$. The editing trajectory $\boldsymbol{X}_t^{\mathrm{FE}}$ evolves under the velocity difference $\boldsymbol{v}(\boldsymbol{X}_t^{\mathrm{FE}}) = \boldsymbol{v}_\theta(\boldsymbol{X}_t^{\mathrm{tar}}, t, c_{\mathrm{tar}}) - \boldsymbol{v}_\theta(\boldsymbol{X}_t^{\mathrm{src}}, t, c_{\mathrm{src}})$, where the target trajectory is defined as $\boldsymbol{X}_t^{\mathrm{tar}} = \boldsymbol{X}_t^{\mathrm{FE}} + \boldsymbol{X}_t^{\mathrm{src}} - \boldsymbol{X}_0^{\mathrm{src}}$ and and the source trajectory is obtained by linearly interpolating between the source latent and Gaussian noise:

$$\boldsymbol{X}_t^{\mathrm{src}} = (1-t)\boldsymbol{X}_0^{\mathrm{src}} + t\boldsymbol{N}_t, \quad \boldsymbol{N}_t \sim \mathcal{N}(0, I). \tag{3}$$

For simplicity, we omit auxiliary arguments in the notation of $\boldsymbol{v}(\boldsymbol{X}_t^{\mathrm{FE}})$. Then, the edited sample is updated using an Euler solver with the velocity difference, where δ_t denotes the integration step size:

$$\boldsymbol{X}_{t-\delta_t}^{\mathrm{FE}} = \boldsymbol{X}_t^{\mathrm{FE}} - \delta_t \boldsymbol{v}(\boldsymbol{X}_t^{\mathrm{FE}}). \tag{4}$$

Discussion of Inversion-free Editing. While inversion-free editing performs well in 2D image manipulation, directly extending it to 3D foundation models introduces new challenges. The naive extension often causes insufficient modification (see Fig. 2 (a)). A toy experiment helps reveal the underlying cause and motivates our method.

Specifically, we observe that fixing the Gaussian noise during inference alleviates insufficient modification, but alters the object identity (see Fig. 2 (b)). This observation suggests that the insufficient modification originates from the random Gaussian noise sampled in the denoising process. In inversion-free editing, each denoising step samples a new Gaussian noise, resetting the latent anchor at every timestep. The stochastic flow directions under timestep-wise latent anchors tend to cancel each other out, leading to a near-zero velocity. Consequently, the update trajectory remains close to the source manifold, resulting in insufficient modification. Therefore, the key challenge is to stabilize the latent anchor across timesteps, to maintain a consistent latent reference throughout the entire denoising process. This insight motivates our method, which explicitly aligns source and target trajectories through a global latent anchor, ensuring temporally coherent and geometrically stable edits (see Fig. 2 (c)).

3.2. Latent Anchor-Aligned Flows

Global Latent Anchor. To alleviate the inconsistency across timesteps, we introduce a *global latent anchor* that provides a consistent latent reference for both the source and target trajectories. We define an ideal latent anchor $\boldsymbol{A}$ as a latent point that can simultaneously reconstruct the source

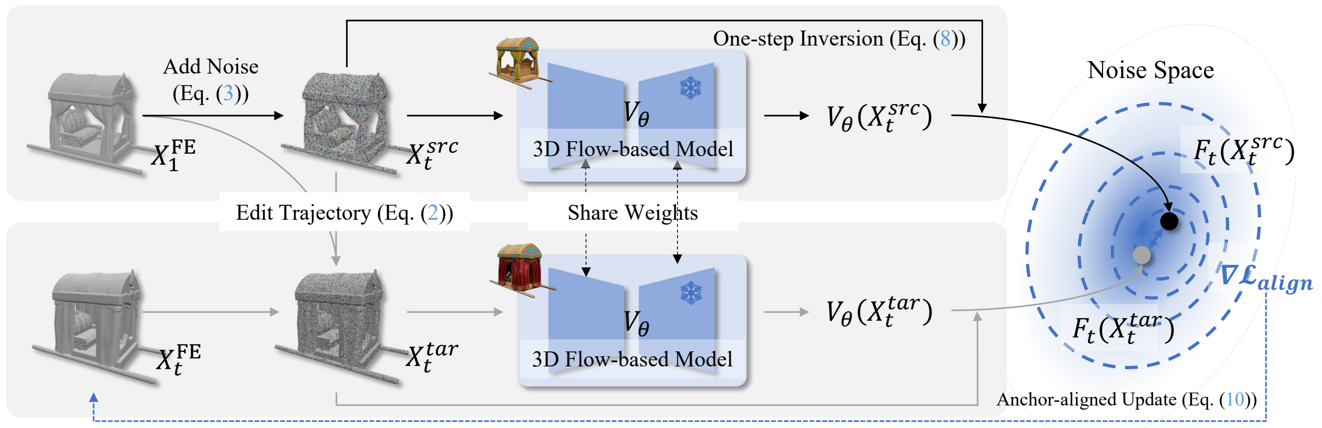


Figure 3. **Overview of the AnchorFlow for Training-free and Mask-free 3D Editing.** Given a source model and an editing instruction, AnchorFlow first constructs the source sample $\boldsymbol{X}_t^{\text{src}}$ and forms the editing sample $\boldsymbol{X}_t^{\text{FE}}$ at the t step. A 3D flow-based model $\boldsymbol{v}_\theta$ predicts velocity fields for both the source and target sample. To stabilize the editing process, AnchorFlow performs a single-step inversion to approximate the latent anchors $F_t(\boldsymbol{X}_t^{\text{src}})$ and $F_t(\boldsymbol{X}_t^{\text{tar}})$, and aligns them in noise space via the anchor-aligned update guided by $\nabla\mathcal{L}_{\text{align}}$. This design enforces consistent latent anchors, mitigates geometric distortions, and produces structurally stable 3D edits.

and target under their respective conditions:

$$\boldsymbol{A} = F_t(\boldsymbol{X}_t^{\text{src}}, t, c_{\text{src}}) = F_t(\boldsymbol{X}_t^{\text{tar}}, t, c_{\text{tar}}), \quad \forall\, t \in [0,1], \tag{5}$$

where $F_t(\boldsymbol{X}_t, t, c)$ denotes the mapping that approximates the latent in the noise space (*e.g.*, the noised latent at $t = 1$). For notational simplicity, we write $F_t(\boldsymbol{X}_t^{\text{src}})$ and $F_t(\boldsymbol{X}_t^{\text{tar}})$ to denote $F_t(\boldsymbol{X}_t^{\text{src}}, t, c_{\text{src}})$ and $F_t(\boldsymbol{X}_t^{\text{tar}}, t, c_{\text{tar}})$, respectively. This equality implies that both trajectories share a globally consistent latent anchor throughout the entire flow.

Latent Anchor-based Optimization. Directly enforcing $\boldsymbol{A}$ to be identical for all timesteps is intractable, since the mapping F_t is implicit in the diffusion dynamics. We therefore relax this hard constraint into a differentiable least-squares objective by minimizing the deviation of both reconstructions from a global anchor:

$$\min_{\boldsymbol{A}} \sum_{t\in[0,1]} \left(\|F_t(\boldsymbol{X}_t^{\text{src}}) - \boldsymbol{A}\|^2 + \|F_t(\boldsymbol{X}_t^{\text{tar}}) - \boldsymbol{A}\|^2\right). \tag{6}$$

Solving for the optimal $\boldsymbol{A}$ yields $\boldsymbol{A}^* = \frac{1}{2T}\sum_t [F_t(\boldsymbol{X}_t^{\text{src}}) + F_t(\boldsymbol{X}_t^{\text{tar}})]$. Substituting $\boldsymbol{A}^*$ back yields the strong-form latent anchor consistency objective, as detailed in Appendix A. For computational efficiency, we adopt a relaxed form, leading to the following practical latent anchor-alignment loss:

$$\mathcal{L}_{\text{align}} = \frac{1}{2} \sum_{t\in[0,1]} \|F_t(\boldsymbol{X}_t^{\text{tar}}) - F_t(\boldsymbol{X}_t^{\text{src}})\|^2. \tag{7}$$

This relaxation serves as a lower bound of the full objective, enforcing both trajectories to reconstruct toward a global latent reference across all timesteps in an implicit manner.

Single-step Approximation. To make F_t tractable, we approximate the inversion using a first-order backward step:

$$F_t(\boldsymbol{X}_t, t, c) \approx \boldsymbol{X}_t + (1-t)\boldsymbol{v}_\theta(\boldsymbol{X}_t, t, c), \tag{8}$$

which introduces negligible computational overhead while retaining differentiability. This allows us to express the alignment loss purely in terms of the velocity field.

Latent Anchor-Aligned Update. Then, we compute the gradient of $\mathcal{L}_{\text{align}}$ with respect to the current editing state $\boldsymbol{X}_t^{\text{FE}}$:

$$\begin{aligned}\nabla_{\boldsymbol{X}_t^{\text{FE}}}\mathcal{L}_{\text{align}} &= \left(\frac{\partial F_t(\boldsymbol{X}_t^{\text{tar}})}{\partial \boldsymbol{X}_t^{\text{FE}}}\right)^{\top} (F_t(\boldsymbol{X}_t^{\text{tar}}) - F_t(\boldsymbol{X}_t^{\text{src}})) \\ &= \left[\mathbf{I} + (1-t)\boldsymbol{J}_\theta\right]^{\top} (F_t(\boldsymbol{X}_t^{\text{tar}}) - F_t(\boldsymbol{X}_t^{\text{src}})),\end{aligned} \tag{9}$$

where $\boldsymbol{J}_\theta = \partial \boldsymbol{v}_\theta(\boldsymbol{X}_t^{\text{tar}}, t, c_{\text{tar}})/\partial \boldsymbol{X}_t^{\text{FE}}$. In practice, computing Jacobians for high-dimensional data is expensive. Following the Jacobian-free approximation in prior work [41], we approximate the term $\left[\mathbf{I} + (1-t)\boldsymbol{J}_\theta\right]^{\top}$ as a scalar multiple of the identity and set $\left[\mathbf{I} + (1-t)\boldsymbol{J}_\theta\right]^{\top} \approx (2-t)\,\mathbf{I}$. This yields:

$$\nabla_{\boldsymbol{X}_t^{\text{FE}}}\mathcal{L}_{\text{align}} \approx (2-t)(F_t(\boldsymbol{X}_t^{\text{tar}}) - F_t(\boldsymbol{X}_t^{\text{src}})). \tag{10}$$

Replacing the conventional velocity-difference update with the gradient-derived latent anchor-aligned direction yields:

$$\boldsymbol{X}_{t-\delta_t}^{\text{FE}} = \boldsymbol{X}_t^{\text{FE}} - \delta_t(2-t)\nabla_{\boldsymbol{X}_t^{\text{FE}}}\mathcal{L}_{\text{align}}, \tag{11}$$

which corresponds to a gradient descent step on the latent anchor alignment loss with respect to the editing state. This reformulation explicitly aligns the source and target trajectories through a shared latent anchor rather than implicitly relying on stochastic noise correspondence. Each update step, therefore, becomes geometrically consistent, progressively transforming the source into the target while preserving structural identity.

3.3. Training-Free 3D Editing

We detail the implementation of training-free 3D editing with AnchorFlow.

Condition Construction. Given a source model and an editing instruction prompt, we first construct the corresponding source condition c_{src} and target condition c_{tar}. Specifically, we render 8 image-condition views of the source model following a predefined camera distribution [61]. We then employ a large multimodal model (*e.g.*, Gemini-2.5-Flash [8]) to rank the alignment between each rendered image and the editing instruction, and select the highest-ranked view as the source condition c_{src}. Subsequently, an image editing model [8] modifies this selected view according to the instruction to generate the target condition c_{tar}. The conditions $(c_{\text{src}}, c_{\text{tar}})$ are then used to guide the editing process.

AnchorFlow Sampling. Given a flow-based 3D generative model $\boldsymbol{v}_\theta$ [51], the source shape (*e.g.*, a mesh) is encoded into its latent code $\boldsymbol{X}_0^{\text{src}}$. With $\boldsymbol{X}_0^{\text{src}}$, $(c_{\text{src}}, c_{\text{tar}})$, and $\boldsymbol{v}_\theta$, we perform sampling (Algorithm 1) to generate an editing trajectory in latent space. The editing process integrates over T time steps, where editing is from $n_{\max}$ to $n_{\min}$, and the balance between source and target guidance is controlled by s_{src} and s_{tar}. In our experiments, we use $T = 50$, $s_{\text{src}} = 3.5$, $s_{\text{tar}} = 7.5$, $n_{\min} = 1$, and $n_{\max} = 41$ as default. The resulting latent $\boldsymbol{X}_0^{\text{FE}}$ is finally decoded into 3D space, producing an edited shape that incorporates the desired modification while preserving the source identity.

Algorithm 1: AnchorFlow Sampling

```
Input: Source $\boldsymbol{X}_0^{\text{src}}$, conditions $c_{\text{src}}, c_{\text{tar}}$, flow
       model $\boldsymbol{v}_\theta$, time grid $\{t_i\}_{i=0}^T$, schedule
       $\{\sigma_{t_i}\}_{i=0}^T$
Output: Edited asset $\boldsymbol{X}_0^{\text{FE}}$
Initialize: $\boldsymbol{X}_{t_{t_i}}^{\text{FE}} \leftarrow \boldsymbol{X}_0^{\text{src}}$;
for $i = T, \dots, 1$ do
    $\delta_i \leftarrow \sigma_{t_i} - \sigma_{t_{i-1}}$
    $\boldsymbol{N}_{t_i} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\boldsymbol{X}_{t_i}^{\text{src}} \leftarrow (1-t_i)\boldsymbol{X}_0^{\text{src}} + t_i \boldsymbol{N}_{t_i}$
    $\boldsymbol{X}_{t_i}^{\text{tar}} \leftarrow \boldsymbol{X}_{t_i}^{\text{FE}} + \boldsymbol{X}_{t_i}^{\text{src}} - \boldsymbol{X}_0^{\text{src}}$
    Anchor-Aligned Update
        $F_{t_i}(\boldsymbol{X}_{t_i}^{\text{tar}}) \leftarrow \boldsymbol{X}_{t_i}^{\text{tar}} + (1-t_i)\boldsymbol{v}_\theta(\boldsymbol{X}_{t_i}^{\text{tar}}, t_i, c_{\text{tar}})$
        $F_{t_i}(\boldsymbol{X}_{t_i}^{\text{src}}) \leftarrow \boldsymbol{X}_{t_i}^{\text{src}} + (1-t_i)\boldsymbol{v}_\theta(\boldsymbol{X}_{t_i}^{\text{src}}, t_i, c_{\text{src}})$
        $\nabla\mathcal{L}_{\text{align}} \leftarrow (2-t_i)(F_{t_i}(\boldsymbol{X}_{t_i}^{\text{tar}}) - F_{t_i}(\boldsymbol{X}_{t_i}^{\text{src}}))$
    Update: $\boldsymbol{X}_{t_{i-1}}^{\text{FE}} \leftarrow \boldsymbol{X}_{t_i}^{\text{FE}} - \delta_i \nabla\mathcal{L}_{\text{align}}$;
return $\boldsymbol{X}_0^{\text{FE}}$
```

4. Experiment

4.1. Experimental Setups

Datasets. We first construct **Eval3DEdit**, a benchmark dataset designed for evaluating 3D editing performance. Eval3DEdit consists of 100 editing samples, evenly distributed across five editing categories: action change, object addition, object removal, object replacement, and style change. Each sample includes an editing instruction, a source 3D shape, a source image, and a target image. The source 3D shapes are selected from Objaverse-XL [11] using an aesthetics score threshold of 7.0, ensuring high-quality geometry. Editing instructions are generated using Gemini 2.5 Pro [8] to promote editing diversity and maintain semantic consistency with the source 3D shapes. Additional details are provided in the Appendix B.1.

Metrics. We quantitatively evaluate the editing performance using two CLIP [16, 43]-based similarity metrics. Specifically, CLIP$_{\text{img}}$ measures the similarity between the rendering from edited 3D shape and the target condition image, reflecting the identity preservation. CLIP$_{\text{txt}}$ quantifies the correspondence between the rendering from edited 3D shape and the target editing prompt, reflecting the semantic modification.

Baselines. We compare our approach against seven baseline methods specifically designed for mask-free 3D editing, comprising optimization-based, LRM-based, and LFM-based methods. Among optimization-based methods, we adopt TextDeformer [13] as the representative approach. For LRM-based methods, we include MVEdit [5] and EditP23 [1]. For the LFM-based methods, we adopt Hunyuan3D 2.1 [51] as the base model and implement three representative paradigms: (1) direct editing, which performs inference directly using the edited image as input; (2) editing-by-inversion, which first applies flow-based inversion [20] followed by editing inference; and (3) inversion-free editing, realized through FlowEdit [26]. Further implementation details are provided in the Appendix C.2.

4.2. Comparison with Prior Methods

4.2.1. Qualitative Comparisons

We first evaluate the effectiveness of our method through qualitative comparisons. As shown in Fig. 4, our method achieves consistent improvements in both geometric integrity and instruction following. Compared with the baseline method inversion-free editing [26], our method effectively mitigates the issues of insufficient editing (*e.g.*, the second row in Fig. 4) and geometric distortions in the edited

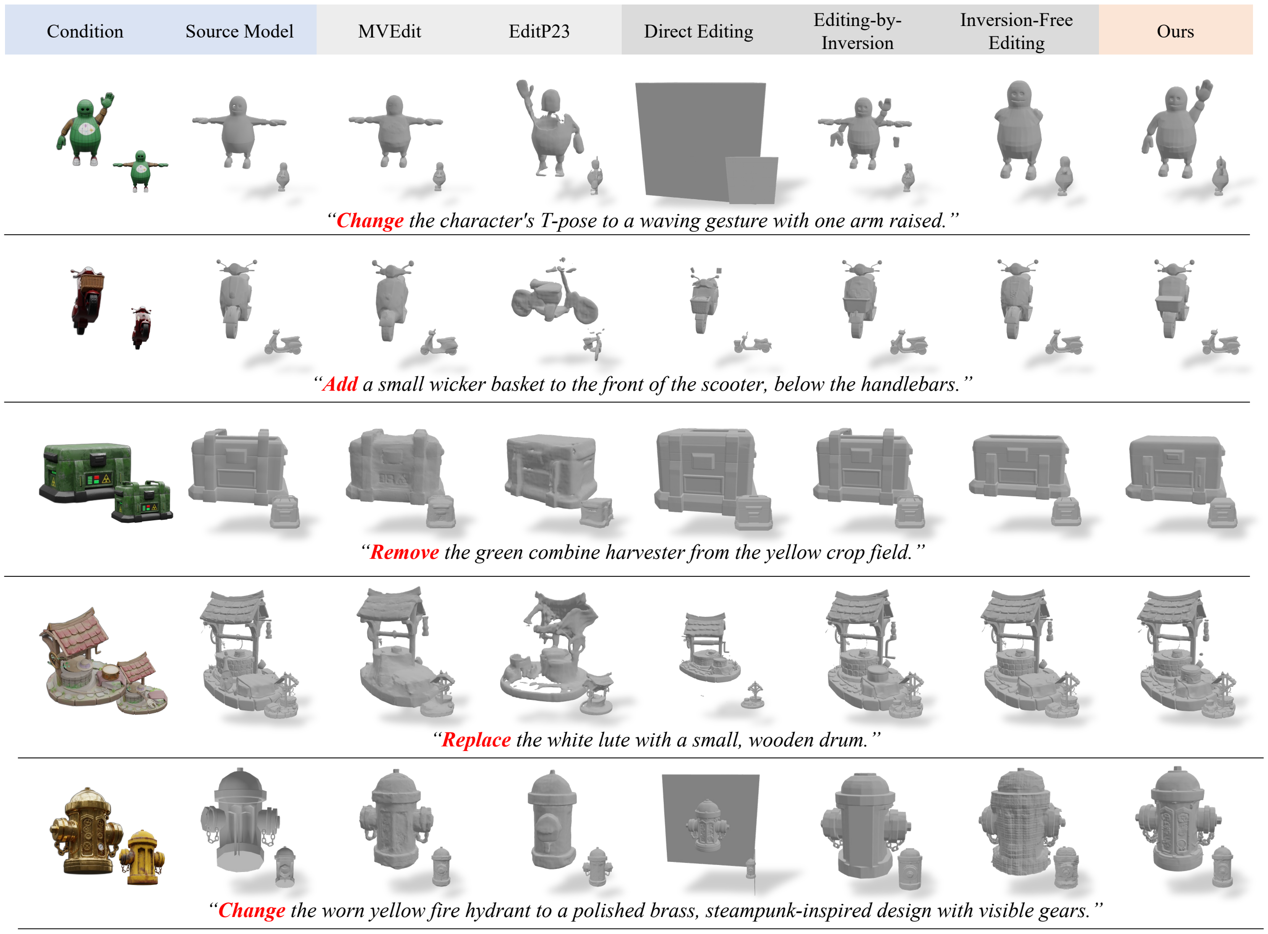

Condition
Source Model
MVEdit
EditP23
Direct Editing
Editing-by-Inversion
Inversion-Free Editing
Ours
"Change the character's T-pose to a waving gesture with one arm raised."
"Add a small wicker basket to the front of the scooter, below the handlebars."
"Remove the green combine harvester from the yellow crop field."
"Replace the white lute with a small, wooden drum."
"Change the worn yellow fire hydrant to a polished brass, steampunk-inspired design with visible gears."

Figure 4. **Qualitative Comparisons.** Each column shows condition pairs, source model, and the corresponding results from various baselines and our method. Compared with previous approaches, especially Inversion-Free Editing [26], our method produces edits that are both semantically faithful and geometrically consistent, effectively mitigating cases of insufficient edits and distorted geometry.

regions (*e.g.*, the first row in Fig. 4).

Moreover, as a mask-free editing method, we validate the effectiveness of our approach in both rigid and non-rigid editing scenarios. For rigid editing, including object addition, removal, and replacement, the goal is to enable precise local editing while preserving the unchanged regions. As shown in the second to fourth rows of Fig. 4, our method achieves high-quality local edits without explicit masks. Consequently, our approach facilitates the construction of large-scale pairwise 3D editing datasets at minimal cost, highlighting its strong potential for scalable 3D data curation. For non-rigid editing, including action and style change, the task requires global transformations while maintaining object identity. For instance, editing prompts such as "Change the character's T-pose to a waving gesture with one arm raised." (the first row in Fig. 4) require flexible yet consistent shape adaptation. Experimental results show that our method exhibits global editing capability while effectively preserving identity, highlighting the advantages of mask-free 3D editing.

4.2.2. Quantitative Comparisons

We quantitatively evaluate our method on the Eval3DEdit benchmark. As shown in Tab. 1, our method achieves the best overall performance, with $\text{CLIP}_{\text{img}} = 0.7173$ and $\text{CLIP}_{\text{txt}} = 0.4866$. Compared with the baseline, inversion-free editing [26], our method improves the average CLIP_{img} and CLIP_{txt} by 0.0067 and 0.0161, respectively, demonstrating a more favorable balance between semantic modification and identity preservation.

Compared with LRM-based 3D editing methods [1, 5], LFM-based methods show improved consistency. This indicates that performing edits through multi-view diffusion models can introduce cross-view inconsistencies, which may reduce editing quality. In contrast, LFM–based methods use end-to-end editing pipelines that operate directly in 3D latent space. Within the LFM-based 3D editing methods, we conduct three comparative experiments to examine

Table 1. **Quantitative Comparison on Eval3DEdit.** We report results using $\text{CLIP}_{\text{img}} \uparrow$ for identity preservation and $\text{CLIP}_{\text{txt}} \uparrow$ for semantic modification, where higher values indicate better performance. Metrics are evaluated across five editing categories and the Overall denotes the average performance across all categories. **Bold** indicates the best result and <u>underline</u> indicates the second-best result.

Method	Action Change		Object Addition		Object Removal		Object Replace.		Style Change		Overall	
	CLIP_{img}	CLIP_{txt}	CLIP_{img}	CLIP_{txt}	CLIP_{img}	CLIP_{txt}	CLIP_{img}	CLIP_{txt}	CLIP_{img}	CLIP_{txt}	CLIP_{img}	CLIP_{txt}
Optimization-based 3D Editing Methods												
TextDeformer [13]	0.5030	0.4389	0.5081	0.3859	0.4969	0.4268	0.5355	0.4281	0.4923	0.3926	0.5074	0.4150
LRM-based 3D Editing Methods												
MVEdit [5]	0.6015	0.4346	0.4753	0.3232	0.4784	0.3679	0.4982	0.3511	0.4836	0.3391	0.5074	0.3632
EditP23 [1]	0.5312	0.4221	0.4561	0.3313	0.4686	0.3918	0.4620	0.3549	0.4697	0.3493	0.4775	0.3699
LFM-based 3D Editing Methods												
Direct Editing [51]	0.6709	<u>0.4784</u>	0.5990	0.4287	0.5668	0.4321	0.6737	0.4732	0.5659	0.4131	0.6152	0.4451
Editing-by-Inversion [20]	<u>0.7109</u>	0.4742	**0.7081**	<u>0.4622</u>	<u>0.7282</u>	<u>0.4830</u>	<u>0.7189</u>	<u>0.4825</u>	<u>0.6933</u>	0.4666	<u>0.7119</u>	<u>0.4737</u>
Inversion-free Editing [26]	0.7096	0.4710	0.7023	0.4441	**0.7293**	**0.4884**	0.7187	0.4787	0.6929	<u>0.4701</u>	0.7106	0.4705
AnchorFlow (Ours)	**0.7313**	**0.5085**	<u>0.7050</u>	**0.4758**	0.7167	0.4735	**0.7272**	**0.4975**	**0.7061**	**0.4778**	**0.7173**	**0.4866**

the behavior of different methods. 1) Direct Editing suffers the lowest identity preservation score, indicating that directly generating from the 2D edited image causes the loss of identity information. 2) Editing-by-Inversion performs well and achieves the second-best results in several categories. Compared with this method, our method maintains comparable performance without using inversion anchors. 3) Inversion-free Editing tends to produce insufficient changes for rigid edits, while introducing geometric distortions in the edited regions for non-rigid edits. The quantitative results show that our method alleviates these issues and achieves balanced performance in both semantic modification and identity preservation. The result for object removal is moderate, which may be attributed to parameter selection. Adjusting the editing parameters (see Fig. 6) can further improve performance.

4.3. More Analyses and Justifications

The Effect of Average Directions. Inversion-free editing [26] proposes to average multiple noisy flow directions to obtain a stable update. Theoretically, it achieves more faithful edits by averaging the results of multiple random anchors, thereby reducing stochastic deviations and inconsistent directions among them. This aligns with our motivation to mitigate randomness in anchor estimation, and the two methods are thus compatible. When combined, they can yield enhanced performance. Fig. 5 shows results for inversion-free editing and our method under different numbers of averaging samples. The experiments show that both methods benefit from averaging directions. Moreover, compared with the additional computational cost introduced by averaging, our method achieves a larger performance gain with almost no extra time overhead.

The Analysis of Parameter Selection. We further analyze the influence of editing parameters. In our experiments, we

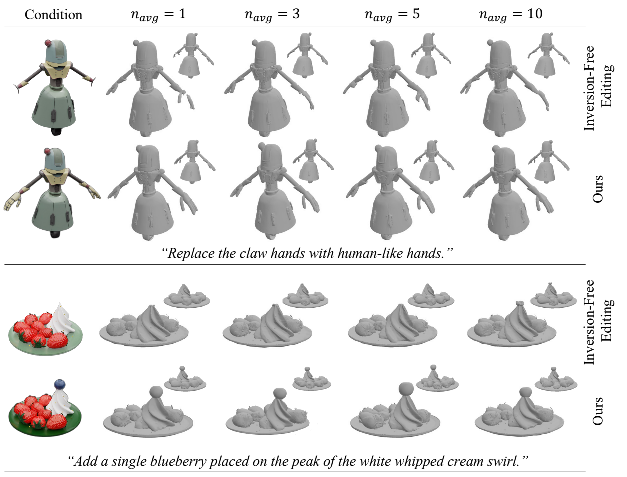

Condition
$n_{avg}=1$
$n_{avg}=3$
$n_{avg}=5$
$n_{avg}=10$
Inversion-Free Editing
Ours
"Replace the claw hands with human-like hands."
Inversion-Free Editing
Ours
"Add a single blueberry placed on the peak of the white whipped cream swirl."

Figure 5. **The Effect of Averaging Directions.** Averaging n_{avg} noisy flow directions stabilizes updates. Compared with the computational cost introduced by averaging, AnchorFlow achieves better results with almost no extra time overhead.

fix $T = 50$, $s_{\text{src}} = 3.5$, and $n_{\text{min}} = 1$, and investigate how variations in n_{max} and s_{tar} affect the editing results. Specifically, we vary $(n_{\text{max}}, s_{\text{tar}})$ among (35, 5.0), (37, 6.0), and (41, 7.5), which jointly control the strength of the editing process.. In general, Fig. 6 shows that larger values of n_{max} and s_{tar} lead to stronger semantic modification, whereas smaller values favor better identity preservation. A balanced setting of $n_{\text{max}} = 37$ and $s_{\text{tar}} = 6.0$ provides a good trade-off. Meanwhile, different editing types show distinct parameter preferences, which can be summarized into three patterns. 1) Smaller $(n_{\text{max}}, s_{\text{tar}})$. For object removal and object style change, smaller values yield simultaneous improvements in both CLIP_{img} and CLIP_{txt}. 2) Balanced configuration. For object addition and object replacement,

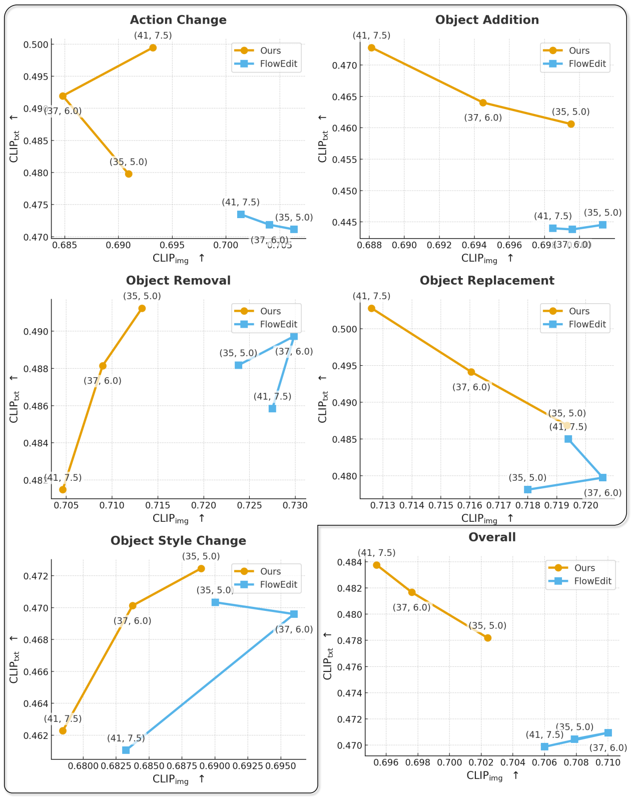


Figure 6. **Quantitative Analysis of Parameter Selection.** We investigate the effect of $(n_{\max}, s_{\text{tar}})$ across five representative editing categories. We report results using CLIP$_{\text{img}}$ ↑ for identity preservation and CLIP$_{\text{txt}}$ ↑ for semantic modification, where higher values indicate better performance.

moderate values ($n_{\max} = 37$, $s_{\text{tar}} = 6.0$) achieve balanced results, maintaining structural consistency while introducing intended semantic changes. 3) Larger $(n_{\max}, s_{\text{tar}})$. For action change, stronger edits are generally required to accomplish the desired motion transformations, as shown in Fig. 7. The results demonstrate that parameter selection has a substantial impact on editing performance, and proper adjustment can improve performance.

The Analysis of Time Cost. We further analyze the computational efficiency of our method. As shown in Tab. 2, compared with LRM-based 3D editing methods, LFM-based methods incur lower computational costs due to their one-stage 3D editing process. Moreover, our method achieves comparable runtime to inversion-free editing [26], while achieving higher editing quality.

Table 2. **Comparison of the Time Cost.** Our method matches the runtime of [26] while achieving higher editing quality.

	Text-Deformer [13]	MVEdit [5]	EditP23 [1]	Direct Editing [51]	Editing-by-Inversion [20]	Inversion-free Editing [26]	Ours
Time (s)	2229.75	513.55	50.91	21.01	34.86	25.77	26.71

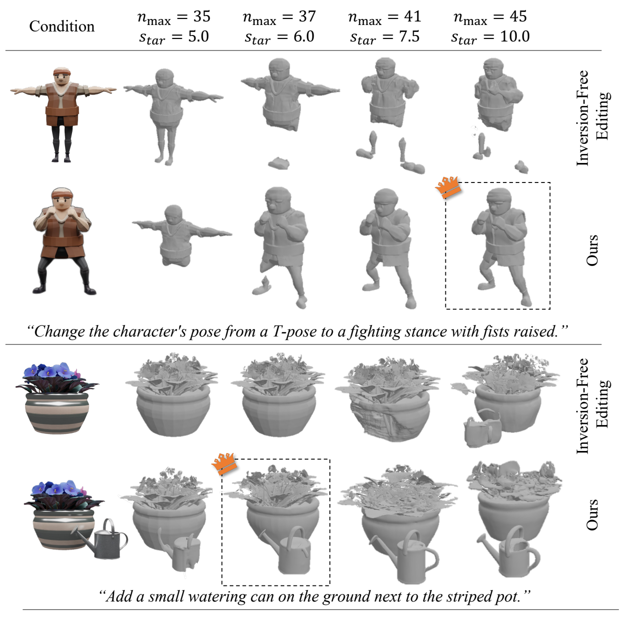

Condition
$n_{\max} = 35$ $s_{tar} = 5.0$
$n_{\max} = 37$ $s_{tar} = 6.0$
$n_{\max} = 41$ $s_{tar} = 7.5$
$n_{\max} = 45$ $s_{tar} = 10.0$
Inversion-Free Editing
Ours
"Change the character's pose from a T-pose to a fighting stance with fists raised."
"Add a small watering can on the ground next to the striped pot."

Figure 7. **Qualitative Analysis of Parameter Selection.** We visualize the effect of $(n_{\max}, s_{\text{tar}})$. Parameter selection varies across editing types, and proper adjustment can improve performance.

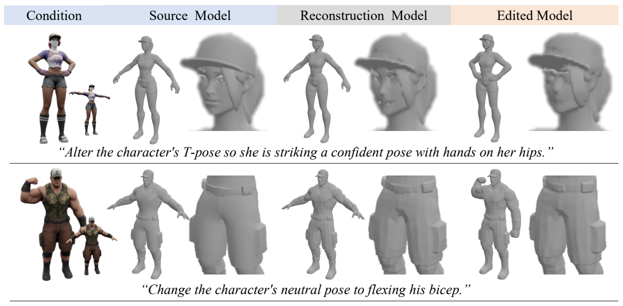

Condition
Source Model
Reconstruction Model
Edited Model
"Alter the character's T-pose so she is striking a confident pose with hands on her hips."
"Change the character's neutral pose to flexing his bicep."

Figure 8. **Limitation.** The reconstruction fidelity of the 3D VAE constrains detail preservation, and the fine features may appear degraded. Future high-fidelity 3D foundation models are expected to alleviate this limitation.

4.4. Limitation

Although our method shows good editing performance, it is limited by the reconstruction capability of the 3D VAE [51]. Compared with the source model, the reconstructed results lose high-frequency geometry (*e.g.*, facial details in the first row in Fig. 8) and structural features (*e.g.*, overalls in the second row in Fig. 8). As a result, the reconstruction inherently limits detail preservation during editing. As 3D foundation models evolve with higher-fidelity latent representations and improved reconstruction quality, our method will benefit from these advances, enabling more accurate and detail-preserving 3D editing.

5. Conclusion

In this work, we present AnchorFlow, a training-free and mask-free framework for 3D content editing that resolves the instability caused by timestep-dependent latent anchors. By enforcing global anchor consistency and updating trajectories through anchor-aligned flows, our method enables stronger semantic edits while preserving geometric structure. Evaluations on the newly constructed Eval3DEdit benchmark demonstrate that AnchorFlow achieves semantically faithful and structurally coherent 3D edits across diverse editing types, establishing a baseline for future explorations in controllable and scalable 3D content creation.

References

[1] Roi Bar-On, Dana Cohen-Bar, and Daniel Cohen-Or. Editp23: 3d editing via propagation of image prompts to multi-view. *arXiv preprint arXiv:2506.20652*, 2025. 2, 3, 5, 6, 7, 8, 16, 17, 18

[2] Amir Barda, Vladimir Kim, Noam Aigerman, Amit Haim Bermano, and Thibault Groueix. Magicclay: Sculpting meshes with generative neural fields. In *SIGGRAPH Asia*, 2024. 3

[3] Amir Barda, Matheus Gadelha, Vladimir G Kim, Noam Aigerman, Amit H Bermano, and Thibault Groueix. Instant3dit: Multiview inpainting for fast editing of 3d objects. In *CVPR*, pages 16273–16282, 2025. 3

[4] Henning Biermann, Ioana Martin, Fausto Bernardini, and Denis Zorin. Cut-and-paste editing of multiresolution surfaces. *TOG*, 2002. 2

[5] Hansheng Chen, Ruoxi Shi, Yulin Liu, Bokui Shen, Jiayuan Gu, Gordon Wetzstein, Hao Su, and Leonidas Guibas. Generic 3d diffusion adapter using controlled multi-view editing. *arXiv preprint arXiv:2403.12032*, 2024. 2, 5, 6, 7, 8, 16, 17, 18

[6] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *CVPR*, 2024. 3

[7] Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. In *CVPR*, 2024. 2

[8] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 5, 13, 14, 15

[9] Sabine Coquillart. Extended free-form deformation: A sculpturing tool for 3d geometric modeling. In *SIGGRAPH*, 1990. 2

[10] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, pages 13142–13153, 2023. 2

[11] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. In *NeurIPS*, 2024. 2, 5, 13

[12] Ziya Erkoç, Can Gümeli, Chaoyang Wang, Matthias Nießner, Angela Dai, Peter Wonka, Hsin-Ying Lee, and Peiye Zhuang. Preditor3d: Fast and precise 3d shape editing. In *CVPR*, 2025. 3

[13] William Gao, Noam Aigerman, Thibault Groueix, Vova Kim, and Rana Hanocka. Textdeformer: Geometry manipulation using text guidance. In *SIGGRAPH*, pages 1–11, 2023. 2, 3, 5, 7, 8, 16, 18

[14] William Gao, Dilin Wang, Yuchen Fan, Aljaz Bozic, Tuur Stuyck, Zhengqin Li, Zhao Dong, Rakesh Ranjan, and Nikolaos Sarafianos. 3d mesh editing using masked lrms. In *ICCV*, pages 7154–7165, 2025. 3

[15] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *ICCV*, 2023. 3

[16] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 5, 15

[17] Fangzhou Hong, Jiaxiang Tang, Ziang Cao, Min Shi, Tong Wu, Zhaoxi Chen, Shuai Yang, Tengfei Wang, Liang Pan, Dahua Lin, et al. 3dtopia: Large text-to-3d generation model with hybrid diffusion priors. *arXiv preprint arXiv:2403.02234*, 2024. 2

[18] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023. 2

[19] Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *CVPR*, 2022. 2

[20] Guanlong Jiao, Biqing Huang, Kuan-Chieh Wang, and Renjie Liao. Uniedit-flow: Unleashing inversion and editing in the era of flow models. *arXiv preprint arXiv:2504.13109*, 2025. 2, 5, 7, 8, 16, 17, 18

[21] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 2

[22] Takashi Kanai, Hiromasa Suzuki, Jun Mitani, and Fumihiko Kimura. Interactive mesh fusion based on local 3d metamorphosis. In *Graphics Interface*, 1999. 2

[23] Oren Katzir, Or Patashnik, Daniel Cohen-Or, and Dani Lischinski. Noise-free score distillation. *arXiv preprint arXiv:2310.17590*, 2023. 2

[24] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 2

[25] Hyunwoo Kim, Itai Lang, Noam Aigerman, Thibault Groueix, Vladimir G Kim, and Rana Hanocka. Meshup: Multi-target mesh deformation via blended score distillation. In *3DV*, 2025. 3

[26] Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *ICCV*, 2025. 2, 3, 5, 6, 7, 8, 16, 17, 18

[27] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. *arXiv preprint arXiv:2311.06214*, 2023. 2

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

[29] Peng Li, Suizhi Ma, Jialiang Chen, Yuan Liu, Congyi Zhang, Wei Xue, Wenhan Luo, Alla Sheffer, Wenping Wang, and Yike Guo. Cmd: Controllable multiview diffusion for 3d editing and progressive generation. In *SIGGRAPH*, 2025. 3

[30] Weiyu Li, Xuanyang Zhang, Zheng Sun, Di Qi, Hao Li, Wei Cheng, Weiwei Cai, Shihao Wu, Jiarui Liu, Zihao Wang, et al. Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. *arXiv preprint arXiv:2505.07747*, 2025. 2

[31] Yuhan Li, Yishun Dou, Yue Shi, Yu Lei, Xuanhong Chen, Yi Zhang, Peng Zhou, and Bingbing Ni. Focaldreamer: Text-driven 3d editing via focal-fusion assembly. In *AAAI*, 2024. 3

[32] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 3

[33] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. In *NeurIPS*, 2024. 2

[34] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*, pages 9298–9309, 2023.

[35] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023.

[36] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *CVPR*, pages 9970–9980, 2024. 2

[37] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *CVPR*, pages 2837–2845, 2021. 2

[38] Andrew Nealen, Olga Sorkine, Marc Alexa, and Daniel Cohen-Or. A sketch-based interface for detail-preserving mesh editing. In *SIGGRAPH*, 2005. 2

[39] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 2

[40] Evangelos Ntavelis, Aliaksandr Siarohin, Kyle Olszewski, Chaoyang Wang, Luc V Gool, and Sergey Tulyakov. Autodecoding latent 3d diffusion models. In *NeurIPS*, pages 67021–67047, 2023. 2

[41] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2, 4

[42] Zhangyang Qi, Yunhan Yang, Mengchen Zhang, Long Xing, Xiaoyang Wu, Tong Wu, Dahua Lin, Xihui Liu, Jiaqi Wang, and Hengshuang Zhao. Tailor3d: Customized 3d assets editing and generation with dual-side images. *arXiv preprint arXiv:2407.06191*, 2024. 3

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 5, 15

[44] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 2

[45] Thomas W Sederberg and Scott R Parry. Free-form deformation of solid geometric models. In *SIGGRAPH*, 1986. 2

[46] Etai Sella, Gal Fiebelman, Peter Hedman, and Hadar Averbuch-Elor. Vox-e: Text-guided voxel editing of 3d objects. In *ICCV*, pages 430–440, 2023. 3

[47] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 2

[48] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *CVPR*, pages 20875–20886, 2023. 2

[49] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. Laplacian surface editing. In *SIGGRAPH*, 2004. 2

[50] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024. 2

[51] Tencent Hunyuan3D Team. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material, 2025. 1, 2, 3, 5, 7, 8, 15, 16, 17, 18

[52] Tencent Hunyuan3D Team. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation, 2025. 2

[53] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *CVPR*, 2022. 2

[54] Junjie Wang, Jiemin Fang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. Gaussianeditor: Editing 3d gaussians delicately with text instructions. In *CVPR*, 2024. 3

[55] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. *arXiv preprint arXiv:2212.06135*, 2022. 2

[56] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023. 2

[57] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. Crm: Single image to 3d textured mesh with convolutional reconstruction model. In *ECCV*, pages 57–74. Springer, 2024. 2

[58] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. In *NeurIPS*, 2024. 2

[59] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, et al. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. In *NeurIPS*, 2025. 2

[60] Zike Wu, Pan Zhou, Xuanyu Yi, Xiaoding Yuan, and Hanwang Zhang. Consistent3d: Towards consistent high-fidelity text-to-3d generation with deterministic sampling prior. In *CVPR*, 2024. 2

[61] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *CVPR*, 2025. 1, 2, 3, 5, 14

[62] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. In *ECCV*, pages 1–20. Springer, 2024. 2

[63] Bangbang Yang, Chong Bao, Junyi Zeng, Hujun Bao, Yinda Zhang, Zhaopeng Cui, and Guofeng Zhang. Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. In *ECCV*, 2022. 3

[64] Chongjie Ye, Yushuang Wu, Ziteng Lu, Jiahao Chang, Xiaoyang Guo, Jiaqing Zhou, Hao Zhao, and Xiaoguang Han. Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. *arXiv preprint arXiv:2503.22236*, 3:2, 2025. 2

[65] Xin Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Song-Hai Zhang, and Xiaojuan Qi. Text-to-3d with classifier score distillation. *arXiv preprint arXiv:2310.19415*, 2023. 2

[66] Andrei Zanfir, Eduard Gabriel Bazavan, Mihai Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Neural descent for visual 3d human pose and shape. *arXiv preprint arXiv:2008.06910*, 2020. 2

[67] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics*, 42(4):1–16, 2023. 2

[68] Bowen Zhang, Yiji Cheng, Jiaolong Yang, Chunyu Wang, Feng Zhao, Yansong Tang, Dong Chen, and Baining Guo. Gaussiancube: Structuring gaussian splatting using optimal transport for 3d generative modeling. In *NeurIPS*, 2024. 2

[69] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics*, 43(4):1–20, 2024. 2, 3

[70] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, BIN FU, Tao Chen, Gang YU, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. In *NeurIPS*, 2023. 2

[71] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. In *ICLR*, 2024. 16

[72] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *ICCV*, pages 5826–5835, 2021. 2

[73] Zhenglin Zhou, Xiaobo Xia, Fan Ma, Hehe Fan, Yi Yang, and Tat-Seng Chua. Dreamdpo: Aligning text-to-3d generation with human preferences via direct preference optimization. In *ICML*, 2025. 2

[74] Junzhe Zhu, Peiye Zhuang, and Sanmi Koyejo. Hifa: High-fidelity text-to-3d generation with advanced diffusion guidance. *arXiv preprint arXiv:2305.18766*, 2023. 2

[75] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *CVPR*, pages 10324–10335, 2024. 2

Appendix

The Appendix provides additional technical details, derivations, implementation procedures, and experimental results that complement the main paper. It is organized as follows:

- Appendix A presents the full derivation of the latent anchor-based optimization framework introduced in the main paper, including the strong-form global objective, its closed-form solution, the reduced objective, and the relaxed per-timestep formulation adopted in practice.
- Appendix B provides additional implementation details for our framework, including dataset construction, training-free 3D editing, scalable dataset curation, and the experimental pipeline.
- Appendix C summarizes supplementary experimental settings, covering evaluation metrics, baseline implementations, and further comparisons.
- Appendix D reports additional qualitative and quantitative results that further validate the effectiveness and robustness of the proposed method.

A. Derivation of the Latent Anchor-based Optimization

In this section, we detail the derivation of the latent anchor-based optimization introduced in Sec. 3.2.

Definition of Latent Anchor-based Optimization. As defined in Sec. 3.2, we assume an ideal latent anchor $\boldsymbol{A}$. Let

$$\mathbf{s}_t \triangleq F_t(\boldsymbol{X}_t^{\mathrm{src}}, t, c_{\mathrm{src}}), \qquad \mathbf{g}_t \triangleq F_t(\boldsymbol{X}_t^{\mathrm{tar}}, t, c_{\mathrm{tar}}),$$

where $F_t(\cdot)$ maps a latent state at time t to a reference latent (*e.g.*, via a single-step inversion approximation). The strong form of the global anchor objective is

$$\mathcal{J}(\boldsymbol{A}) = \sum_{t=1}^{T} \Big(\|\mathbf{s}_t - \boldsymbol{A}\|_2^2 + \|\mathbf{g}_t - \boldsymbol{A}\|_2^2 \Big), \tag{12}$$

which enforces a single latent anchor $\boldsymbol{A}$ to explain both the source and target reconstructions across all timesteps.

Solving for the optimal latent anchor $\boldsymbol{A}^*$. The objective in (12) is quadratic and convex in $\boldsymbol{A}$. Setting the gradient to zero gives

$$\nabla_{\boldsymbol{A}} \mathcal{J}(\boldsymbol{A}) = 2\sum_{t=1}^{T} \big[(\boldsymbol{A} - \mathbf{s}_t) + (\boldsymbol{A} - \mathbf{g}_t) \big] = 4T\,\boldsymbol{A} - 2\sum_{t=1}^{T} (\mathbf{s}_t + \mathbf{g}_t) = \mathbf{0}.$$

Hence,

$$\boxed{\boldsymbol{A}^* = \frac{1}{2T} \sum_{t=1}^{T} (\mathbf{s}_t + \mathbf{g}_t)} \tag{13}$$

That is, the optimal latent anchor is the mean of all per-timestep midpoints $\mathbf{m}_t \triangleq \frac{1}{2}(\mathbf{s}_t + \mathbf{g}_t)$.

Substituting $\boldsymbol{A}^*$ Back. To obtain the reduced objective $\mathcal{J}(\boldsymbol{A}^*)$, we use the parallelogram identity:

$$\|\mathbf{x} - \mathbf{m}\|^2 + \|\mathbf{y} - \mathbf{m}\|^2 = \tfrac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2 + 2\big\|\mathbf{m} - \tfrac{\mathbf{x}+\mathbf{y}}{2}\big\|^2. \tag{14}$$

Applying (14) with $\mathbf{x}=\mathbf{s}_t$, $\mathbf{y}=\mathbf{g}_t$, and $\mathbf{m}=\boldsymbol{A}$ yields

$$\sum_{t=1}^{T} \big(\|\mathbf{s}_t - \boldsymbol{A}\|^2 + \|\mathbf{g}_t - \boldsymbol{A}\|^2 \big) = \frac{1}{2}\sum_{t=1}^{T} \|\mathbf{g}_t - \mathbf{s}_t\|^2 + 2\sum_{t=1}^{T} \Big\| \boldsymbol{A} - \frac{\mathbf{s}_t + \mathbf{g}_t}{2} \Big\|^2.$$

Minimizing the right-hand side over $\boldsymbol{A}$ gives $\boldsymbol{A}^* = \bar{\mathbf{m}} \triangleq \frac{1}{T}\sum_t \mathbf{m}_t$, and the exact reduced objective is

$$\boxed{\mathcal{J}(\boldsymbol{A}^*) = \frac{1}{2}\sum_{t=1}^{T} \|\mathbf{g}_t - \mathbf{s}_t\|^2 + 2\sum_{t=1}^{T} \|\mathbf{m}_t - \bar{\mathbf{m}}\|^2, \qquad \mathbf{m}_t = \tfrac{1}{2}(\mathbf{s}_t + \mathbf{g}_t),\ \bar{\mathbf{m}} = \tfrac{1}{T}\sum_t \mathbf{m}_t.} \tag{15}$$

The first term measures pairwise alignment between source and target reconstructions, while the second term enforces the midpoints to remain consistent across timesteps, corresponding to a single global anchor.

Relaxed Latent Anchor-based Optimization. For computational efficiency, we adopt a relaxed formulation. Since the second term in Eq. (15) is nonnegative, the objective satisfies

$$\mathcal{J}(\boldsymbol{A}^*) \geq \frac{1}{2} \sum_{t=1}^{T} \|\mathbf{g}_t - \mathbf{s}_t\|^2, \tag{16}$$

with equality if and only if $\mathbf{m}_t = \bar{\mathbf{m}}$ for all t. In practice, we adopt the simplified per-timestep form

$$\mathcal{L}_{\text{align}} = \tfrac{1}{2} \sum_{t=1}^{T} \|\mathbf{g}_t - \mathbf{s}_t\|^2 = \tfrac{1}{2} \sum_{t=1}^{T} \|F_t(\boldsymbol{X}_t^{\text{tar}}) - F_t(\boldsymbol{X}_t^{\text{src}})\|^2, \tag{17}$$

which corresponds to the natural relaxation (*i.e.*, a lower bound) of the strong-form objective (15), obtained by omitting the temporal midpoint-consistency term.

B. Additional Implementation Details

B.1. Details of Eval3DEdit

Eval3DEdit is a benchmark dataset specifically designed to evaluate 3D editing performance. It comprises 100 samples tailored for 3D editing tasks, categorized into five distinct types: action change, object addition, object removal, object replacement, and style change, with 20 samples allocated to each category. Each sample consists of an editing instruction, a source 3D shape, a source image, and a target image. The dataset is constructed through a multi-stage pipeline that collects high-quality 3D shapes, generates editing instructions, selects optimal source viewpoints, produces target images, and finally curates representative samples for comprehensive 3D editing evaluation.

- **Step 1: Source 3D Shape Collection.** We collect source models and their corresponding captions from Objaverse-XL [11]. To ensure data quality, we filtered for high-quality shapes using an aesthetics score threshold of 7.0.
- **Step 2: Editing Instruction Construction.** Given the caption associated with each 3D shape, we employed Gemini 2.5 Pro [8] to assign an appropriate editing category and to generate a corresponding editing instruction. The prompt used for this process is shown below.

Step2: Editing Instruction Construction

[Task Definition]: Generation of a Balanced Editing Instruction Dataset
[Objective]:
Given an input CSV file containing assets (*e.g.*, images or videos) and their corresponding textual captions, generate a dataset of editing instructions. This task involves assigning each asset to one of five predefined editing categories and subsequently formulating a precise instruction based on the caption and the assigned category.
[Editing Categories]:
The five (5) mandatory editing categories are:
Object Addition
Object Removal
Object Replacement
Style Change
Action Change
[Procedure and Requirements]:
Data Processing: For each asset in the input CSV, analyze its original caption.
Category Assignment: Assign one of the five predefined editing categories that is semantically appropriate based on the content described in the caption.
Instruction Generation: Based on the original caption and the assigned category, generate a specific, textual "Editing Instruction" detailing the desired modification.
Balanced Distribution: The final output dataset must maintain an approximately equal (balanced) distribution of samples across all five editing categories.
Constraint 1 (Contextual Relevance): All generated instructions (e.g., specifying additions, removals, replacements, or action modifications) must be contextually relevant and semantically coherent with the content of the original asset

caption.
Constraint 2 (Specificity): Instructions must be unambiguous and explicit. They must clearly define the specific outcome expected after the edit is applied. Avoid generalized or vague commands. The content to be edited should be specifically designated and clearly defined. For example, for a water cup, the content to be edited should not be a general 'accessory' but rather a specific 'cup handle'.
Constraint 3 (Content Preservation): The edit must not constitute a complete replacement of the original asset's core subject or scene, as this would be equivalent to new generation rather than modification. The core identity of the asset must be preserved.
Constraint 4 (Instructional Diversity): The set of generated instructions must exhibit high diversity. Repetitiveness should be minimized, particularly among instructions generated within the same editing category.
Constraint 5: Action change is only applicable to assets such as humans, animals, and others capable of producing actions. Careful selection of assets suitable for Action change is required when generating editing instructions.
[Output Format]:
The results should be delivered in a structured CSV file mapping each original asset (and its caption) to its assigned Editing Category and the corresponding generated Editing Instruction. Please generate this CSV file for me.

- **Step 3: Source Image Selection.** We rendered eight viewpoints for each source model using the rendering scripts provided by TRELLIS [61]. To select the most suitable source condition for editing, we used Gemini 2.5 Flash to identify the optimal viewpoint. The prompt used for this selection is provided below.

Step3: Source Image Selection

You are an expert visual assistant. You will be given an editing category, a specific instruction, and 8 images of the same object from different angles, labeled Image 0 through Image 7.
Your task is to determine which single image is the most suitable canvas for the requested edit. The best image should provide a clear and direct view of the object or area that needs to be modified.
- Editing Category: "{category}"
- Editing Instruction: "{instruction}"
Analyze the 8 images provided and identify the best one for this edit.
Your response MUST be a single integer number from 0 to 7, corresponding to the best image. Do not add any other text or explanation.

- **Step 4: Target Image Generation.** Next, we combined the selected source condition with the editing instruction and used Nano Banana [8] to generate the target condition. The prompt used for this generation is given below.

Step 4: Target Image Generation

As an expert image editor, your task is to edit the following image.
You must strictly and precisely follow the provided category and instruction. The edit should be realistic and seamlessly integrated into the image.
- Editing Category: "{editing_category}"
- Editing Instruction: "{editing_instruction}"

- **Step 5: Final Sample Selection.** To ensure high-quality editing outcomes, we prioritized samples where the source condition represented a frontal viewpoint. Ultimately, we selected 20 samples for each editing category to construct Eval3DEdit, a benchmark dataset covering representative rigid and non-rigid edits for comprehensive 3D editing evaluation.

B.2. Details of Training-Free 3D Editing

This section describes the implementation details of our training-free 3D editing pipeline. Given a source model and an editing instruction, we first construct the source and target conditions, and then apply the proposed AnchorFlow sampling procedure to perform 3D editing.

Condition Construction. We construct $(c_{\text{src}}, c_{\text{tar}})$ following Step 3 and Step 4 of Appendix B.1. We begin by rendering multiple candidate viewpoints of the source model and use an vision language model-based selector [8] to determine the optimal source view as c_{src}. We then pair this selected view with the editing instruction to generate the target condition c_{tar} using a text-conditioned image editor [8]. This produces a condition pair that jointly encodes the source appearance and the desired editing semantics.

AnchorFlow Sampling. We adopt Hunyuan3D 2.1 [51] as the base shape model $\boldsymbol{v}_\theta$. Hunyuan3D 2.1 is a high-fidelity 3D generative framework combining a ShapeVAE with a flow-based DiT backbone, and operates on compact vector-set latent representations suitable for 3D geometry encoding. The source 3D shape is encoded into a latent code $\boldsymbol{X}_0^{\text{src}}$. Given $\boldsymbol{X}_0^{\text{src}}$, the condition pair $(c_{\text{src}}, c_{\text{tar}})$, and the base model $\boldsymbol{v}_\theta$, we perform AnchorFlow sampling following Algorithm 1 to generate an editing trajectory in latent space. The trajectory integrates over T time steps, evolving from $n_{\max}$ to $n_{\min}$. Source and target constraints are balanced by two guidance scales, s_{src} and s_{tar}, which regulate identity preservation and editing strength, respectively. We use the default configuration $T = 50, s_{\text{src}} = 3.5, s_{\text{tar}} = 7.5, n_{\min} = 1, n_{\max} = 41$. The final latent $\boldsymbol{X}_0^{\text{FE}}$ is decoded by Hunyuan3D 2.1 into a mesh, producing an edited shape that incorporates the desired modification while preserving the source identity. On an NVIDIA H100 (96GB) GPU, each editing instance takes approximately 26.71 s.

B.3. Scalable 3D Editing Dataset Curation

Building upon the construction pipeline in Appendix B.1 and our training-free 3D editing framework in Appendix B.2, we develop a scalable and fully automated procedure for curating large-scale 3D editing datasets. Requiring no training or manual mask annotation, the pipeline efficiently produces richly annotated editing pairs.

Condition Pair Construction. Our curation pipeline begins by collecting high-quality 3D shapes and their associated captions from 3D shape datasets. Given the captions, we employ an LLM to assign an editing category and generate a corresponding editing instruction. To establish the source condition, we render multiple viewpoints for each 3D shape and use an LLM-based selector to identify the most suitable view. The selected view and the editing instruction are then fed into a text-conditioned image editor to generate the corresponding target condition, forming a semantically aligned condition pair.

Training-Free Shape Editing. We then leverage our training-free 3D editing framework to synthesize the edited 3D shape. The source model is encoded into the latent space of flow-based 3D generative model, and AnchorFlow sampling produces an edited latent trajectory that faithfully applies the specified modification while preserving shape identity. The resulting latent is decoded into a mesh, yielding the edited 3D shape paired with its source shape and editing instruction.

Dataset Curation and Applications. Our pipeline enables scalable creation of high-quality 3D editing pairs, producing data in the form of (editing instruction, source shape, edited shape). It offers an efficient approach for constructing large-scale training data for instruction-following 3D editing, thereby supporting the development of more capable and generalizable 3D editing models.

C. Supplementary Experimental Settings

C.1. Details of Measurement Metrics

In the main paper, we employ two evaluation strategies to demonstrate the superiority of the proposed method. Here we supplement the details of the measurements.

C.1.1. Evaluation with CLIP.

Based on the CLIP (EVA02-E-14-plus) model [16, 43], we introduce two CLIP-based similarity metrics CLIP_{img} and CLIP_{txt} to assess the editing performance on identity preservation and semantic modification, respectively.

- **Quantitative Evaluation of Identity Preservation.** Since the image editor [8] shows identity-preserving edits, we treat the target image as the ground truth. Based on this assumption, we quantitatively evaluate identity preservation by measuring the alignment between the edited 3D shape and the target image. Specifically, for each edited shape, we render six views at a resolution of 512×512, with a 15° elevation and a camera radius of 2.4. We then compute the CLIP image similarity between each rendered view and the target image, and use the average similarity score across all views as the identity preservation score CLIP_{img}.
- **Quantitative Evaluation of Semantic Modification.** To evaluate semantic modification, we first construct a textual description of the edited shape based on the source shape caption and the editing instruction. Specifically, we use the prompt shown below. We then compute the CLIP image-text similarity between each rendered view and the generated target caption, and use the average similarity across all views as the semantic modification score CLIP_{txt}.

Target Image Caption

You are a rewrite assistant. Given the original object captions and an editing instruction, produce ONE English sentence that describes ONLY the final, edited state:
- Describe the final state only, in the format 'subject + change'. Do NOT mention the editing process or use verbs like add/remove/replace/change.
- Be specific and objective; avoid first/second person and irrelevant speculation.
- Output exactly one English sentence. No numbering, no explanations, no quotes.
- Keep it short (<= 10 words), lower-case, no quotes, minimal punctuation.
- Examples:
instruction: 'change the robot's pose to be kneeling on one knee.' -> 'robot kneeling on one knee'
instruction: 'add a hat' -> 'figure with hat'
instruction: 'remove the sword' -> 'knight without sword'
instruction: 'replace chair with stool' -> 'scene with stool'
Editing category: {category}
Editing instruction: {instruction}

C.1.2. Evaluation with Uni3D

For further validation of identity preservation and semantic modification, we introduce two metrics based on Uni3D [71]: Uni3D_{pc} and $\text{Uni3D}_{\text{txt}}$. Unlike CLIP, Uni3D processes 3D shapes in the form of point clouds. All evaluations are conducted using the Uni3D (eva_giant_patch14_560) model.

- **Quantitative Evaluation of Identity Preservation.** We measure identity preservation using the Uni3D similarity between the source shape and the edited shape. Specifically, we extract point clouds from both shapes and assign them a fixed color (*e.g.*, gray with RGB values (100, 100, 100)). Features are then extracted using Uni3D, and identity preservation is computed as the cosine similarity between the two point-cloud feature vectors.
- **Quantitative Evaluation of Semantic Modification.** Similar to CLIP_{txt}, we evaluate semantic modification $\text{Uni3D}_{\text{txt}}$ by computing the similarity between the Uni3D point-cloud feature of the edited shape and the target caption.

C.2. Details of Baselines

In this section, we give the implementation details of the baseline methods, including TextDeformer [13], MVEdit [5], EditP23 [1], Direct Editing [51], Editing-by-Inversion [20], and Inversion-free Editing [26].

- **Implementation Details of TextDeformer.** We reproduce TextDeformer [13] following the official implementation[1]. TextDeformer deforms a source mesh into a text-specified target shape using differentiable rendering and CLIP-based semantic alignment. Instead of directly optimizing vertex displacements, it optimizes per-face Jacobians and reconstructs the deformation via a Poisson solve, enabling smooth, globally coherent geometry updates and mitigating artifacts from noisy CLIP gradients (see Fig. 4 in the original paper):contentReference[oaicite:1]index=1. The method also employs a view-consistency loss to enforce multi-view coherence and a Jacobian regularization term to preserve source shape identity. TextDeformer requires a pair of captions: a source caption describing the input mesh and an edited caption specifying the desired target geometry. To obtain these, we first construct the target caption following Appendix C.1. We then generate a concise source caption from the original asset description using Gemini 2.5 Pro with the prompt below. For each editing shape, we select the output from the `mesh_best_clip` directory as the final edited result. Since TextDeformer occasionally fails to optimize certain shapes (as noted in the original paper, where optimization may diverge or produce degenerate geometry), we exclude failure cases when computing the final averaged quantitative metrics.

Concise Source Caption

Write EXACTLY ONE short English noun phrase (<= 5 words) that summarizes the primary subject(s) described below. Use lower-case, no quotes, no punctuation.
- examples:
captions: 'a humanoid robot ...' -> 'humanoid robot'
captions: 'a low-poly house ...' -> 'small house'

[1]https://github.com/threedle/TextDeformer

captions: captions_text or ”,
Now output the phrase:

- **Implementation Details of MVEdit.** MVEdit [5] is a training-free 3D editing framework that adapts pretrained 2D diffusion models into 3D-consistent multi-view editors using a 3D Adapter that jointly denoises multi-view images and reconstructs coherent geometry. For each editing task, the input consists of a source shape and an editing instruction, and the output is an edited 3D shape. We reproduce MVEdit using the official Gradio-based implementation[2]. Specifically, we use the `Instruct 3D-to-3D` mode, which performs instruction-guided 3D shape editing. We fix the random seed to 42 for all experiments, and keep all other hyperparameters identical to the “polnareff” example provided in the official interface. The edited shape produced by the pipeline is used as the final result.
- **Implementation Details of EditP23.** EditP23 [1] is a mask-free and training-free 3D editing framework that propagates a user-provided 2D edit to a full multi-view editing. We reproduce EditP23 following the official implementation[3]. For each sample, we first render six multi-view images of the source shape using the rendering setup described in EditP23. Each editing run takes as input the source view, the edited target view, and the rendered multi-view grid. EditP23 outputs an edited multi-view grid, which we then reconstruct into a 3D mesh following the reconstruction pipeline detailed in the official implementation. Following the configuration strategy in [1], we adopt different presets for different editing types. For style-change edits, we set the target guidance scale to 5.0 and use $n_{\text{max}} = 31$. For all other editing categories, we set the target guidance scale to 21.0 and use $n_{\text{max}} = 39$. The resulting reconstructed mesh is used as the final edited shape.
- **Implementation Details of Direct Editing.** Given the target image, we perform inference using the flow-based 3D generative model [51] to obtain the edited shape. All experiments follow the official implementation[4].
- **Implementation Details of Editing-by-Inversion.** Editing-by-Inversion extends Direct Editing [51] by inserting a flow-based inversion prior to inference. We follow the conditional inversion procedure of UniEdit-Flow [20]. UniEdit-Flow introduces Uni-Inv, a predictor-corrector inversion method designed to more accurately recover the flow trajectory by mitigating the error accumulation inherent to straight, non-crossing rectified-flow paths. In our setting, we perform conditional Uni-Inv using the source image as the conditioning input during inversion. In specific, we set the target guidance scale to 1.0 and run 50 inversion steps to obtain an latent aligned with the source geometry, following the official implementation[5]. After inversion, we perform standard flow-based inference using the editing image to synthesize the final edited shapes.
- **Implementation Details of Inversion-free Editing.** We reproduce the inversion-free editing [26] on top of the flow-based 3D generative model [51]. It constructs a direct ODE between the source and target conditioned distributions, avoiding the Gaussian-noise traversal required in inversion-based editing. Concretely, it introduces a stochastic forward process that linearly interpolates between the source latent and random noise, and uses the resulting paired source-target latent to compute the velocity difference field, which is then averaged across multiple noise samples to update the editing trajectory. For fair comparison, we use the same hyperparameter settings as those in our method.

D. Additional Experiments

D.1. More Quantitative Results.

Quantitative Evaluation with Uni3D-based Metrics. Beyond the CLIP-based evaluation in main paper, we further validate these observations using Uni3D-based identity and semantic metrics (*cf*. Appendix C.1.2), as reported in Tab. 3. The results show a consistent trend in Tab. 1. LRM-based methods exhibit the weakest performance across both identity preservation and semantic modification due to multi-view inconsistency. LFM-based approaches achieve substantially higher scores, confirming the benefit of operating directly in a 3D latent space. Within this category, editing-by-inversion and inversion-free editing obtain strong identity preservation but either rely on expensive inversion or tend to produce under-edited or distorted geometry. In contrast, our AnchorFlow method achieves competitive identity scores while delivering the strongest semantic alignment among all LFM methods, showing that our approach provides a more balanced and reliable editing results. These Uni3D results corroborate the CLIP-based findings and further demonstrate the effectiveness and robustness of our training-free 3D editing pipeline.

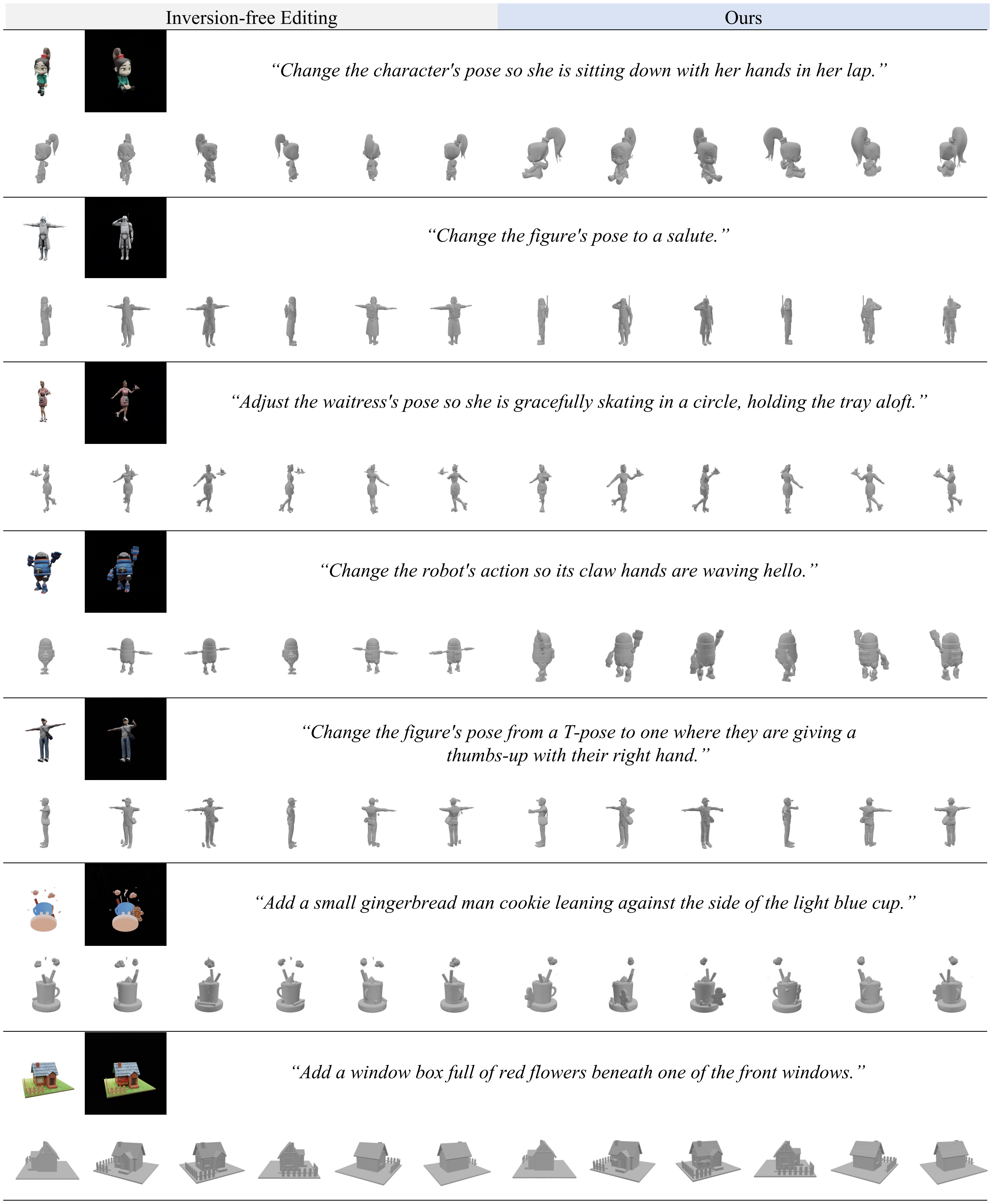
[2] https://github.com/Lakonik/MVEdit
[3] https://github.com/editp23/editp23
[4] https://github.com/Tencent-Hunyuan/Hunyuan3D-2.1
[5] https://github.com/DSL-Lab/UniEdit-Flow

Table 3. **Quantitative Comparison across Different 3D Editing Methods on the Eval3DEdit benchmark.** We report results using $\text{Uni3D}_{\text{pc}} \uparrow$ for identity preservation and $\text{Uni3D}_{\text{txt}} \uparrow$ for semantic modification, where higher values indicate better performance. Metrics are evaluated across five representative editing categories, including action change, object addition, object removal, object replacement, and style change. The Overall denotes the average performance across all categories.

Method	Action Change		Object Addition		Object Removal		Object Replace.		Style Change		Overall	
	Uni3D_{pc}	$\text{Uni3D}_{\text{txt}}$	Uni3D_{pc}	$\text{Uni3D}_{\text{txt}}$	Uni3D_{pc}	$\text{Uni3D}_{\text{txt}}$	Uni3D_{pc}	$\text{Uni3D}_{\text{txt}}$	Uni3D_{pc}	$\text{Uni3D}_{\text{txt}}$	Uni3D_{pc}	$\text{Uni3D}_{\text{txt}}$
Optimization-based 3D Editing Methods												
TextDeformer [13]	0.1321	0.1393	0.2170	0.1293	0.1549	0.1249	0.2579	0.1269	0.1444	0.1288	0.1818	0.1298
LRM-based 3D Editing Methods												
MVEdit [5]	0.4891	0.1204	0.1366	0.0653	0.2224	0.0571	0.2741	0.0859	0.2162	0.0685	0.2677	0.0794
EditP23 [1]	0.1272	0.1527	0.0879	0.0667	0.0733	0.0774	0.1240	0.0862	0.0887	0.0743	0.1002	0.0915
LFM-based 3D Editing Methods												
Direct Editing [51]	0.2611	0.2227	0.2982	0.1878	0.2658	0.1184	0.4006	0.2250	0.2485	0.1274	0.2948	0.1762
Editing-by-Inversion [20]	0.4512	0.2432	0.4990	0.2445	0.5094	0.2320	0.5807	0.2767	0.5317	0.2609	0.5144	0.2515
Inversion-free Editing [26]	0.4701	0.2375	0.5242	0.2377	0.5447	0.2436	0.6052	0.2742	0.5407	0.2566	0.5370	0.2499
AnchorFlow (Ours)	0.3123	0.2756	0.4438	0.2504	0.4501	0.2314	0.5220	0.2788	0.4577	0.2494	0.4372	0.2571

D.2. More Qualitative Results.

Beyond the qualitative comparisons provided in the main paper, we include additional examples in Fig. 9 and Fig. 10 to further assess the effectiveness of our method. Note that in all comparative experiments, AnchorFlow and the baseline [26] share same hyperparameter settings to ensure a fair comparison. The results demonstrate that our method produces edits that are both semantically faithful to the instruction and geometrically coherent. Across non-rigid editing tasks such as action change (*e.g.*, rows 1–5 in Fig. 9), AnchorFlow generates smooth and globally consistent articulations. In contrast, inversion-free editing often exhibits insufficient deformation (*e.g.*, rows 2 and 4 in Fig. 9) or noticeable distortions in limb shape or body proportions (*e.g.*, rows 1 and 5 in Fig. 9). These additional examples confirm that our method maintains stable geometry even under large pose changes while accurately following fine-grained instructions such as "saluting" and "waving." For rigid editing tasks, such as object addition (*e.g.*, rows 1–3 in Fig. 10), AnchorFlow performs precise and localized modifications without affecting unrelated regions. This further supports that AnchorFlow enables mask-free editing. Overall, the extended qualitative results provide additional evidence that AnchorFlow achieves strong semantic modification while preserving identity across both rigid and non-rigid editing tasks.



Inversion-free Editing
Ours
"Change the character's pose so she is sitting down with her hands in her lap."
"Change the figure's pose to a salute."
"Adjust the waitress's pose so she is gracefully skating in a circle, holding the tray aloft."
"Change the robot's action so its claw hands are waving hello."
"Change the figure's pose from a T-pose to one where they are giving a thumbs-up with their right hand."
"Add a small gingerbread man cookie leaning against the side of the light blue cup."
"Add a window box full of red flowers beneath one of the front windows."

Figure 9. More qualitative results using AnchorFlow.

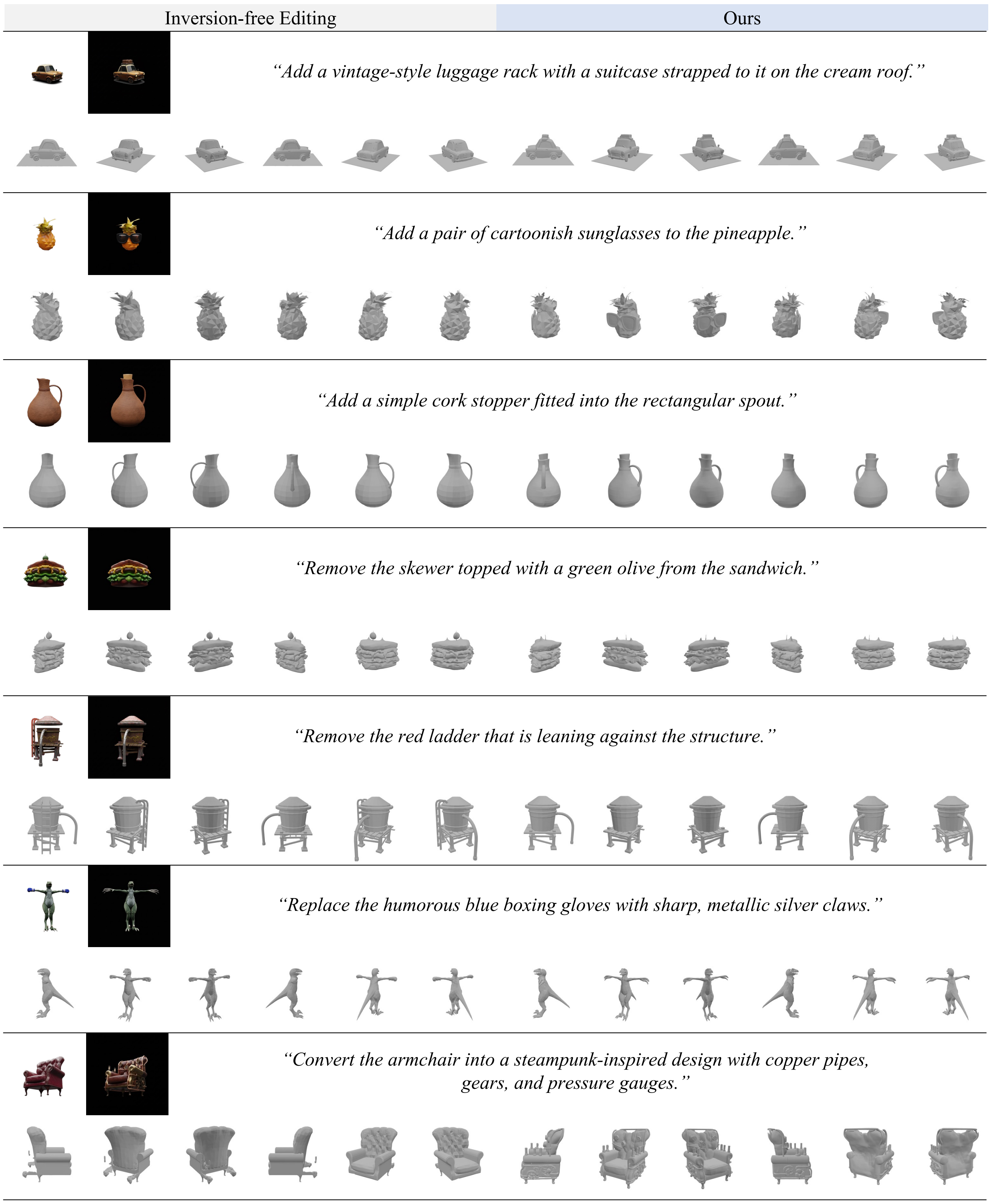

Inversion-free Editing
Ours
"Add a vintage-style luggage rack with a suitcase strapped to it on the cream roof."
"Add a pair of cartoonish sunglasses to the pineapple."
"Add a simple cork stopper fitted into the rectangular spout."
"Remove the skewer topped with a green olive from the sandwich."
"Remove the red ladder that is leaning against the structure."
"Replace the humorous blue boxing gloves with sharp, metallic silver claws."
"Convert the armchair into a steampunk-inspired design with copper pipes, gears, and pressure gauges."

Figure 10. More qualitative results using AnchorFlow.