TanGO: Training-Free 3D Editing via Tangent-space Guidance and Optimization

Siwoo Lim[1], Sunjae Yoon[2], Gwanhyeong Koo[1], Hyeonseo Yun[1], and Chang D. Yoo[1*]

[1] Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea
{siu4450, kookie, hyunseo, cd_yoo}@kaist.ac.kr
[2] Chung-Ang University, Seoul, Republic of Korea
{sunjaeyoon}@cau.ac.kr

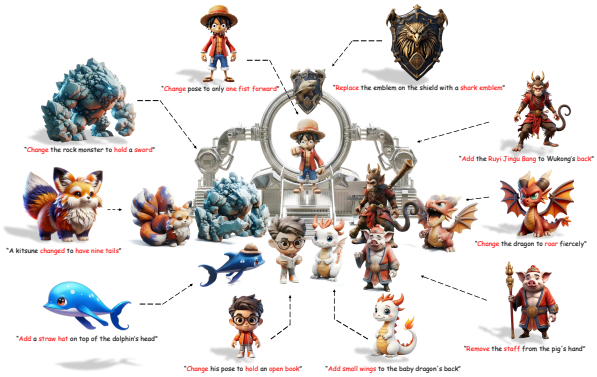

Fig. 1: Overview of 3D Editing Results. TanGO achieves precise localized edits across diverse categories, preserving unedited geometry and source identity.

Abstract. While recent flow-matching 3D generative models (e.g., VecSet) adopt structured representations, their tokens share global context, causing conventional training-free editing to suffer from semantic artifacts such as collapsed preserved regions or incomplete transformations. To address this, we propose **TanGO**, a training-free framework that enables adaptive per-token steering in the tangent space of generative dynamics. To realize this selective control, we formulate a one-step optimal control rule and determine the strength of each token's control signal using a von Mises-Fisher inspired directional discrepancy derived from the source and target velocity fields. Experiments show that TanGO substantially reduces structural artifacts and achieves state-of-the-art performance, outperforming existing 3D editing baselines. The code is publicly available at `https://github.com/siw00-lim/TanGO`.

Keywords: 3D Editing · Training-free · Optimal Control

* Corresponding Author

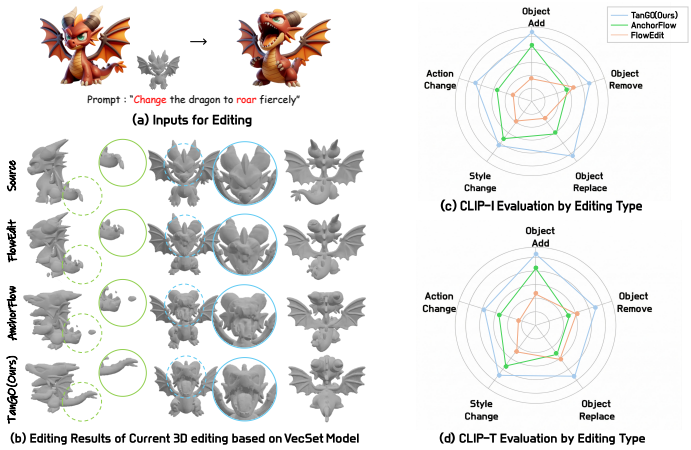

Fig. 2: (a) Input for text-driven 3D editing. (b) Prior methods often introduce semantic artifacts, such as incomplete edits in target regions or structural degradation in preserved areas. In contrast, TanGO preserves unedited regions while achieving precise, localized edits. (c-d) CLIP-I and CLIP-T scores across five editing categories show that TanGO consistently improves the balance between localized preservation and semantic modification over existing baselines.

1 Introduction

Recently, 3D editing technology has become an essential tool in digital asset creation, with its impact stretching from gaming and entertainment to architecture and industrial design [2, 7, 14]. As digital twins and virtual environments become more sophisticated, there is a growing demand for tools that can precisely modify specific parts of a 3D model without having to rebuild the entire asset from scratch. Alongside this trend, the success of 3D generative models [8, 15, 23, 24, 26, 30] has greatly accelerated the field by providing a strong foundation for high-quality editing. This has allowed the technology to move beyond simple shape changes toward smart, semantic-aware modifications that understand the underlying structure of 3D content [10, 13, 25].

Even with these technical advancements, 3D editing still faces practical challenges. Most existing methods rely on low-level representations like voxels and fail to utilize the structured space of advanced high-level representations such as VecSet [9, 17, 20, 22, 34, 35]. The main issue is that applying traditional editing methods to these high-level formats often ignores their natural semantic boundaries. As shown in Fig. 2(b), results edited by existing methods such as the 2D editing framework FlowEdit [19] or the 3D-based editing method AnchorFlow [36] suffer from significant semantic artifacts, including mesh tearing or surface collapsing in edited structures. Furthermore, these methods often fail to preserve the original source identity, even in structures that should be preserved.

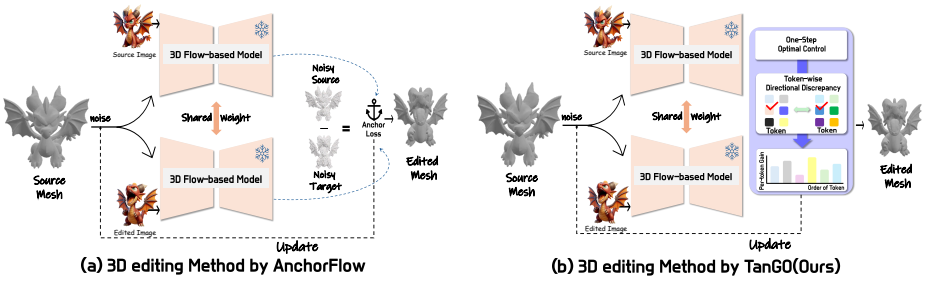

Fig. 3: Comparison of 3D editing frameworks. (a) AnchorFlow optimizes an anchor-alignment loss to update the edited mesh. (b) TanGO (Ours) formulates the editing process as a one-step optimal control problem to modulate update signals at the token level. Our formulation selectively amplifies signals for tokens in regions designated for editing while suppressing those in regions to be preserved. To achieve this, we compute the token-wise directional discrepancy between the source and target velocities, and use it to derive an adaptive gain that scales the update intensity according to local editing requirements.

To demonstrate that these issues are not limited to specific samples, we conducted a quantitative evaluation using various types of editing datasets as illustrated in Fig. 2(c,d). We introduced CLIP-I [27] and CLIP-T [27] scores to measure semantic alignment. The results show that existing methods fail to achieve consistently high performance across all data types, and the scores themselves remain significantly low. This quantitative result confirms the persistent mismatch between the text prompts and the final 3D outputs in prior methods.

For a deeper understanding of why these artifacts occur when applying prior editing methods to high-level representations, we analyze the Hunyuan3D 2.1 model [17]. Our findings reveal that while tokens interact globally, each token's geometric influence is highly localized. This indicates that the imprecision of prior methods stems from the absence of explicit per-token actuation. Therefore, successful editing requires selective control. Instead of relying on a uniform global anchor loss, we propose **TanGO**, a training-free framework for localized 3D editing that allocates adaptive per-token steering in the tangent space (Fig. 3).

Specifically, TanGO formulates editing as an instantaneous one-step optimal control problem, yielding a simple closed-form update that balances editing progress and control energy. To achieve this, we define a vMF-grounded token-wise directional discrepancy between the normalized source and target velocities, which determines each token's steering strength. This enables adaptive per-token steering via modulated gains and mean-gain normalization.

In summary, TanGO enables mask-free localized 3D editing by explicitly allocating control at the token level in tangent space. Additionally, to address the lack of structural diversity in existing benchmarks, we introduce the TanGOEdit dataset, which consists of both rigid and non-rigid editing tasks.

2 Related Works

2.1 3D mesh generation

Modern 3D generation has rapidly progressed from SDS-based lifting of 2D priors [8, 24, 26, 29] to native 3D foundation models based on diffusion and flow matching. Representative examples include TRELLIS [31], which introduces structured 3D latents (SLAT) for scalable and versatile generation, as well as recent large-scale systems such as Hunyuan3D [17], TripoSG [22], and VecSet-based generators [34] that improve quality, efficiency, and condition alignment. These models provide the representational backbone for downstream 3D editing.

2.2 3D mesh editing

3D editing is more challenging than 2D editing [4, 5, 18, 33] because it must preserve geometric coherence and consistency over the whole shape. Instant3dit [3] formulates 3D editing as multiview image inpainting followed by reconstruction, while more recent methods directly edit native 3D latent spaces. VoxHammer [21] improves local editing by reusing inverted latents and cached features for preserved regions, Nano3D [32] extends FlowEdit [19] into TRELLIS [31] and introduces connectivity-aware region merging, and AnchorFlow [36] stabilizes inversion-free editing through latent-anchor consistency under timestep-dependent noise. Compared with these methods, our approach focuses on explicit per-token actuation in the tangent space, directly exploiting the spatial locality of VecSet tokens for localized editing.

3 Preliminaries

3.1 VecSet-based 3D Generative Model

We consider a pretrained flow-based 3D generative model defined over a VecSet latent representation. Given a condition c (e.g., a text prompt or a reference image), the model operates on a latent state

$$X_t = \{x_t^1, x_t^2, \ldots, x_t^N\}, \qquad x_t^i \in \mathbb{R}^d, \tag{1}$$

where each x_t^i is a latent token and N is the number of tokens. In our setting, the pretrained Hunyuan3D 2.1 model uses $N = 4096$ tokens. The model predicts a conditional velocity field $v_\theta(X_t, c, t)$, which defines the generative dynamics in latent space. Starting from a noisy latent state, the model follows this velocity field to produce a clean latent representation, which is then decoded into a 3D mesh. Although VecSet represents a 3D object as a set of latent tokens, it has often been regarded as a relatively global representation because the tokens interact through self-attention and share global contextual information. In this view, perturbing one token may influence others through global token interactions, making VecSet-based models appear less suitable for localized editing than voxel-based representations.

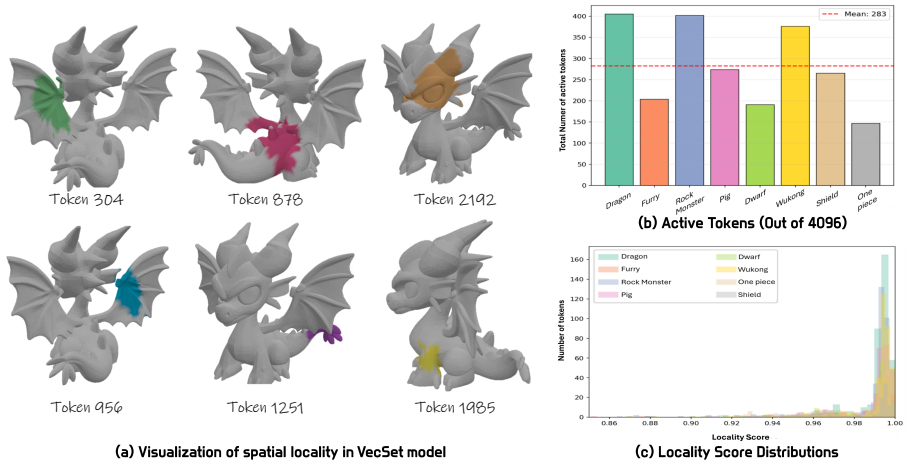

Fig. 4: (a) visualizes the localized influence of individual tokens in the VecSet model, where the colored areas indicate specific regions activated by each respective token. (b) illustrates the number of active tokens out of a total of 4096 triggered during the inference of each data category shown on the x-axis. (c) presents the locality score distribution for datasets across various categories, quantifying how locally the VecSet model responds. The y-axis represents the number of activated tokens corresponding to each locality score.

3.2 Training-free Editing in Flow-based Models

In training-free editing, we are given a source 3D object and a target condition describing the desired edit, and the goal is to obtain an edited 3D mesh without finetuning the pretrained generator. Let c_{src} and c_{tar} denote the source and target conditions. At the current editing state X_t^{FE}, the model predicts two conditional velocity fields,

$$v_{src} = v_\theta(X_t^{FE}, c_{src}, t), \qquad v_{tar} = v_\theta(X_t^{FE}, c_{tar}, t). \tag{2}$$

Most existing methods derive an editing signal from their difference, $\Delta v = v_{tar} - v_{src}$, and use it to update the editing trajectory. Although Δv is already defined token-wise because both v_{src} and v_{tar} inherit the tokenized VecSet structure, prior methods do not explicitly actuate it at the token level. Instead, they typically apply a globally shared scaling factor to the full velocity difference, so preserved and editable regions are often perturbed together.

3.3 Observation: Token-wise Spatial Locality in VecSet

Despite reports that transferring trajectory-based editing to VecSet-based 3D generators can be unstable potentially due to their more global token inter-

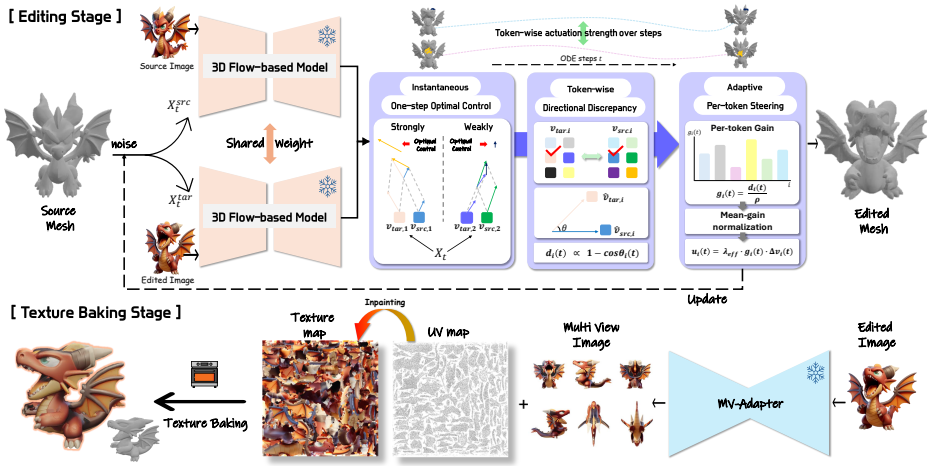

Fig. 5: Overview of TanGO. Editing stage: At each ODE step, we compute token-wise source/target velocities and their difference $\Delta v_i(t)$. To selectively amplify update signals in editable regions while suppressing those in regions to be preserved, we formulate an instantaneous one-step optimal control that yields a per-token steering gain and applies $u_i(t)$. The demand $d_i(t)$ is defined by the token-wise directional discrepancy between normalized velocities, and mean-gain normalization keeps the overall guidance energy consistent across steps. Texture baking stage: We render multi-view images via an MV-Adapter, construct UV/texture maps, inpaint missing texels, and bake the texture onto the edited mesh.

actions [32], our empirical analysis shows that tokens in the VecSet-based Hunyuan3D 2.1 model often exhibit strong spatial locality in practice[3]. As illustrated in Fig. 4(a), visualizing individual tokens reveals that their geometric influence is typically concentrated on a specific part (e.g., wing, tail, arm, or facial region) rather than being distributed across the entire object. This suggests that each token tends to predominantly affect a limited geometric component, while other regions are shaped by different tokens.

This locality is also supported quantitatively. Across multiple object categories, we find that only a subset of the 4096 latent tokens becomes active for a given object. As shown in Fig. 4(b), the number of active tokens remains well below the full token budget, indicating that generation in VecSet space is sparse rather than uniformly dense.

To quantify locality more directly, we compute a locality score to measure the spatial concentration of each token's geometric influence. As illustrated in Fig. 4(c), the majority of tokens achieve high locality scores (e.g., above 0.96)

[3] We also demonstrate in the Appendix that this strong token locality holds true across other generative models trained with the VecSet representation.

across datasets. Therefore, while self-attention enables global token interactions, the geometric effect of individual tokens remains highly localized in our model.

This observation has an important implication for training-free 3D editing: if different tokens correspond to different local geometric regions, then editing should not apply a uniform control signal to all tokens. Instead, control should be selectively strengthened for tokens in modified areas while being suppressed for tokens in preserved regions. Motivated by this, we propose **TanGO**, which derives token-wise control signals directly from the source and target velocity fields to perform localized, geometry-aware steering without explicit spatial masks.

4 Method

Existing training-free 3D editing methods steer the latent trajectory using the source–target velocity difference $\Delta v = v_{\text{tar}} - v_{\text{src}}$, but typically apply a *token-shared* scaling schedule. As a result, editable and preserved regions are often perturbed together. Motivated by the token-wise spatial locality of VecSet latents (sec. 3.3), we cast editing as *tangent-space control* and decouple two coupled questions at every ODE step: **(i) where to actuate** (localization across tokens) and **(ii) how strongly to actuate** (trajectory-level guidance). TanGO addresses this by casting editing as a per-token *allocation* problem: a closed-form minimal-intervention control law is fully determined once we specify a principled token-wise demand $d_i(t)$, while a trajectory-level normalization stabilizes the overall guidance energy (Fig. 5). Additionally, to overcome the absence of a native texture pipeline in VecSet models, we integrate a process that generates and bakes texture maps using multi-view diffusion [16] and inpainting models [28].

4.1 Instantaneous One-step Optimal Control

We perturb the source dynamics by a token-wise control input $u_i(t)$, yielding

$$\dot{x}_t^i = v_{\text{src},i}(t) + u_i(t). \tag{3}$$

At each ODE step, we choose $u_i(t)$ to maximize first-order progress toward the target tangent direction $\Delta v_i(t)$, while penalizing excessive intervention with a quadratic control-energy regularizer:

$$\max_{u_i} \; d_i(t)\,\langle u_i, \Delta v_i(t)\rangle - \frac{\rho}{2}\|u_i\|^2 \quad \text{s.t.} \quad u_i \in \text{span}(\Delta v_i(t)), \tag{4}$$

where $d_i(t) \geq 0$ is the token-wise demand and $\rho > 0$ penalizes control energy. The span constraint enforces a minimal-intervention principle: it restricts updates to the only direction that directly transfers the source flow toward the target flow, thereby avoiding orthogonal perturbations that may harm preservation. Parameterizing $u_i(t) = g_i(t)\Delta v_i(t)$ reduces Eq. (4) to a concave quadratic in $g_i(t)$, which admits the closed-form solution

$$g_i^*(t) = \frac{d_i(t)}{\rho}, \qquad u_i^*(t) = \frac{d_i(t)}{\rho}\Delta v_i(t). \tag{5}$$

Eq. (5) shows that TanGO reduces editing to a per-token *allocation* rule: each token receives a gain proportional to its demand $d_i(t)$, while $\Delta v_i(t)$ preserves the full transformation direction. Thus, the entire control law is determined once we specify a principled demand $d_i(t)$. In the next section, we design $d_i(t)$ to be (i) invariant to velocity magnitude, (ii) bounded to prevent spurious spikes, and (iii) informative of whether token i already follows the target-conditioned tangent direction.

4.2 Token-wise Directional Discrepancy

We define $d_i(t)$ from a probabilistic model of *directional* consistency between the source and target-conditioned velocity predictions. Since velocity magnitudes vary across timesteps and assets, we use directional agreement as a scale-stable indicator of whether token i already follows the target-conditioned flow. We normalize token-wise velocities and compute their cosine similarity as

$$\hat{v}_{\mathrm{src},i}(t) = \frac{v_{\mathrm{src},i}(t)}{\|v_{\mathrm{src},i}(t)\| + \varepsilon}, \qquad \hat{v}_{\mathrm{tar},i}(t) = \frac{v_{\mathrm{tar},i}(t)}{\|v_{\mathrm{tar},i}(t)\| + \varepsilon}, \tag{6}$$

$$\cos\theta_i(t) = \hat{v}_{\mathrm{src},i}(t)^\top \hat{v}_{\mathrm{tar},i}(t). \tag{7}$$

Because $\hat{v}_{\mathrm{src},i}, \hat{v}_{\mathrm{tar},i}$ lie on the unit hypersphere, we model $\hat{v}_{\mathrm{tar},i}$ as directional data concentrated around mean direction $\hat{v}_{\mathrm{src},i}$ using the von Mises-Fisher (vMF) likelihood:

$$p(x \mid \mu, \kappa) = C_d(\kappa) \exp\big(\kappa\, \mu^\top x\big), \qquad x = \hat{v}_{\mathrm{tar},i}(t),\ \mu = \hat{v}_{\mathrm{src},i}(t). \tag{8}$$

Up to an additive constant independent of x, the negative log-likelihood satisfies

$$-\log p(\hat{v}_{\mathrm{tar},i} \mid \hat{v}_{\mathrm{src},i}, \kappa) = \mathrm{const} - \kappa\cos\theta_i(t) \ \equiv\ \mathrm{const} + \kappa(1 - \cos\theta_i(t)). \tag{9}$$

We therefore define the demand as the normalized directional negative log-likelihood:

$$d_i(t) = 1 - \cos\theta_i(t), \tag{10}$$

and absorb the concentration to energy ratio κ/ρ into the global scale λ used later. This yields a bounded and geometry-consistent signal: $d_i(t) \in [0, 2]$ and $\|\hat{v}_{\mathrm{src},i} - \hat{v}_{\mathrm{tar},i}\|^2 = 2\, d_i(t)$. Hence, aligned directions receive near-zero demand, while larger mismatch implies stronger actuation. Eq. (10) corresponds to the (scaled) vMF negative log-likelihood over directions after removing constants independent of $\hat{v}_{\mathrm{tar},i}$. [4]

[4] We provide the detailed vMF derivation and related geometric identities in the Appendix.

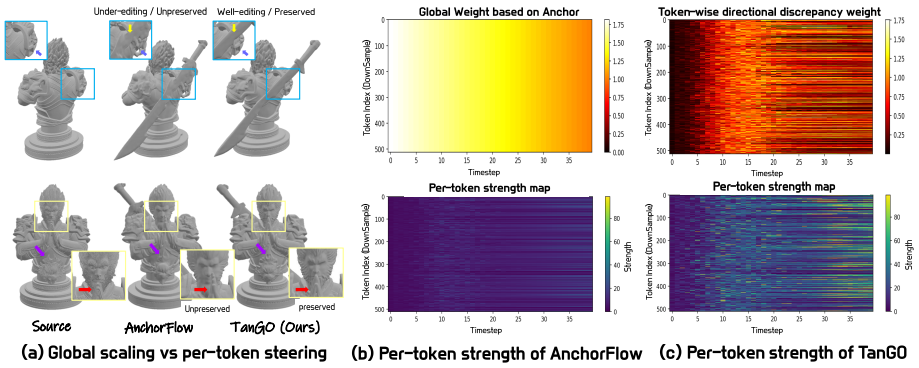

Fig. 6: (a) Comparison between AnchorFlow (global scaling) and TanGO (per-token steering). AnchorFlow exhibits under-editing and unpreserved regions, whereas TanGO yields well-edited results. (b) Token-wise contribution of AnchorFlow across timesteps t, shown via a global weight map and per-token strength map. (c) Per-token strength of TanGO, demonstrating its ability to adaptively assign stronger signals to target tokens and weaker signals to regions to be preserved.

4.3 Adaptive Per-token Steering

Given $d_i(t)$, we define the raw per-token gain $g_i(t) = \frac{d_i(t)}{\rho}$. This decouples *where* to edit from *how strongly* to edit: $g_i(t)$ allocates actuation across tokens based on directional mismatch (thus insensitive to magnitude), while $\Delta v_i(t)$ retains the full direction and magnitude needed to execute the transformation. However, the mean gain $\bar{g}(t) = \frac{1}{N}\sum_i g_i(t)$ can fluctuate over time, inducing unintended variation in the overall guidance energy. We stabilize global edit strength via mean-gain normalization:

$$\lambda_{\text{eff}}(t) = \lambda \cdot \frac{\eta}{\bar{g}(t) + \varepsilon}, \tag{11}$$

where η is a target energy level and ε prevents division by zero. The final token-wise control and dynamics are

$$u_i(t) = \lambda_{\text{eff}}(t)\, g_i(t)\, \Delta v_i(t), \tag{12}$$

$$\dot{x}_t^i = v_{\text{src},i}(t) + u_i(t). \tag{13}$$

Fig. 6 visualizes this decoupling. Unlike a global scaling that offers limited token selectivity, TanGO produces a non-uniform $d_i(t)$ and concentrates effective steering on high-demand tokens while suppressing changes on tokens corresponding to regions to be preserved, enabling localized, mask-free edits.

5 Experiment

5.1 Implementation Details

Settings The editing process integrates over $T = 50$ timesteps from $n_{\max} = 41$ down to $n_{\min} = 1$. We set the classifier-free guidance scales to $s_{\text{src}} = 3.5$ and $s_{\text{tar}} = 7.5$, with additional default parameters $\lambda = 5.0$ and $\eta = 0.2$. The final edited latent X_0^{edit} is directly decoded into 3D space. All experiments are conducted on NVIDIA A100 GPUs.

TanGOEdit Datasets To systematically evaluate the capability of training-free 3D editing and topology preservation, we construct a comprehensive benchmark named TanGOEdit.[5] Comprising exactly 100 high-quality editing samples, our benchmark is built by carefully selecting 3D assets with high aesthetic scores from the Objaverse-XL [11], TRELLIS-500K [31], and Google Scanned Objects (GSO) [12] datasets. Inspired by Nano3D [32], we utilize Qwen 2.5 VL [1] to generate diverse editing instructions.

Metrics. To comprehensively assess the editing efficacy, we utilize three complementary metrics. Semantic alignment is evaluated via CLIP-T [27], which calculates the correspondence between the rendered views of the edited 3D geometry and the target text prompt. For structural and identity preservation, we compute CLIP-I [27] and DINO-I [6] to measure the visual and feature-level similarity between the edited renderings and the source condition image.

5.2 Qualitative Results

Fig. 1, 7 presents qualitative comparisons using our newly constructed TanGOEdit dataset across diverse text-driven 3D editing tasks, including object addition, part replacement, pose change, and object removal. Compared to baseline methods, TanGO achieves highly localized edits while strictly preserving the geometry in regions that are not intended to change. In particular, for rigid editing tasks such as addition, replacement, and removal, baseline methods often suffer from under-editing due to limited editing strength, or they propagate edits to adjacent areas, leading to undesired deformations or the loss of fine structures. These failures often appear as boundary bleeding or collateral distortions in nearby parts (e.g., unintended warping or collapse), even when the requested edit is spatially confined. Conversely, TanGO successfully modifies the target elements without introducing noticeable distortions to adjacent body parts. For non-rigid editing tasks, such as pose changes, our method better maintains global shape coherence and prevents the mesh from breaking or stretching. This indicates that the per-token steering mechanism operates stably even under strong transformations. Importantly, this stability is where naive global guidance often destabilizes the trajectory and amplifies artifacts. Overall, these results demonstrate that TanGO can successfully control and decouple edit strength from

[5] A detailed description of the dataset is provided in the Appendix.

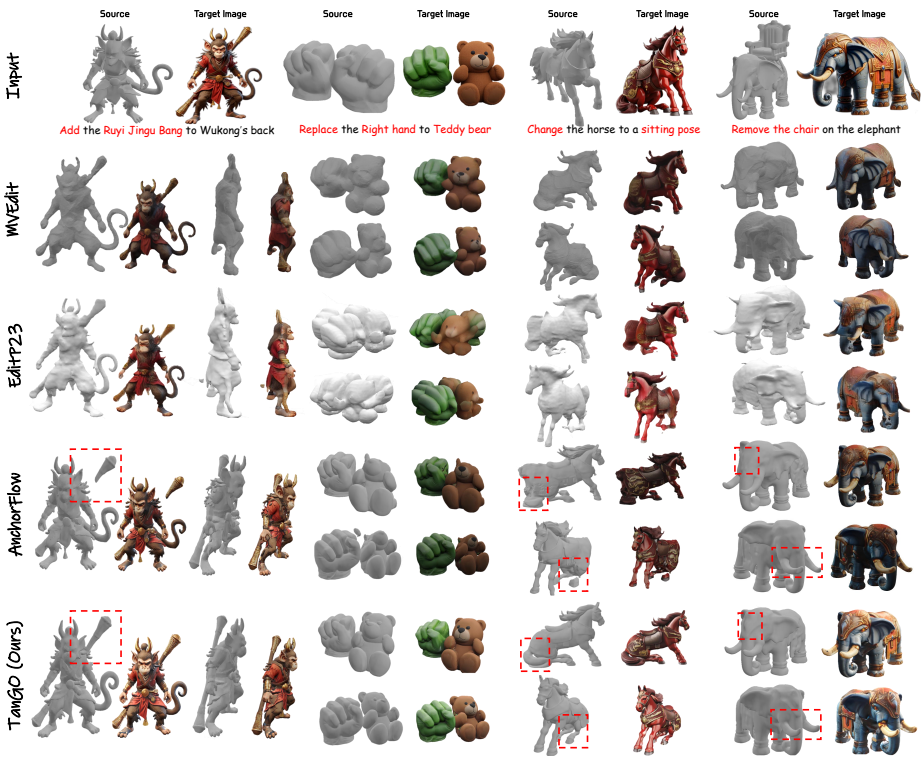

Fig. 7: Qualitative comparisons on text-driven 3D editing. Compared to MVEdit, EditP23, and AnchorFlow, TanGO achieves highly localized edits across diverse tasks while preserving regions that should remain unchanged, effectively avoiding unintended artifacts (see the red boxes).

source preservation, yielding stable and geometry-consistent edits without the need for external masks or additional training.

5.3 Quantitative Results

In this section, we compare **TanGO** with existing 3D editing methods, including MVEdit, EditP23, FlowEdit, and AnchorFlow. As shown in Table 1, TanGO achieves the best performance on the main TanGOEdit benchmark, with DINO-I, CLIP-I, and CLIP-T scores of 0.6510, 0.7679, and 0.2301, respectively. The improvement in CLIP-T indicates stronger alignment with the target editing instruction, while the gains in DINO-I and CLIP-I suggest better preservation of the source identity and visual structure. These results support our claim that adaptive per-token steering provides a more favorable balance between semantic editability and source preservation than prior global trajectory steering methods.

Table 1: Quantitative comparison on TanGOEdit. Our proposed TanGO consistently achieves the highest scores across all metrics: DINO-I, CLIP-I, and CLIP-T, demonstrating superior semantic alignment and generation fidelity compared to state-of-the-art 3D editing baselines.

Method	DINO-I↑	CLIP-I↑	CLIP-T↑
MVEdit [7]	0.5861	0.4994	0.1289
EditP23 [2]	0.5838	0.4971	0.1313
FlowEdit [19]	0.6127	0.7205	0.2081
AnchorFlow [36]	0.6426	0.7492	0.2124
TanGO(Ours)	**0.6510**	**0.7679**	**0.2301**

Table 2: Results of the User Study. Participants strongly preferred TanGO over prior methods across all criteria. Our method significantly outperforms AnchorFlow, particularly in shape preservation (87%) and visual quality (84%), confirming its effectiveness in practical editing scenarios.

Method	Prompt Align.	Visual Quality	Shape Preserv.
AnchorFlow [36]	21%	16%	13%
TanGO(Ours)	79%	84%	87%

The user study in Table 2 further supports this trend. Participants preferred TanGO over AnchorFlow in prompt alignment, visual quality, and shape preservation, with preference rates of 79%, 84%, and 87%, respectively. The largest margin is observed in shape preservation, indicating that TanGO more effectively suppresses unintended changes in preserved regions while maintaining the requested semantic edit.

We additionally conduct a mask-aware 3D evaluation on a separate 150-sample benchmark, consisting of 100 samples from Edit3DBench [21] and 50 TanGOEdit samples annotated in Blender. This separate protocol is necessary because 3D geometric preservation metrics require ground-truth edit masks, which are not available for the full TanGOEdit benchmark. Under this setting, TanGO also outperforms MVEdit, EditP23, FlowEdit, AnchorFlow, VoxHammer, and Nano3D on 3D-native text–shape alignment and mask-aware geometric preservation metrics, including $\text{Uni3D}_{\text{txt}}$, $\text{CD}_{\text{preserve}}$, and $\text{NC}_{\text{preserve}}$. Detailed quantitative results and qualitative comparisons are provided in the supplementary material.

5.4 Ablation Study

Hyperparameter sensitivity We study the effect of hyperparameters on performance by presenting both qualitative and quantitative analyses. We first examine how varying λ changes the trade-off between edit strength and preservation, and then report CLIP-based alignment scores over a range of (λ, η) settings.

Fig. 8(a) shows qualitative results when increasing λ while fixing η=0.2. As λ grows, edits become stronger, but overly large values may start to degrade preservation. Across the tested examples, we observe that our default setting provides a favorable balance between effective editing and geometric preservation.

Fig. 8(b) visualizes CLIP-based alignment scores (CLIP-I and CLIP-T) evaluated on multiple (λ, η) combinations. Specifically, we vary $\lambda \in \{1, 2, 3, 5, 7, 10\}$ and combine it with $\eta \in \{0.05, 0.1, 0.2, 0.3, 0.5\}$. The scores change smoothly

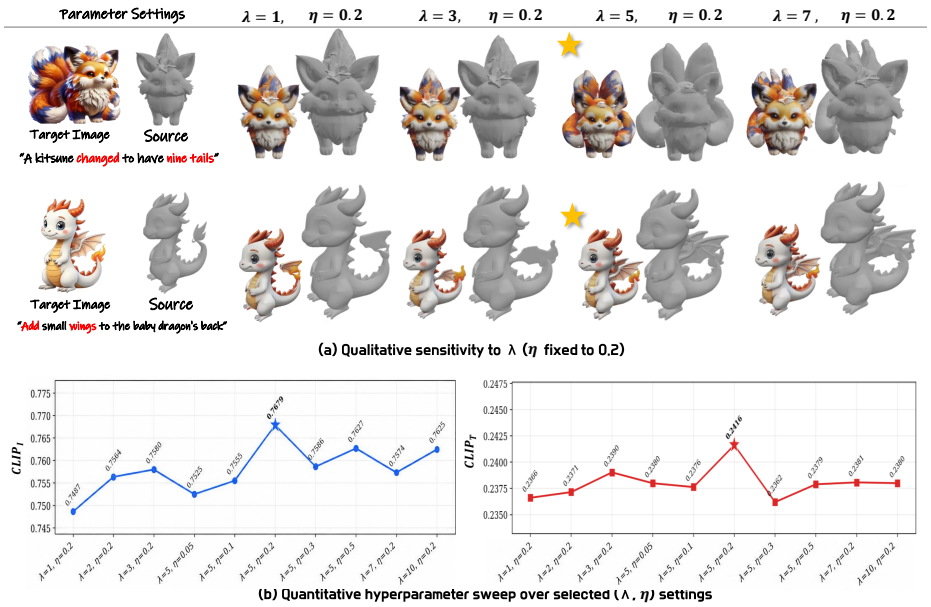

Fig. 8: Qualitative and Quantitative Analysis of Parameter Selection (a) Qualitative comparison under varying λ settings ($\eta = 0.2$). The optimal setting $\lambda = 5$ (marked with a star) achieves clear geometric modifications, such as the distinct nine tails and wings, while preserving source integrity. (b) Quantitative hyperparameter sweep on **TanGOEdit** datasets. The CLIP-I and CLIP-T scores peak at $\lambda = 5, \eta = 0.2$, justifying our choice of optimal default parameters.

across settings rather than exhibiting sharp spikes or drops, and the default choice (λ=5, η=0.2) lies near a high-performing region. Overall, these results suggest that TanGO behaves stably under moderate variations of (λ, η).

Effect of gain normalization To verify the necessity of the proposed gain normalization, we conduct an ablation study by replacing the timestep dependent scaling factor $\lambda_{\text{eff}}(t) = \lambda \cdot \frac{\eta}{\bar{g}(t)+\varepsilon}$ with a fixed constant λ. This experiment aims to analyze how the mechanism that compensates for the mean gain $\bar{g}(t)$ which fluctuates significantly along the integration trajectory affects the final editing quality.

Our results show that without gain normalization, it becomes difficult to maintain a consistent steering strength throughout the entire editing process. Even when the relative weights for each token remain the same, the lack of normalization causes the guidance to be too weak at certain timesteps, leading to *under-editing* where the target changes are not fully realized. Conversely, at other stages, the guidance can become overly aggressive, potentially harming the structural stability of the mesh.

Fig. 9(a) visually demonstrates this effect through a representative example. When λ_{eff} is set to a fixed value, the model lacks sufficient power to implement

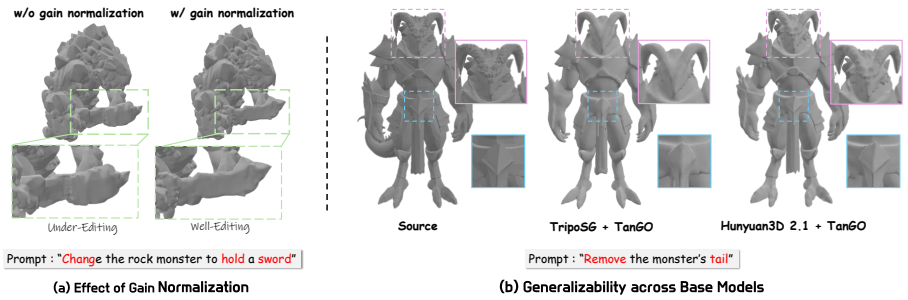

Fig. 9: Qualitative Analysis of TanGO Components. (a) Gain Normalization prevents under-editing, ensuring accurate geometric changes (e.g., holding a sword). (b) TanGO shows high generalizability across various base models like TripoSG and Hunyuan3D 2.1.

the target shape. In contrast, the full TanGO model with gain normalization provides a balanced guide across all timesteps. This allows for strong and stable edits while preserving the structural integrity of the surrounding geometry. Ultimately, gain normalization ensures that the global edit strength is optimized without changing the per-token steering patterns, making the editing process more robust even during significant shape transformations.

Generalizability of TanGO To evaluate whether TanGO generalizes beyond Hunyuan3D 2.1, we apply the same token-wise steering mechanism to TripoSG, another VecSet-based 3D foundation model. As shown in Fig. 9(b), TanGO successfully performs the requested semantic edit without additional training. Quantitative results further show that TanGO improves over AnchorFlow under both 2D alignment metrics and mask-aware 3D preservation metrics, supporting its applicability beyond a single backbone. We observe that TripoSG results exhibit slightly smoother geometry than Hunyuan3D 2.1, which is mainly due to the reconstruction capacity of the underlying 3D VAE rather than the TanGO optimization process itself. Detailed cross-model results are provided in the supplementary material.

Ablation on Per-token Gain Design To verify that the proposed directional demand is not merely an arbitrary token-wise rescaling, we compare TanGO with simpler gain designs, including global scaling, L2-magnitude-based gain $|v_{\mathrm{tar},i} - v_{\mathrm{src},i}|$, and inner-product-based gain $\langle v_{\mathrm{src},i}, v_{\mathrm{tar},i} \rangle$. TanGO consistently performs better than these alternatives, supporting the use of a bounded and scale-invariant directional discrepancy rather than raw velocity magnitude or unnormalized similarity. Full results are provided in the supplementary material.

Comparison of Time Cost As shown in Table 3, TanGO achieves an inference time of 28.73s per edit, which is comparable to FlowEdit (28.13s) and An-

Table 3: Comparison of inference time across different 3D editing frameworks.

	MVEdit	EditP23	FlowEdit	AnchorFlow	**TanGO (Ours)**
Time (s)	632.45	59.32	28.13	28.76	28.73

chorFlow (28.76s), while being much faster than MVEdit (632.45s) and EditP23 (59.32s). This indicates that the proposed per-token steering and gain normalization introduce negligible computational overhead despite adding token-level control to the editing process. As a result, TanGO maintains the efficiency of recent flow-based editing methods while providing more localized and preservation-aware edits, making it practical for iterative 3D design workflows.

6 Conclusion

In this paper, we introduced TanGO, a novel, training-free framework for 3D mesh editing that leverages the rich priors of pre-trained flow-based 3D foundation models. By formulating the editing process as an instantaneous one-step optimal control problem, TanGO achieves precise and localized semantic modifications. Our core mechanism, adaptive per-token steering, dynamically scales the control input based on directional discrepancy in the tangent space. This allows for stable editing in targeted regions while preserving the geometry of unedited areas, all without the need for costly fine-tuning or model retraining. While our extensive evaluations demonstrate TanGO's robust generalizability across different network architectures, we also identify that the ultimate preservation of high-frequency geometric details is currently bounded by the representational capacity of the underlying model's 3D VAE. Overcoming this VAE bottleneck potentially through the integration of more expressive latent representations or hybrid pixel-3D refinement strategies presents a promising avenue for future work. Ultimately, TanGO offers a highly efficient and adaptable solution, advancing the practical utility of 3D foundation models for high-fidelity, user-guided 3D creation.

Acknowledgements

This work was supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2021-II211381, Development of Causal AI through Video Understanding and Reinforcement Learning, and Its Applications to Real Environments) and (No.RS-2022-II220184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics).

References

1. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report (2025), https://arxiv.org/abs/2502.13923
2. Bar-On, R., Cohen-Bar, D., Cohen-Or, D.: Editp23: 3d editing via propagation of image prompts to multi-view. arXiv preprint arXiv:2506.20652 (2025)
3. Barda, A., Gadelha, M., Kim, V.G., Aigerman, N., Bermano, A.H., Groueix, T.: Instant3dit: Multiview inpainting for fast editing of 3d objects. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 16273–16282 (2025)
4. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18392–18402 (2023)
5. Cao, M., Wang, X., Qi, Z., Shan, Y., Qie, X., Zheng, Y.: Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 22560–22570 (2023)
6. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
7. Chen, H., Shi, R., Liu, Y., Shen, B., Gu, J., Wetzstein, G., Su, H., Guibas, L.: Generic 3d diffusion adapter using controlled multi-view editing. arXiv preprint arXiv:2403.12032 (2024)
8. Chen, R., Chen, Y., Jiao, N., Jia, K.: Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 22246–22256 (2023)
9. Chen, R., Zhang, J., Liang, Y., Luo, G., Li, W., Liu, J., Li, X., Long, X., Feng, J., Tan, P.: Dora: Sampling and benchmarking for 3d shape variational auto-encoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16251–16261 (2025)
10. Chen, Y., Chen, Z., Zhang, C., Wang, F., Yang, X., Wang, Y., Cai, Z., Yang, L., Liu, H., Lin, G.: Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 21476–21485 (2024)
11. Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., et al.: Objaverse-xl: A universe of 10m+ 3d objects. Advances in Neural Information Processing Systems **36**, 35799–35813 (2023)
12. 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: 2022 International Conference on Robotics and Automation (ICRA). pp. 2553–2560. Ieee (2022)
13. Gao, W., Aigerman, N., Groueix, T., Kim, V., Hanocka, R.: Textdeformer: Geometry manipulation using text guidance. In: ACM SIGGRAPH 2023 conference proceedings. pp. 1–11 (2023)
14. Haque, A., Tancik, M., Efros, A.A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 19740–19750 (2023)

15. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: Lrm: Large reconstruction model for single image to 3d. In: The Twelfth International Conference on Learning Representations (2023)
16. Huang, Z., Guo, Y.C., Wang, H., Yi, R., Ma, L., Cao, Y.P., Sheng, L.: Mv-adapter: Multi-view consistent image generation made easy. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16377–16387 (2025)
17. 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)
18. Koo, G., Yoon, S., Hong, J.W., Yoo, C.D.: Flexiedit: Frequency-aware latent refinement for enhanced non-rigid editing. In: European Conference on Computer Vision. pp. 363–379. Springer (2024)
19. Kulikov, V., Kleiner, M., Huberman-Spiegelglas, I., Michaeli, T.: Flowedit: Inversion-free text-based editing using pre-trained flow models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19721–19730 (2025)
20. Lai, Z., Zhao, Y., Zhao, Z., Liu, H., Lin, Q., Huang, J., Guo, C., Yue, X.: Lattice: Democratize high-fidelity 3d generation at scale. arXiv preprint arXiv:2512.03052 (2025)
21. Li, L., Huang, Z., Feng, H., Zhuang, G., Chen, R., Guo, C., Sheng, L.: Voxhammer: Training-free precise and coherent 3d editing in native 3d space. arXiv preprint arXiv:2508.19247 (2025)
22. 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. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025)
23. Liang, Y., Yang, X., Lin, J., Li, H., Xu, X., Chen, Y.: Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6517–6526 (2024)
24. 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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 300–309 (2023)
25. Michel, O., Bar-On, R., Liu, R., Ben-Efraim, S., Hanocka, R.: Text2mesh: Text-driven neural stylization for meshes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13492–13502 (2022)
26. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988 (2022)
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: International conference on machine learning. pp. 8748–8763. PMLR (2021)
28. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 2149–2159 (2022)
29. 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. Advances in neural information processing systems **36**, 8406–8441 (2023)

30. Wu, S., Lin, Y., Zhang, F., Zeng, Y., Xu, J., Torr, P., Cao, X., Yao, Y.: Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. Advances in Neural Information Processing Systems **37**, 121859–121881 (2024)
31. 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: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 21469–21480 (2025)
32. Ye, J., Xie, S., Zhao, R., Wang, Z., Yan, H., Zu, W., Ma, L., Zhu, J.: Nano3d: A training-free approach for efficient 3d editing without masks. arXiv preprint arXiv:2510.15019 (2025)
33. Yoon, S., Koo, G., Hong, J.W., Yoo, C.D.: Dni: Dilutional noise initialization for diffusion video editing. In: European Conference on Computer Vision. pp. 180–195. Springer (2024)
34. Zhang, B., Tang, J., Niessner, M., Wonka, P.: 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. ACM Transactions On Graphics (TOG) **42**(4), 1–16 (2023)
35. Zhang, L., Wang, Z., Zhang, Q., Qiu, Q., Pang, A., Jiang, H., Yang, W., Xu, L., Yu, J.: Clay: A controllable large-scale generative model for creating high-quality 3d assets. ACM Transactions on Graphics (TOG) **43**(4), 1–20 (2024)
36. Zhou, Z., Ma, F., Gui, C., Xia, X., Fan, H., Yang, Y., Chua, T.S.: Anchorflow: Training-free 3d editing via latent anchor-aligned flows. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14387–14397 (2026)