HiFi3D: High-Fidelity 3D Asset Editing for Sparse Voxel Representations

Anonymous authors
Paper under double-blind review



Figure 1: **Editing results of HiFi3D across diverse 3D editing tasks.** HiFi3D enables diverse editing operations, including add, remove, replace, and texture control, while maintaining high-fidelity geometry and consistent visual appearance. We support superposed way of multi-editing include both basis and texture editing.

Abstract

Existing 3D editing methods suffer from two critical limitations: optimization-based approaches require lengthy per-instance tuning, while recent flow-based methods struggle to preserve fine geometric structures and material details during editing. We introduce HiFi3D, a training-free 3D editing framework built upon TRELLIS2's factored O-Voxel representation, enabling precise instruction-guided modifications without model retraining. Our method overcomes two fundamental challenges: fragility of sparse voxel topology under stochastic perturbations, and the entanglement between geometric structure and surface appearance in joint editing pipelines. For geometry manipulation, we propose Dual-Space Geometry-Preserving Flow Editing, which combines mask-free regional dynamics with dual-level preservation, maintaining trajectory stability while voxel-supervised alignment ensures decoded geometric consistency through a differentiable decoder. For appearance control, we introduce Geometry-Conditioned Material Editing with PBR consistency constraints, reformulating texture editing as a geometry-aware process where multi-sample velocity estimation stabilizes material flow and decoded-space regularization enforces physically plausible PBR attributes. Extensive experiments demonstrate that HiFi3D achieves accurate instruction-following

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

modifications while outperforming state-of-the-art baselines in both visual quality and geometric fidelity.

1 INTRODUCTION

High-quality 3D asset editing is important for digital content creation, games, virtual reality, simulation, and embodied AI Lu et al. (2024). The success of large-scale 2D diffusion models Rombach et al. (2022); Esser et al. (2024) and instruction-based image editing Brooks et al. (2023); Meng et al. (2022); Hertz et al. (2023) has motivated their adaptation to 3D, which is harder than 2D editing because a local semantic change may alter surface topology, sparse spatial support, fine geometric detail, and physically based material attributes at the same time.

Existing 3D such as Optimization-based methods Gao et al. (2023); Kim et al. (2024); Haque et al. (2023); Zhuang et al. (2023); Wang et al. (2024) deform meshes or Gaussians via differentiable rendering, offering control but requiring costly per-asset optimization. Reconstruction-based pipelines Chen et al. (2024a); Bar-On et al. (2025) edit images then reconstruct 3D, benefiting from 2D priors but risking blur and inconsistency. Optimization-based text-to-3D methods Poole et al. (2023); Lin et al. (2023); Wang et al. (2023) and feed-forward reconstruction Liu et al. (2024a) leverage score distillation for content creation. While inversion-free flow editing Kulikov et al. (2024); Zhou et al. (2026), built upon rectified flow Liu et al. (2023b), enables training-free 3D editing via source-target velocity differences. However, lattest O-Voxel's sparse geometry illustrates small perturbations may flip voxel activations, induce decoded geometric inconsistencies. Global velocity lacks explicit region preservation, and latent errors amplify through decoding.

We propose a training-free editing framework for factored O-Voxel latents. For geometry, we combine the condition-driven edit velocity with invariant-region latent anchoring and a voxel-space alignment objective that compares decoded occupancy along source and edited trajectories. This dual-space mechanism preserves non-edited structure both before and after decoding. For material, we formulate appearance editing as a geometry-conditioned material flow, estimate edit velocities by multi averaging and regularize decoded PBR attributes for stable, coherent surface appearance. Our system outputs edit-category-specific data. It supports geometry modification, object insertion and deletion, replacement, material changes, and coupled style transformations in a unified native 3D editing pipeline. Experiments show improved instruction adherence, source preservation, and geometric fidelity compared with optimization-based, reconstruction-based, and latent flow editing baselines.

Our contributions are summarized as follows:

- Introduce a training-free 3D editing framework for factored O-Voxel latents.
- Propose dual-space geometry-preserving flow editing, combine invariant-region latent alignment with voxel supervision through a differentiable decoder.
- Develop geometry-conditioned material editing with multi-sample velocity estimation and PBR-aware decoded-space regularization.
- Evaluate the method across diverse 3D editing tasks and show improved instruction following, source preservation, and high-fidelity reconstruction over strong baselines.

2 RELATED WORK

2.1 NATIVE 3D GENERATION

3D generation has moved from explicit surface modeling toward learned priors over neural fields, 3D Gaussians, triplanes, multi-view features, and structured latent representations Mildenhall et al. (2020); Kerbl et al. (2023); Hong et al. (2023); Li et al. (2023). Multi-view diffusion models Liu et al. (2023a); Shi et al. (2024); Liu et al. (2024b); Long et al. (2024) generate consistent views for 3D reconstruction, while alternative paradigms use point clouds Nichol et al. (2022) or hybrid explicit-implicit representations Shen et al. (2021). These models can synthesize high-quality objects, but many rendering-centric representations entangle shape, material, and viewpoint-dependent appear-

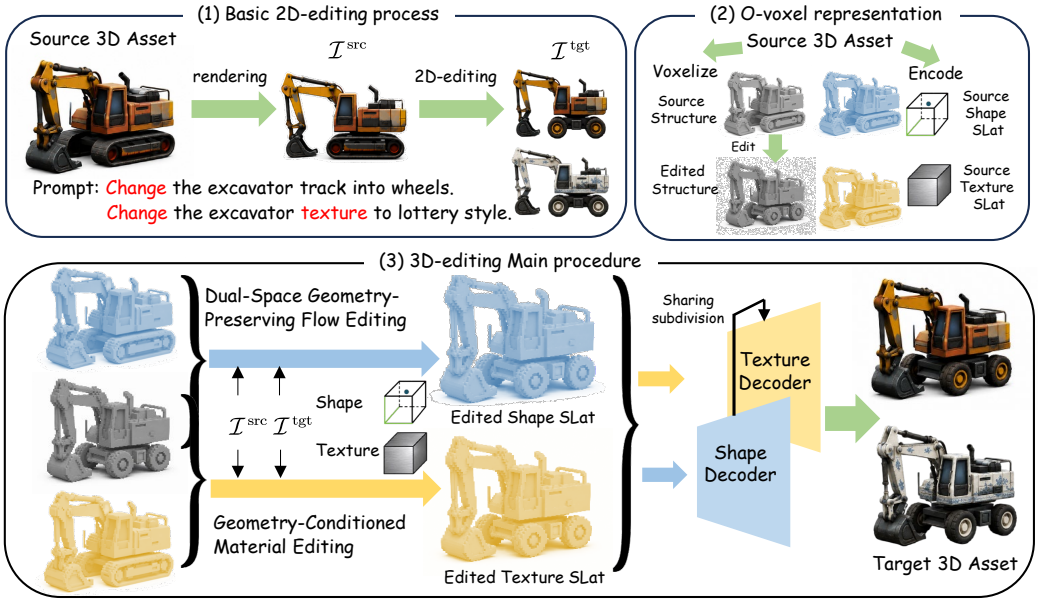

Figure 2: **Overview of HiFi3D.** From a source 3D asset and text instruction, a 2D editor generates a target reference image; the source is encoded into O-Voxel geometry and material latents; dual-branch editing applies condition-driven flow with latent anchoring and voxel alignment for geometry, and geometry-conditioned flow with PBR regularization for material; the edited latents are decoded into the final high-fidelity 3D asset.

ance. Such entanglement remains problematic for editing as latent update may cause uncontrollable geometry and appearance.

Native 3D generators based on sparse spatial or flow-based latents provide a better substrate for editable assets Xiang et al. (2024); Zhao et al. (2025); Li et al. (2025). O-Voxel represents objects with active voxels whose features separate geometry from PBR material attributes, giving the generator explicit spatial support and a useful geometry-appearance factorization. Nevertheless, generation is not editing: a generator only needs to produce a plausible asset, while an editor must preserve the source identity and unedited regions. Our work therefore uses O-Voxel as the native representation, but adds explicit trajectory constraints for source-preserving geometry and material control.

2.2 Instruction-Guided 3D Editing

Instruction-guided 3D editing methods differ mainly in where they apply the edit. Optimization-based methods directly update explicit 3D assets using differentiable rendering, visual-language supervision, or score distillation Poole et al. (2023); Wang et al. (2023); Gao et al. (2023); Kim et al. (2024); Haque et al. (2023); Zhuang et al. (2023); Chen et al. (2024b); Wang et al. (2024). They provide geometric control but require expensive per-instance optimization. Image- and multi-view-based methods edit rendered observations and reconstruct the 3D result Chen et al. (2024a); Bar-On et al. (2025); they benefit from strong 2D editing priors Brooks et al. (2023); Hertz et al. (2023); Ruiz et al. (2023) but can lose fine geometry, introduce view inconsistency, or drift from the source object during cross-representation transfer.

Latent generative editing is more efficient because it adapts pretrained models without task-specific retraining. FlowEdit uses source-target velocity differences for inversion-free flow editing Kulikov et al. (2024), and AnchorFlow improves trajectory alignment with latent anchors Zhou et al. (2026). However, these formulations generally treat the state as a continuous latent. O-Voxel editing requires representation-aware control: sparse activations, connectivity variables, and nonlinear decoding can turn small latent errors into topology changes, while material attributes require different constraints from geometry. Our method addresses these issues with geometry preservation in latent and voxel space and material regularization in decoded PBR space.

3 METHOD

As illustrated in Figure 2, HiFi3D is a training-free framework for instruction-guided 3D editing. Given source asset X^{src} and text instruction c_{edit}, we render source view $\mathcal{I}^{\text{src}}$ and apply 2D editor Nano Banana Pro to produce target reference $\mathcal{I}^{\text{tgt}}$. The source is encoded into geometry and material latents, edited through separate flow processes conditioned on the image pair. Geometry editing combines condition-driven velocity with latent anchoring and voxel alignment to suppress drift. Material editing updates latents via multi-sample estimation and PBR regularization while keeping geometry fixed. The edited latents decode into the final asset following the instruction while preserving source identity.

3.1 PRELIMINARY

3.1.1 O-VOXEL REPRESENTATION

We adopt the method of Ovoxel from TRELLIS.2 Xiang et al. (2025), which leverages self-supervised visual features Oquab et al. (2024) and rectified flow Liu et al. (2023b) for high-quality 3D generation. An O-Voxel represents a 3D asset as a sparse set of active voxels,

$$\mathcal{V} = \left\{ \left(\mathbf{f}_i^{\text{shape}}, \mathbf{f}_i^{\text{mat}}, \mathbf{p}_i \right) \right\}_{i=1}^{L}, \tag{1}$$

where $\mathbf{p}_i$ is the voxel coordinate, $\mathbf{f}_i^{\text{shape}}$ describes local geometry, and $\mathbf{f}_i^{\text{mat}}$ stores material attributes. Geometry is represented by a Flexible Dual Grid,

$$\mathbf{f}_i^{\text{shape}} = (\mathbf{v}_i, \boldsymbol{\delta}_i, \boldsymbol{\gamma}_i), \tag{2}$$

where $\mathbf{v}_i$ denotes the dual-grid vertex, $\boldsymbol{\delta}_i$ encodes local edge connectivity, and $\boldsymbol{\gamma}_i$ controls adaptive face splitting. Material properties follow the standard PBR formulation,

$$\mathbf{f}_i^{\text{mat}} = (\mathbf{c}_i, m_i, r_i, \alpha_i), \tag{3}$$

where $\mathbf{c}_i$, m_i, r_i, and α_i denote the base color, metallicity, roughness, and opacity, respectively. O-Voxel unifies geometry and appearance in a sparse volumetric representation and supports efficient mesh reconstruction for rendering.

3.1.2 INVERSION-FREE EDITING

The inversion-free editing paradigm, originally proposed in FlowEdit Kulikov et al. (2024), provides a direct approach to manipulate latent representations without explicit inversion steps. Given a source asset X^{src}, source condition c^{src} and target condition c^{tgt}, the editing objective is to generate X^{tgt} that satisfies c^{tgt} while preserving the identity of X^{src}.

We establish temporal boundaries:

$$x_{t=1} = x^{\text{src}}, \quad x_{t=0} = x^{\text{tgt}}, \tag{4}$$

such that the ODE solution connects the initial source structure to the target edited result.

The key idea is to derive an edit velocity v_{edit} by computing trajectory differences under different conditioning signals. Given source and target condition, the pretrained rectified-flow model defines straight-line paths:

$$x_t^c = (1-t)x_0^c + tx_1^c, c \in \{src, tgt\} \tag{5}$$

linking clean state x_0^c to noise state x_1^c. The learned velocity network generates:

$$\frac{dx_t^c}{dt} = v_\theta(x_t^c, t \mid \mathcal{I}^c). \tag{6}$$

By setting a common noise endpoint $x_1^{\text{src}} = x_1^{\text{tgt}}$, we define the editing path as:

$$x_t = x_0^{\text{src}} - x_t^{\text{src}} + x_t^{\text{tgt}}. \tag{7}$$

Hence structural edits leads to differentiating yields of velocity:

$$v_{\text{edit}}(x_t, t) = v_\theta(x_t^{\text{tgt}}, t \mid \mathcal{I}^{\text{tgt}}) - v_\theta(x_t^{\text{src}}, t \mid \mathcal{I}^{\text{src}}). \tag{8}$$

3.2 Dual-Space Geometry-Preserving Flow Editing

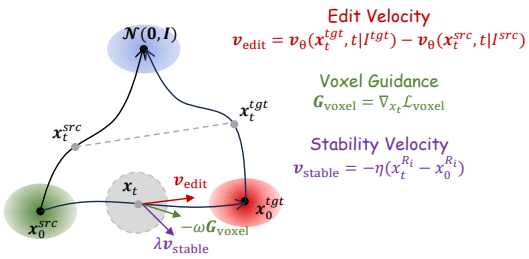

Figure 3: Diagram of dual-sensitivity flow decomposition.

Mask-Free Regional Dynamic trajectory Prior work editing Kulikov et al. (2024) models the editing process through a masked ODE:

$$\frac{dx_t}{dt} = M_l \odot v_{\text{edit}}(x_t, t), \tag{9}$$

where M_l is a binary mask restricting updates to editable regions. However, hard masking disrupts topological coherence in dual-grid structures and lacks active geometric preservation for invariant regions.

Let $x_t \in \mathbb{R}^{N\times D}$ denote the latent representation at time t, where each latent token $x_t^{(i)}$ corresponds to an O-Voxel element $f_i^{\text{shape}} = (v_i, \delta_i, \gamma_i)$. We define the editable region in latent space based on 2D conditions $\mathcal{I}^{\text{src}}$ and $\mathcal{I}^{\text{tgt}}$:

$$\mathcal{R}_e = \{x_t^{(i)} \mid \pi(v_i) \in \mathcal{D}(\mathcal{I}^{\text{src}}, \mathcal{I}^{\text{tgt}})\}, \tag{10}$$

where $\pi(\cdot)$ projects the corresponding dual-grid vertex to the image plane and $\mathcal{D}(\cdot)$ identifies semantic difference regions. The invariant region is $\mathcal{R}_i = \{x_t^{(i)} \mid x_t^{(i)} \notin \mathcal{R}_e\}$.

We propose a mask-free ODE with explicit geometric preservation:

$$\frac{dx_t}{dt} = v_{\text{edit}}(x_t, t, \mathcal{R}_e) + \lambda v_{\text{stable}}(x_t, \mathcal{R}_i), \tag{11}$$

where v_{edit} is the global edit velocity defined in Eq. (8), while v_{stable} preserves the invariant region $\mathcal{R}_i$. Unlike hard masking in Eq. (9), the region partition is used only for preservation and never masks the edit velocity.

Voxel-Supervised Geometry Alignment While the region-conditioned trajectory provides spatial awareness, numerical integration may introduce accumulated errors that remain small in latent space but amplify into significant geometric inconsistencies after decoding, particularly in the invariant region $\mathcal{R}_i$.

Geometry Consistency Constraint. The key insight is that small latent deviations can cause substantial decoded geometry changes due to decoder nonlinearity. Let $P_t(\cdot) : \mathbb{R}^{N\times D} \to [0,1]^{H\times W\times D}$ denote the voxel decoder at time t. Ideally, the decoded occupancy in $\mathcal{R}_i$ should remain invariant throughout the trajectory:

$$P_t(x_t^{\mathcal{R}_i}) = P_0(x_0^{\mathcal{R}_i}), \quad \forall t \in [0,1]. \tag{12}$$

However, even when the latent preservation term v_{stable} maintains $\|x_t^{\mathcal{R}_i} - x_0^{\mathcal{R}_i}\| \approx 0$, decoder nonlinearity can still cause geometric drift: $\|P_t(x_t^{\mathcal{R}_i}) - P_t(x_0^{\mathcal{R}_i})\| \gg 0$. This motivates the need for explicit geometric-level supervision.

Voxel-Space Alignment Objective. We reformulate the invariance constraint as a trajectory alignment objective in voxel space. For each voxel coordinate $(i,j,k) \in \mathcal{V}(\mathcal{R}_i)$ corresponding to the invariant region, we measure the occupancy discrepancy between source and target trajectories:

$$\mathcal{L}_{\text{voxel}}(x_t) = \sum_{(i,j,k)\in\mathcal{V}(\mathcal{R}_i)} \|P_t(x_t^{\text{tgt}})_{ijk} - P_t(x_t^{\text{src}})_{ijk}\|^2. \tag{13}$$

This formulation enforces both trajectories to decode to the same geometric structure in $\mathcal{R}_i$, implicitly anchoring them to the source geometry. Unlike latent-space preservation, this objective directly measures structural consistency at the geometric representation level.

Algorithm 1 Dual-Space Geometry-Preserving Flow Editing

Require: Source latent x_0^{src}, timesteps $\{t_i\}_{i=0}^{T}$, source/target conditions $\mathcal{I}^{\text{src}}, \mathcal{I}^{\text{tgt}}$, editable region $\mathcal{R}_e$, hyperparameters λ, ω, η
Ensure: Edited latent x_0^{tgt}
Initialize: $x_T \leftarrow x_0^{\text{src}}$
for $i = T$ **down to** 1 **do**
  // Sample shared noise and construct trajectories
  $\varepsilon_{t_i} \sim \mathcal{N}(0, I)$
  $x_{t_i}^{\text{src}} \leftarrow (1 - t_i)x_0^{\text{src}} + t_i\varepsilon_{t_i}$
  $x_{t_i}^{\text{tgt}} \leftarrow x_{t_i}^{\text{src}} + x_{t_i} - x_0^{\text{src}}$
  // Compute edit velocity
  $v_{\text{edit}}(x_{t_i}, t_i) \leftarrow v_\theta(x_{t_i}^{\text{tgt}}, t_i \mid \mathcal{I}^{\text{tgt}}) - v_\theta(x_{t_i}^{\text{src}}, t_i \mid \mathcal{I}^{\text{src}})$
  // Compute preservation velocity for invariant region
  $v_{\text{stable}}(x_{t_i}, \mathcal{R}_i) \leftarrow -\eta(x_{t_i}^{\mathcal{R}_i} - x_0^{\mathcal{R}_i})$
  // Compute voxel-supervised geometry gradient
  $P_{t_i}^{\text{src}} \leftarrow P_{t_i}(x_{t_i}^{\text{src}}), \quad P_{t_i}^{\text{tgt}} \leftarrow P_{t_i}(x_{t_i}^{\text{tgt}})$
  $\mathcal{L}_{\text{voxel}} \leftarrow \sum_{(i,j,k)\in\mathcal{V}(\mathcal{R}_i)} \|P_{t_i}^{\text{tgt}}[i,j,k] - P_{t_i}^{\text{src}}[i,j,k]\|^2$
  $G_{\text{voxel}} \leftarrow \nabla_{x_{t_i}} \mathcal{L}_{\text{voxel}}$
  // Integrated flow update
  $x_{t_{i-1}} \leftarrow x_{t_i} + \Delta t \left(v_{\text{edit}}(x_{t_i}, t_i; \mathcal{R}_e) + \lambda v_{\text{stable}}(x_{t_i}, \mathcal{R}_i) - \omega G_{\text{voxel}}\right)$
end for
return $x_0^{\text{tgt}} \leftarrow x_{t_0}$

Although the loss is defined in voxel space, optimization occurs in latent space. The gradient G_{voxel} with respect to the latent state is computed via the chain rule:

$$G_{\text{voxel}} = \nabla_{x_t} \mathcal{L}_{\text{voxel}} = \frac{\partial P_t}{\partial x_t}^{\top} \cdot \nabla_{P_t} \mathcal{L}_{\text{voxel}}, \tag{14}$$

where the Jacobian $\frac{\partial P_t}{\partial x_t}$ backpropagates geometric discrepancies through the differentiable voxel decoder. This gradient provides a correction signal that adjusts the latent state to minimize decoded geometric deviations.

Geometry-Supervised Flow Update. Integrating the voxel-supervised gradient into the flow dynamics yields our final trajectory equation:

$$\frac{dx_t}{dt} = v_{\text{edit}}(x_t, t; \mathcal{R}_e) + \lambda v_{\text{stable}}(x_t, \mathcal{R}_i) - \omega G_{\text{voxel}} \tag{15}$$

where $v_{\text{stable}}(x_t, \mathcal{R}_i) = -\eta(x_t^{\mathcal{R}_i} - x_0^{\mathcal{R}_i})$ provides latent-level anchoring with complexity $O(N \cdot D)$, and $\nabla_{x_t}\mathcal{L}_{\text{voxel}}$ enforces geometry-level consistency with complexity $O(|\mathcal{V}(\mathcal{R}_i)|)$. This dual mechanism ensures structural fidelity in $\mathcal{R}_i$ despite decoder nonlinearity and integration errors. The complete algorithm is summarized in Algorithm 1. We fix $\lambda = 0.5$, $\omega = 0.1$, and $\eta = 0.05$ during experiments.

3.3 Geometry-Conditioned Material Editing

Prior 3D editing methods typically rely directly on pretrained generative models for material output, leading to uncontrolled material drift in non-edited regions and physically implausible PBR parameter combinations. In contrast, our approach introduces explicit PBR consistency constraints and region-aware material flow based on O-Voxel's material features $f_i^{\text{mat}} = (c_i, m_i, r_i, \alpha_i)$, ensuring physically plausible edits while preserving the source material in invariant regions.

Material Velocity Estimation For material editing, we construct the material edit velocity through multi-sample expectation estimation:

$$v_{\text{edit}}^{\text{mat}}(z_t, t) = \frac{1}{K}\sum_{k=1}^{K}\left[v_\theta(z_t + t\xi_k, t \mid \mathcal{I}^{\text{tgt}}) - v_\theta(z_t + t\xi_k, t \mid \mathcal{I}^{\text{src}})\right], \tag{16}$$

Table 1: Quantitative comparison with state-of-the-art methods. Best results in **bold**.

Method	Geometry Metrics			Appearance Metrics				Time(s)↓
	CD($\mathcal{R}_i$)↓	CD($\mathcal{R}_e$)↓	F-Score↑	CLIP-T↑	DINO-I↑	LPIPS↓	FID↓	
MVEdit	0.0234	0.0178	0.623	0.267	0.683	0.312	42.3	35.2
Trellis2	0.0198	0.0156	0.687	0.281	0.701	0.298	38.7	18.3
Easy3E	0.0342	0.0198	0.591	0.253	0.672	0.334	45.8	22.1
Nano3D	0.0167	0.0151	0.698	0.289	0.708	0.287	36.4	28.6
Ours	**0.0123**	**0.0109**	**0.756**	**0.312**	**0.734**	**0.245**	**28.1**	**16.5**

where $\xi_k \sim \mathcal{N}(\mathbf{0}, I)$ are independent noise samples and z_t denotes the material latent state. Due to material's continuous structure without discrete topology boundaries, fewer samples suffice: empirically $K_{\text{mat}} = 3$ achieves stable edits compared to $K_{\text{shape}} = 5$ for geometry.

PBR Consistency Constraint To enforce physical plausibility, we introduce a material consistency objective that captures fundamental PBR constraints. Let $D_t(z_t) \to (c, m, r, \alpha)$ denote the material decoder. We define:

$$\mathcal{L}_{\text{pbr}}(z_t) = \lambda_m \mathcal{L}_{\text{metal}}(z_t) + \lambda_r \mathcal{L}_{\text{smooth}}(z_t) + \lambda_c \mathcal{L}_{\text{range}}(z_t), \tag{17}$$

Integrating the PBR consistency gradient into the material flow dynamics yields:

$$\frac{dz_t}{dt} = v^{\text{mat}}_{\text{edit}}(z_t, t) - \eta \nabla_{z_t} \mathcal{L}_{\text{pbr}}(z_t), \tag{18}$$

where η controls the constraint strength. The gradient $\nabla_{z_t} \mathcal{L}_{\text{pbr}}$ is computed via backpropagation through the material decoder D_t, analogous to the voxel-supervised geometry alignment. Details are shown in supplementary material..

4 Experiment

4.1 Experimental Setups

Datasets. We introduce **HiFi200 Benchmark**, a comprehensive benchmark comprising 200 high-quality editing samples with balanced coverage across all sorts of editing categories: action change, object addition, object removal, object replacement, and style/texture change (40 samples each). Each sample includes an editing instruction, source 3D shape, source image condition, and target image condition. More detailed results are illustrated in supplementary material.

Metrics. Geometry Metrics: CD($\mathcal{R}_i$) and CD($\mathcal{R}_e$) measure Chamfer Distance in invariant and editable regions respectively, quantifying geometric preservation and edit accuracy without our voxel-space objective optimization; F-Score evaluates overall surface reconstruction quality at threshold 0.01.

Appearance Metrics: CLIP-T measures text-image semantic alignment for instruction adherence; DINO-I captures visual similarity to target reference; LPIPS quantifies perceptual image distance; FID assesses generation quality distribution matching.

Baselines. We compare against several state-of-the-art 3D editing methods: **MVEdit** Chen et al. (2024a) is a representative multi-view consistent editing baseline that applies 2D edits across multiple views and reconstructs 3D geometry; **TRELLIS2** Xiang et al. (2025) is our base generation model with structured latent representation; **Nano3D** Ye et al. (2026) and **Easy3E** Hu et al. (2026) are flow-based editing methods similar to our approach but without region-aware dynamics or dual-level preservation.

4.2 Qualitative Comparison

We present qualitative comparisons across diverse editing scenarios. Our method consistently produces high-quality results superior to baselines in both geometric structure and material appearance.

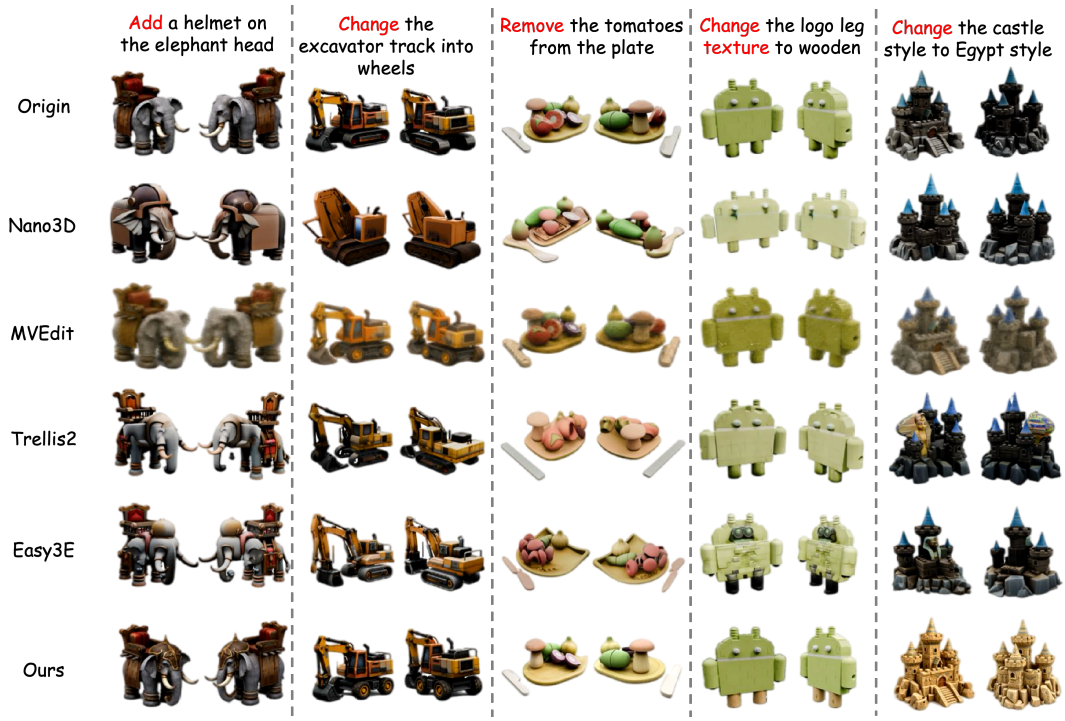

Figure 4: **Qualitative results.** Different types of editing are illustrated in this figure and out method performs the best among all baseline methods.

Table 2: Ablation study on flow dynamics components.

Method	v_{stable}	G_{voxel}	CD($\mathcal{R}_i$)↓
Base	×	×	0.0161
w/o voxel	✓	×	0.0139
w/o latent	×	✓	0.0145
Ours (Full)	✓	✓	**0.0123**

General Editing (Addition, Removement, Replacement). As shown in columns 1-3 of Figure 4, our method preserves the original 3D asset structure perfectly in the invariant region while accurately processing the edited region without over-generation or under-generation artifacts, maintaining identity consistency with the source model. In contrast, MVEdit achieves relatively complete geometry generation but suffers from inevitable detail loss. The other compared methods fail to maintain structural integrity under complex edits, exhibiting artifacts and over-generation in the edited regions.

Texture Editing. We categorize texture editing into local region appearance changes and global style transfer. As shown in columns 4-5 of Figure 4, baseline methods fail to correctly respond to target instructions for desired texture editing effects. Most methods preserve basic geometric structure but cannot control texture generation—even Nano3D, which builds upon TRELLIS with dedicated texture processing, fails to achieve proper texture editing. Our method realizes nearly lossless texture editing with accurate material mapping and view-consistent appearance.

4.3 Quantitative Results

Table 1 presents the quantitative comparison with state-of-the-art methods. Our method achieves the best performance across all geometry and appearance metrics. For geometry preservation, it obtains the lowest CD of 0.0123 on the invariant region and 0.0109 on the editable region, together with the highest F-Score of 0.756, demonstrating improved structural preservation while supporting effec-

tive geometric edits. For appearance quality, our method achieves the highest CLIP-T (0.312) and DINO-I (0.734), as well as the lowest LPIPS (0.245) and FID (28.1), indicating stronger instruction alignment and visual consistency. Moreover, our method requires only 16.5 seconds per edit, achieving the best efficiency among all compared methods. These results demonstrate that HiFi3D improves both geometric fidelity and appearance quality while maintaining rapid inference.

4.4 User Study

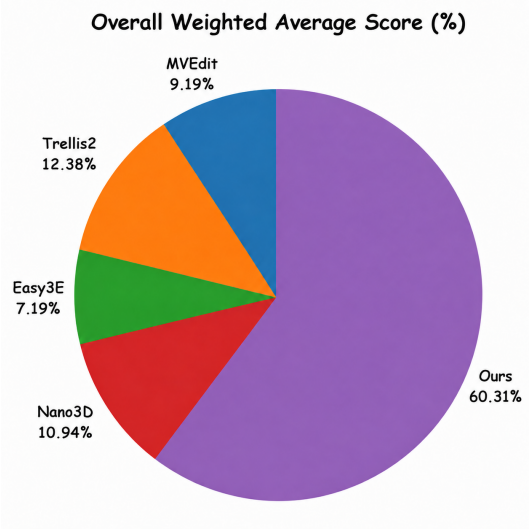

Figure 5: **User study results.** Overall user preference averaged across four evaluation criteria.

We conducted a comprehensive user study with 40 participants to evaluate editing quality and usability. Participants were presented with the editing prompt together with videos rendered from the source 3D asset and the edited results produced by our method and four competing methods. They were asked to select the best result according to four criteria: prompt alignment, visual quality, shape preservation, and material realism.

As shown in Figure 5, our method obtains an overall preference of 60.31%, substantially higher than the competing methods. The detailed per-criterion results are provided in the supplementary material.

4.5 Ablation Studies

Flow Dynamics Components. We conduct ablation studies on the two components in Eq. 15. Specifically, we compare the base flow decomposition update without auxiliary guidance against variants using latent supervision (v_{stable}) or voxel supervision (G_{voxel}), as well as the complete model using both components.



Figure 6: **Comparison of flow dynamics components.** We separately disable latent guidance v_{stable} and voxel guidance G_{voxel} to analyze their contributions.

As shown in Figure 6, either supervision component improves the geometry to some extent, but geometric deviations and undesired deformations remain. Combining both components produces more stable geometry and better preserves the intended structure. The quantitative results in Table 2 show the same trend.

Material Editing Module. For the material editing module, we compare our method with the texture optimization strategy used in the base TRELLIS2 model. The corresponding results are provided in the supplementary material. Compared with the baseline, our method produces more view-consistent appearances and better preserves inter-view consistency during material editing, while responding more accurately to the target editing instructions.

5 Conclusion

We presented HiFi3D, a training-free framework for high-fidelity 3D asset editing that addresses fundamental challenges in sparse voxel manipulation. Our method maintains geometric fidelity in invariant regions while enabling precise edits. Extensive experiments on the HiFi200 benchmark demonstrate that HiFi3D outperforms state-of-the-art baselines across diverse editing tasks, achieving superior visual quality and instruction adherence while maintaining practical efficiency and establishing a foundation for representation-aware 3D editing.

REFERENCES

Roi Bar-On, Dana Cohen-Bar, and Daniel Cohen-Or. EditP23: 3d editing via propagation of image prompts to multi-view, 2025.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18392–18402, 2023.

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, 2024a.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the International Conference on Machine Learning*, 2024.

William Gao, Noam Aigerman, Thibault Groueix, Vladimir G. Kim, and Rana Hanocka. TextDeformer: Geometry manipulation using text guidance. In *Proceedings of the ACM SIGGRAPH Conference*, 2023.

Ayaan Haque, Matthew Tancik, Alexei A. Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-NeRF2NeRF: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *Proceedings of the International Conference on Learning Representations*, 2023.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3d. In *Proceedings of the International Conference on Learning Representations*, 2023.

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

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.

Hyunwoo Kim, Itai Lang, Noam Aigerman, Thibault Groueix, Vladimir G. Kim, and Rana Hanocka. MeshUp: Multi-target mesh deformation via blended score distillation, 2024.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. FlowEdit: Inversion-free text-based editing using pre-trained flow models, 2024.

Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Gregory Shakhnarovich, and Sai Bi. Instant3D: Fast text-to-3d with sparse-view generation and large reconstruction model. In *Proceedings of the International Conference on Learning Representations*, 2023.

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

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

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 *Advances in Neural Information Processing Systems*, 2024a.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9298–9309, 2023a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *Proceedings of the International Conference on Learning Representations*, 2023b.

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. In *Proceedings of the International Conference on Learning Representations*, 2024b.

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. Wonder3D: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9970–9980, 2024.

Lihua Lu, Ruyang Li, Xiaohui Zhang, Hui Wei, Guoguang Du, and Binqiang Wang. Advances in text-guided 3d editing: A survey. *Artificial Intelligence Review*, 57:321, 2024.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *Proceedings of the International Conference on Learning Representations*, 2022.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision*, 2020.

Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-E: A system for generating 3d point clouds from complex prompts, 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

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

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22500–22510, 2023.

Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: A hybrid representation for high-resolution 3d shape synthesis. In *Advances in Neural Information Processing Systems*, 2021.

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

Junjie Wang, Jiemin Fang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. GaussianEditor: Editing 3d gaussians delicately with text instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

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

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, 2024.

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

Junliang Ye, Shenghao Xie, Zhengyi Wang, Ruowen Zhao, Hongyu Yan, Wenqiang Zu, Lei Ma, and Jun Zhu. NANO3D: A training-free approach for efficient 3d editing without masks. In *Proceedings of the International Conference on Learning Representations*, 2026.

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, et al. Hunyuan3D 2.0: Scaling diffusion models for high resolution textured 3d assets generation, 2025.

Zhenglin Zhou, Fan Ma, Chengzhuo Gui, Xiaobo Xia, Hehe Fan, Yi Yang, and Tat-Seng Chua. AnchorFlow: Training-free 3d editing via latent anchor-aligned flows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Jingyu Zhuang, Chen Wang, Lingjie Liu, Liang Lin, and Guanbin Li. DreamEditor: Text-driven 3d scene editing with neural fields. In *Proceedings of the ACM SIGGRAPH Asia Conference*, 2023.

A AI USE DISCLOSURE

Generative AI tools were used to assist with manuscript editing and readability, and to provide feedback on the presentation and description of the research methodology. All AI-assisted suggestions were reviewed and verified by the authors, and all methodological decisions, experiments, and reported results were conducted and validated by the authors. The authors take full responsibility for the final content of this work.

B DETAILS OF GEOMETRY-CONDITIONED MATERIAL EDITING

B.1 THEORETICAL FOUNDATION OF MATERIAL VELOCITY ESTIMATION

Given a material latent state $z_t \in \mathbb{R}^{d_{\text{mat}}}$ at time $t \in [0, 1]$ in the flow matching framework, our goal is to estimate the edit velocity field that transforms source material appearance to target appearance while maintaining physical plausibility. The ideal material edit velocity is defined as:

$$v^{\text{mat}}_{\text{edit}}(z_t, t) = \mathbb{E}_{\xi\sim\mathcal{N}(0,I)}\big[v_\theta(z_t + t\xi, t \mid \mathcal{I}^{\text{tgt}}) - v_\theta(z_t + t\xi, t \mid \mathcal{I}^{\text{src}})\big] \tag{19}$$

where v_θ is the pretrained flow velocity predictor, and $\mathcal{I}^{\text{src}}, \mathcal{I}^{\text{tgt}}$ are source and target conditioning images.

Theorem 1 (Material Velocity Variance). The variance of single-sample material velocity estimation is bounded by:

$$\text{Var}[v^{\text{mat}}_\theta(\xi)] \geq \text{Var}[v^{\text{shape}}_\theta(\xi)] \cdot (1 + \kappa_{\text{lighting}}) \tag{20}$$

where $\kappa_{\text{lighting}} > 0$ represents the lighting-texture ambiguity factor.

Consider the material velocity predictor v^{mat}_θ. Due to the lighting-texture decomposition ambiguity, a single observation under specific lighting can be explained by multiple (c, m, r) combinations. Formally, let $\mathcal{A}(c, m, r, \ell)$ be the appearance function under lighting ℓ. The ambiguity set is:

$$\mathcal{S}_{\text{equiv}}(I) = \{(c, m, r) : \mathcal{A}(c, m, r, \ell_{\text{obs}}) = I\} \tag{21}$$

This ambiguity increases the prediction variance. The neural network v_θ implicitly marginalizes over possible interpretations, leading to:

$$\mathbb{E}_\xi[\|v^{\text{mat}}_\theta(\xi) - \mathbb{E}[v^{\text{mat}}_\theta]\|^2] = \sigma^2_{\text{base}} + \sigma^2_{\text{ambiguity}} \tag{22}$$

where $\sigma^2_{\text{ambiguity}}$ is the additional variance from lighting-texture ambiguity, absent in geometric prediction. □

We approximate the expectation using K independent samples:

$$\hat{v}^{\text{mat}}_{\text{edit}}(z_t, t) = \frac{1}{K}\sum_{k=1}^{K}\big[v_\theta(z_t + t\xi_k, t \mid \mathcal{I}^{\text{tgt}}) - v_\theta(z_t + t\xi_k, t \mid \mathcal{I}^{\text{src}})\big] \tag{23}$$

Proposition 1 (Estimation Error Bound). The mean squared error of the K-sample estimator is:

$$\mathbb{E}[\|\hat{v}^{\text{mat}}_{\text{edit}} - v^{\text{mat}}_{\text{edit}}\|^2] = \frac{\sigma^2_{\text{total}}}{K} \tag{24}$$

where $\sigma^2_{\text{total}} = \text{Var}[v^{\text{mat}}_\theta(\xi_1)]$.

Given computational budget B and per-sample cost C, we minimize the error-cost tradeoff:

$$K^* = \arg\min_K \left\{\frac{\sigma^2_{\text{total}}}{K} + \lambda \cdot KC\right\} \tag{25}$$

Solving $\frac{d}{dK}\left(\frac{\sigma^2_{\text{total}}}{K} + \lambda KC\right) = 0$ gives:

$$K^* = \sqrt{\frac{\sigma^2_{\text{total}}}{\lambda C}} \tag{26}$$

Empirically, we find $K_{\text{mat}} = 3$ achieves satisfactory error reduction (reducing variance by factor of 3) while maintaining real-time performance.

B.2 PBR Consistency Constraints: Physical Foundations

The material feature space is $\mathcal{M} = \mathbb{R}^3 \times [0,1]^3$, where $(c, m, r, \alpha) \in \mathcal{M}$ represents base color, metallic, roughness, and opacity. However, not all points in $\mathcal{M}$ correspond to physically realizable materials. We define the physically plausible material manifold as:

$$\mathcal{M}_{\text{phys}} = \{(c, m, r, \alpha) \in \mathcal{M} : \Phi_{\text{pbr}}(c, m, r, \alpha) = \text{True}\} \tag{27}$$

where Φ_{pbr} encodes physical constraints: (1) *Metallic-Roughness Correlation*—high metallic materials ($m \to 1$) typically have low roughness ($r \to 0$) due to smooth surface finish; (2) *Diffuse-Roughness Coupling*—high roughness ($r \to 1$) should exhibit smooth color variations (diffuse scattering); (3) *Value Range Consistency*—color values must remain in valid range.

Metallic-Roughness Correlation. In real-world materials, high metallic values (e.g., polished metals) are typically associated with low roughness due to the manufacturing process. The combination $(m \approx 1, r \approx 1)$ is physically implausible—it would represent a "rough mirror," which doesn't occur naturally. We define the violation function:

$$\mathcal{L}_{\text{metal}}(z_t) = \mathbb{E}_i \left[\Psi(m_i, r_i; \theta_m, \theta_r)\right] \tag{28}$$

where $\Psi(m, r; \theta_m, \theta_r) = \max(0, m - \theta_m) \cdot \max(0, r - \theta_r)$ is the penalty function. This function has three key properties: (i) zero penalty in valid region (if $m \le \theta_m$ or $r \le \theta_r$, then $\Psi = 0$); (ii) quadratic growth in violation ($\Psi \approx (m - \theta_m)(r - \theta_r)$ for $m > \theta_m, r > \theta_r$); (iii) piecewise smoothness, enabling gradient-based optimization. The gradient with respect to latent z_t is:

$$\nabla_{z_t} \mathcal{L}_{\text{metal}} = \mathbb{E}_i \left[\frac{\partial \Psi}{\partial m_i}\frac{\partial m_i}{\partial z_t} + \frac{\partial \Psi}{\partial r_i}\frac{\partial r_i}{\partial z_t}\right] \tag{29}$$

where

$$\frac{\partial \Psi}{\partial m} = \begin{cases} \max(0, r - \theta_r) & \text{if } m > \theta_m \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

$$\frac{\partial \Psi}{\partial r} = \begin{cases} \max(0, m - \theta_m) & \text{if } r > \theta_r \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

We set $\theta_m = 0.8, \theta_r = 0.5$ based on empirical analysis of real-world material databases (e.g., Adobe Stock Materials, Polyhaven).

Smoothness-Roughness Coupling. Materials with high roughness exhibit strong diffuse scattering, which tends to smooth out local color variations. Sharp color gradients on rough surfaces are physically implausible (except at material boundaries). The smoothness constraint is:

$$\mathcal{L}_{\text{smooth}}(z_t) = \mathbb{E}_i \left[r_i \cdot \|\nabla c_i\|^2\right] \tag{32}$$

This term penalizes high-frequency color variations weighted by roughness. As $r_i \to 1$, color gradients are strongly penalized; as $r_i \to 0$, sharp color features are permitted (e.g., specular highlights). Using the product rule:

$$\nabla_{z_t} \mathcal{L}_{\text{smooth}} = \mathbb{E}_i \left[\|\nabla c_i\|^2 \frac{\partial r_i}{\partial z_t} + r_i \frac{\partial \|\nabla c_i\|^2}{\partial z_t}\right] \tag{33}$$

For the second term, expanding $\|\nabla c_i\|^2 = \sum_j (\partial c_i / \partial x_j)^2$:

$$\frac{\partial \|\nabla c_i\|^2}{\partial z_t} = 2 \sum_j \frac{\partial c_i}{\partial x_j} \frac{\partial^2 c_i}{\partial x_j \partial z_t} \tag{34}$$

In practice, we approximate spatial gradients using finite differences on the decoded O-Voxel grid.

Value Range Constraint. Base color $c_i \in \mathbb{R}^3$ represents albedo, which must satisfy $c_i \in [0,1]^3$ for physical rendering:

$$\mathcal{L}_{\text{range}}(z_t) = \mathbb{E}_i \left[\max(0, \|c_i\|_\infty - 1)\right] \tag{35}$$

where $\|c_i\|_\infty = \max\{|c_i^r|, |c_i^g|, |c_i^b|\}$. The gradient is:

$$\frac{\partial \mathcal{L}_{\text{range}}}{\partial z_t} = \mathbb{E}_i \left[1_{\|c_i\|_\infty > 1} \cdot \frac{\partial c_i^{\max}}{\partial z_t}\right] \tag{36}$$

where $c_i^{\max}$ is the channel achieving $\|c_i\|_\infty$.

The total PBR consistency loss is:

$$\mathcal{L}_{\text{pbr}}(z_t) = \lambda_m \mathcal{L}_{\text{metal}}(z_t) + \lambda_r \mathcal{L}_{\text{smooth}}(z_t) + \lambda_c \mathcal{L}_{\text{range}}(z_t) \tag{37}$$

We set $\lambda_m = 1.0$ (baseline unit weight for primary constraint), $\lambda_r = 0.5$ (lower weight since smoothness is a soft preference), and $\lambda_c = 2.0$ (higher weight to strictly enforce valid value ranges).

B.3 PBR-Regularized Flow Dynamics

The unconstrained material flow follows $\frac{dz_t}{dt} = v_{\text{edit}}^{\text{mat}}(z_t, t)$. To enforce PBR constraints, we introduce a regularization term derived from the constraint gradient:

$$\frac{dz_t}{dt} = v_{\text{edit}}^{\text{mat}}(z_t, t) - \eta \nabla_{z_t} \mathcal{L}_{\text{pbr}}(z_t) \tag{38}$$

The second term is a projection onto the physical manifold $\mathcal{M}_{\text{phys}}$. At each time step, the flow is adjusted to reduce the PBR violation.

The PBR loss depends on decoded material parameters $(c_i, m_i, r_i, \alpha_i) = D_t(z_t)$, where D_t is the time-dependent material decoder. By the chain rule:

$$\nabla_{z_t} \mathcal{L}_{\text{pbr}}(z_t) = \sum_i \left[\frac{\partial \mathcal{L}_{\text{pbr}}}{\partial c_i} \frac{\partial c_i}{\partial z_t} + \frac{\partial \mathcal{L}_{\text{pbr}}}{\partial m_i} \frac{\partial m_i}{\partial z_t} + \frac{\partial \mathcal{L}_{\text{pbr}}}{\partial r_i} \frac{\partial r_i}{\partial z_t} \right] \tag{39}$$

Define the decoder Jacobian $J_D = \frac{\partial(c,m,r,\alpha)}{\partial z_t} \in \mathbb{R}^{(3N+3N)\times d_{\text{mat}}}$ where N is the number of O-Voxels. Then:

$$\nabla_{z_t} \mathcal{L}_{\text{pbr}} = J_D^T \nabla_{(c,m,r,\alpha)} \mathcal{L}_{\text{pbr}} \tag{40}$$

We compute this using automatic differentiation (backpropagation) through the decoder network D_t.

Theorem 2 (Flow Convergence to Physical Manifold). Under Lipschitz continuity assumptions on $v_{\text{edit}}^{\text{mat}}$ and $\nabla_{z_t} \mathcal{L}_{\text{pbr}}$, the regularized flow converges to a trajectory that satisfies:

$$\lim_{T \to \infty} \mathcal{L}_{\text{pbr}}(z_T) \leq \epsilon \tag{41}$$

for sufficiently large regularization strength η.

Consider the Lyapunov function $V(z_t) = \mathcal{L}_{\text{pbr}}(z_t)$. The time derivative along the flow is:

$$\frac{dV}{dt} = \nabla_{z_t} \mathcal{L}_{\text{pbr}}^T \frac{dz_t}{dt} = \nabla_{z_t} \mathcal{L}_{\text{pbr}}^T v_{\text{edit}}^{\text{mat}} - \eta \|\nabla_{z_t} \mathcal{L}_{\text{pbr}}\|^2 \tag{42}$$

If $|\nabla_{z_t} \mathcal{L}_{\text{pbr}}^T v_{\text{edit}}^{\text{mat}}| \leq L \|\nabla_{z_t} \mathcal{L}_{\text{pbr}}\|$ (Lipschitz bound), then:

$$\frac{dV}{dt} \leq L \|\nabla_{z_t} \mathcal{L}_{\text{pbr}}\| - \eta \|\nabla_{z_t} \mathcal{L}_{\text{pbr}}\|^2 \tag{43}$$

For $\eta > L$, this is negative whenever $\mathcal{L}_{\text{pbr}} > \epsilon$ for some $\epsilon > 0$, proving convergence. □

In practice, we discretize the flow using Euler integration with step size Δt:

$$z_{t+\Delta t} = z_t + \Delta t \left(v_{\text{edit}}^{\text{mat}}(z_t, t) - \eta \nabla_{z_t} \mathcal{L}_{\text{pbr}}(z_t) \right) \tag{44}$$

For numerical stability, we require $\Delta t \leq \frac{2}{\|v_{\text{edit}}^{\text{mat}}\| + \eta \|\nabla_{z_t} \mathcal{L}_{\text{pbr}}\|}$. Empirically, we use $\Delta t = 0.02$ (50 steps from $t = 0$ to $t = 1$) and $\eta = 0.1$.

B.4 Theoretical Guarantees and Computational Complexity

Theorem 3 (Orthogonality of Edit and Constraint Gradients). Under the assumption that the pretrained model v_θ implicitly captures physical priors, the edit velocity $v_{\text{edit}}^{\text{mat}}$ and constraint gradient $\nabla_{z_t} \mathcal{L}_{\text{pbr}}$ are approximately orthogonal in expectation:

$$\mathbb{E}[\langle v_{\text{edit}}^{\text{mat}}, \nabla_{z_t} \mathcal{L}_{\text{pbr}} \rangle] \approx 0 \tag{45}$$

Table 3: User study results. Preference percentage across 40 participants. Detailed breakdown in supplementary material.

Method	Prompt Align.↑	Visual Quality↑	Shape Preserve↑	Material Real.↑	Overall↑
MVEdit	9.25%	8.75%	11.50%	7.25%	9.19%
Trellis2	13.75%	14.25%	9.50%	12.00%	12.38%
Easy3E	6.50%	8.00%	5.75%	8.50%	7.19%
Nano3D	9.00%	11.25%	13.75%	9.75%	10.94%
Ours	**61.50%**	**57.75%**	**59.50%**	**62.50%**	**60.31%**

This implies the regularization primarily corrects "off-manifold" deviations without significantly opposing the edit direction, preserving edit quality.

The regularized flow update requires: (1) velocity estimation—K forward passes through v_θ with cost $O(K \cdot C_{\text{model}})$; (2) decoder forward pass $D_t(z_t)$ with cost $O(C_{\text{decoder}})$; (3) PBR loss computation $\mathcal{L}_{\text{pbr}}$ on N voxels with cost $O(N)$; (4) gradient backpropagation through D_t with cost $O(C_{\text{decoder}})$. The total per-step complexity is $O(K \cdot C_{\text{model}} + C_{\text{decoder}} + N)$. Compared to unconstrained flow ($O(K \cdot C_{\text{model}})$) and post-hoc clamping ($O(K \cdot C_{\text{model}} + N)$), our method adds minimal overhead ($C_{\text{decoder}} \ll C_{\text{model}}$) while providing strong physical guarantees.

C Dataset description

We construct **HiFi200**, a high-standard evaluation benchmark for structured 3D editing, comprising 200 carefully curated editing samples. Compared to existing editing benchmarks, we establish higher standards in data quality, category balance, and evaluation comprehensiveness.

Source Asset Collection. We collect 3D assets from multiple carefully selected high-quality data sources:

- **Objaverse-XL**: Filtered using a strict aesthetic score threshold ≥ 7.8 to ensure geometric integrity and visual quality
- **ABO (Amazon Berkeley Objects)**: Industrial-grade product scans providing precise geometry and textures
- **TRELLIS-500K**: Large-scale high-quality generative assets covering diverse shape spaces
- **Sketchfab**: Commercial-grade 3D objects modeled by professional artists
- **NPHM**: High-precision scans of real human heads preserving fine-grained features
- **THuman2.1**: Diverse clothed human data encompassing complex garment topology

These diverse assets cover rigid objects (tools, furniture, products), deformable characters (animals, figures), and human models with complex topology, ensuring broad applicability of the evaluation.

Edit Categories. Drawing from the automatic annotation approach of 3D-Alpaca, we employ Qwen2.5-VL for scene classification and ensure annotation accuracy through manual review. Our benchmark spans five core editing types, with each category precisely assigned 40 samples to achieve strict category balance:

1. **Action Change**: Pose adjustments, e.g., "sitting" → "standing", "static" → "running"
2. **Object Addition**: Adding components, e.g., adding hats, glasses, or accessories
3. **Object Removal**: Removing parts, e.g., eliminating background objects, simplifying structures
4. **Object Replacement**: Replacing components, e.g., changing hairstyles, materials, clothing
5. **Style/Texture Change**: Appearance editing, e.g., material remapping, color theme transformation

Each sample contains a complete editing quadruple: (1) natural language editing instruction, (2) source 3D model (O-Voxel format), (3) source image condition, (4) target image condition.

Edit Instruction Generation. We use **Claude 3.5 Sonnet** to generate semantically rich and diverse editing instructions, which outperforms Gemini 2.5 Pro in 3D understanding and instruction generation quality. All instructions undergo two-stage validation: (1) automatic verification of semantic consistency and feasibility, (2) manual expert review to eliminate ambiguity or physically infeasible editing requirements. Instruction length is controlled between 10-30 words to ensure clarity and executability.

D Complete User Study

Study Design and Participants. We conduct a user study with 40 participants, including 15 graduate students in graphics/vision, 12 professional 3D artists, 8 researchers with publications in 3D generation/editing, and 5 industry practitioners. All participants have at least 2 years of experience with 3D content and complete a 10-minute training session before evaluation.

We evaluate five dimensions: (1) **Prompt Alignment**—semantic match with text instruction; (2) **Visual Quality**—aesthetic appeal and rendering quality; (3) **Shape Preservation**—maintaining original geometry in non-edited regions; (4) **Material Realism**—physical plausibility of PBR parameters; (5) **Overall Preference**—holistic assessment for practical use.

Evaluation Protocol. We randomly select 50 tasks from OVoxel-Edit-200, balanced across five editing categories. For each task, participants view: (1) source model from four viewpoints; (2) editing instruction; (3) target image; (4) five edited results (ours + four baselines: MVEdit, Trellis2, Easy3E, Nano3D), anonymized and randomly ordered. Participants interact with 360° renderings and select their preferred method for each dimension. Each participant evaluates all 50 tasks, yielding 2000 total judgments (40 participants $\times$ 50 tasks) per dimension.

Results and Analysis. Table 3 shows our method achieves 60.31% overall preference, significantly outperforming all baselines. We observe particularly strong performance in Material Realism (62.50%) and Prompt Alignment (61.50%). The high preference in Shape Preservation (59.50%) validates our masked velocity estimation strategy. Pairwise Wilcoxon signed-rank tests confirm significant superiority over all baselines ($p < 0.001$) in every dimension. Inter-rater agreement (Fleiss' kappa $\kappa = 0.68$) indicates substantial agreement, validating evaluation reliability.

Key Findings. Participants praised our method's material consistency and physically plausible PBR parameters. Baseline methods often produce "plasticky" or "waxy" appearance due to unconstrained material flow, while our PBR-regularized approach maintains realistic transitions. Participants also noted cleaner edit boundaries compared to baselines exhibiting "bleeding" artifacts. Identified limitations include extreme pose changes, fine-grained details beyond O-Voxel resolution, and global illumination adjustments.

E Material Editing module ablation details

We compare our material editing module against the texture optimization approach from the base TRELLIS2 editing module. We introduce three specialized metrics: (1) **View Consistency Score (VCS)** measures cross-view appearance consistency via average LPIPS distance between rendered view pairs; (2) **Realism Score** evaluates material plausibility through human assessment (1-5 scale) under novel lighting; (3) **CLIP-Sim** quantifies semantic alignment between edited results and target instructions via CLIP embedding similarity.

As shown in Fig7 and table4, our method addresses a critical limitation of TRELLIS2's optimization: the tendency to bake lighting information into albedo channels, conflating intrinsic material color with extrinsic illumination. Our $\mathcal{L}_{\text{pbr}}$ constraint ensures physically plausible material parameters, maintaining appearance correctness across different lighting and viewing angles.

Furthermore, our masked velocity estimation $v_{\text{edit}}^{\text{mat}}(z_t, t)$ provides precise instruction alignment through conditional flow matching. While TRELLIS2 often produces partial matches to editing



Figure 7: **Ablation on Material Editing module.** We compare our module with basic Trellis2 editing module. Results shown that our method performs much better on texture while maintaining precise structure.

Table 4: Material editing comparison against TRELLIS2 texture optimization. Metrics averaged over 30 material editing tasks.

Method	VCS↓	Realism↑	CLIP-Sim↑
TRELLIS2	0.087	2.8	0.61
Ours	**0.023**	**4.2**	**0.78**

instructions (e.g., introducing marble patterns without adjusting roughness), our method correctly modifies all relevant material aspects. Quantitatively, we achieve View Consistency Score of 0.023 (vs. TRELLIS2's 0.087) and CLIP semantic similarity of 0.78 (vs. 0.61).