UMAS3D: UNIFIED MASK-FREE 3D EDITING WITH REGION-ADAPTIVE SOURCE PRESERVATION

Anonymous authors
Paper under double-blind review

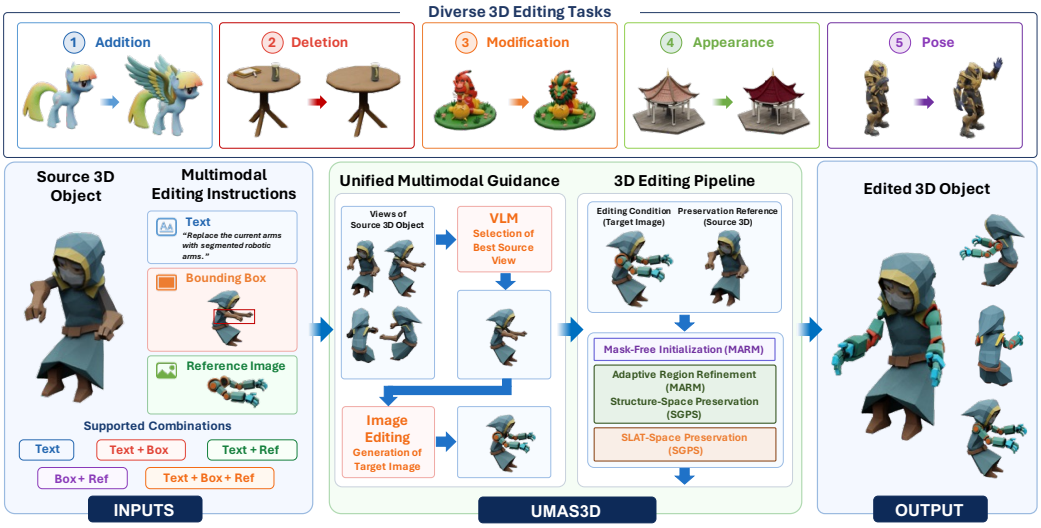

Figure 1: **Overview of UMAS3D.** A training-free framework that unifies multimodal conditions and five 3D editing tasks through adaptive region refinement and selective source preservation.

ABSTRACT

3D editing aims to modify an existing 3D asset according to user intent while preserving geometry and appearance outside the intended edit region. Existing methods often rely on task-specific training, source-3D inversion, or external 3D masks, and lack a unified framework for diverse editing conditions and tasks. We propose UMAS3D, a unified 3D editing framework that is training-free, inversion-free, and free from external 3D masks. UMAS3D converts text, bounding boxes, reference images, and their combinations into common target-view guidance, after which addition, deletion, modification, appearance, and pose editing share the same downstream 3D generation process. Our key insight is to couple online edit-region estimation with region-aware source preservation. Mask-Free Adaptive Region Modeling (MARM) initializes an Edit/Keep prior from source-target visual differences and progressively refines it as the target structure evolves, while Source-Guided Preservation Steering (SGPS) constrains the current Keep-region states to retain source geometry and appearance without suppressing target-driven changes. Experiments show that UMAS3D supports addition, deletion, modification, appearance, and pose editing while effectively preserving unedited source content. These advances make UMAS3D a unified framework for flexible and source-preserving 3D editing across diverse editing conditions. Project Page: `https://paperprojectpage.github.io/UMAS3D`

1 INTRODUCTION

Recent advances (Xiang et al., 2025; Zhao et al., 2025) in 3D generative models and representation techniques have substantially improved 3D asset quality, enabling applications in digital en-

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.

tertainment, AR/VR, and interactive content creation. Building on this progress, controllable 3D editing (Barda et al., 2025; Bar-On et al., 2025; Xia et al., 2025b) further enhances content-creation flexibility by enabling users to modify and customize existing 3D assets under specified editing conditions. Given a source 3D asset and an editing condition, the goal is to realize the requested modification while preserving the geometry, appearance, and identity of unaffected regions.

Existing 3D editing methods can be broadly categorized into training-based and training-free approaches. Training-based methods learn editing transformations from paired 3D data through additional model optimization (Gat et al., 2026; Ma et al., 2025a), whereas training-free methods reuse pretrained 3D generative priors through mechanisms such as source-3D inversion, source–target trajectory manipulation, or latent-space operations (Li et al., 2026; Chi et al., 2026; Zhou et al., 2026). Existing methods also differ in the user-facing editing conditions and editing tasks they support (Barda et al., 2025; Bar-On et al., 2025; Xia et al., 2025b). For localized editing, some approaches additionally rely on a 3D editing mask either provided as input or estimated before generation to specify where the edit is applied (Barda et al., 2025; Li et al., 2026; Nie et al., 2026).

These differences motivate three requirements for a more general 3D editor. **First**, editing intent may be expressed through text, bounding boxes, reference images, or their combinations; mapping these inputs to a common interface allows them to share the same downstream editing process. **Second**, a training-free formulation avoids dependence on paired 3D editing data and editing-specific model optimization, enabling the same pretrained generative prior to be reused across tasks. **Third**, fixed edit regions computed before generation cannot reflect subsequent changes in the target geometry. As the target evolves, such a region may exclude areas that later require editing or include source content that should remain unchanged, resulting in incomplete edits or unintended modifications. This motivates adjusting edit localization during generation to keep it aligned with the evolving target. The resulting challenge is to unify diverse conditions and tasks in a training-free framework while supporting adaptive edit localization.

To address this challenge, we propose **UMAS3D**, a unified 3D editing framework that operates without additional model training, source-3D inversion, or externally supplied 3D masks. As illustrated in Figure 1, **UMAS3D** achieves two levels of unification: text, bounding boxes, reference images, and their combinations are converted into common target-view guidance, while addition, deletion, modification, appearance, and pose editing share the same downstream 3D editing pipeline. Its core editing mechanism consists of two complementary modules. **Mask-Free Adaptive Region Modeling (MARM)** constructs a coarse Edit/Keep prior from multi-view source–target visual differences and adaptively refines the regional assignments as the target structure evolves. **Source-Guided Preservation Steering (SGPS)** then applies representation-specific source constraints only within the current Keep regions while leaving Edit regions target-driven. Thus, MARM determines where source content should be retained, whereas SGPS controls how it is preserved, allowing source preservation to follow the evolving target while keeping source steering spatially localized.

We evaluate UMAS3D on 150 source objects from Pxform (Weng et al., 2026) and Omni3DEdit (Fan et al., 2026), covering five editing tasks. Under a controlled fixed-target-view protocol, UMAS3D achieves the best reported results across all evaluated metrics among the compared baselines. Qualitative results under five constructed multimodal input settings and a user study provide additional evidence of the framework's flexibility and perceptual quality.

Overall, our contributions can be summarized as follows:

- We present UMAS3D, a unified framework that supports heterogeneous conditions and five editing tasks through a shared 3D editing pipeline, without additional training, source-3D inversion, or externally supplied 3D masks.
- We introduce Mask-Free Adaptive Region Modeling (MARM) and Source-Guided Preservation Steering (SGPS) for adaptive source supervision. MARM updates an explicit Edit/Keep partition from source–target evidence, while SGPS preserves source content only within the current Keep regions.
- Experiments on 150 objects show that UMAS3D achieves the best reported results across all evaluated metrics among the compared baselines. Ablations, multimodal results, and a user study further support its effectiveness.

2 Related Work

2.1 3D Editing Paradigms

3D editing methods span several technical paradigms. A major line of work leverages pre-trained 2D diffusion priors to optimize 3D representations through SDS or related objectives (Chen et al., 2024e; Wang et al., 2024; Palandra et al., 2024; Kim et al., 2025b; Huang et al., 2025; Sella et al., 2023; Zhuang et al., 2023; Koo et al., 2024; Li et al., 2024). Another class combines semantic localization, image-space editing signals, or generative trajectories with NeRF or 3D Gaussian representations (Sun et al., 2024; Lee et al., 2025a; Luo et al., 2024; Hong et al., 2025; Haque et al., 2023; Bao et al., 2023; Song et al., 2023). Multi-view and image-space methods propagate editing signals across views and subsequently reconstruct or optimize the underlying 3D representation (Chen et al., 2024c; Wu et al., 2024a; Chen et al., 2024a; Qi et al., 2024; Chen et al., 2024b; Erkoç et al., 2025; Zhang et al., 2026; Wu et al., 2026). Recent work further explores feed-forward editing directly in native 3D latent or voxel representations (Chen et al., 2024d; Weng et al., 2026; Cai et al., 2025; Hu et al., 2026). Spatial cues such as masks, local regions, and control points are also widely used for local 3D editing (Gao et al., 2025; Shen et al., 2024; Qu et al., 2025; Kim et al., 2025a).

2.2 Training-Free Native 3D Editing

With the emergence of pre-trained 3D generative models, training-free methods leverage frozen 3D generative priors for editing (Ye et al., 2026; Li et al., 2026; Chi et al., 2026; Zhou et al., 2026; Liu et al., 2026; Nie et al., 2026; Parelli et al., 2026). Some preserve unedited content through source-3D inversion. VoxHammer replaces preserved-region features with inverted latents and cached attention states, while Vinedresser3D performs inversion-based rectified-flow inpainting (Li et al., 2026; Chi et al., 2026). Others manipulate source-to-target generation in native voxel or latent spaces through flow trajectories, latent anchors, velocity fields, or internal features (Ye et al., 2026; Zhou et al., 2026; Liu et al., 2026; Nie et al., 2026; Parelli et al., 2026). **UMAS3D** instead avoids source-3D inversion and updates Edit/Keep preservation support online within the same Stage-1 trajectory as the target structure evolves.

2.3 Multimodal Guidance and 3D Edit Localization

Visual editing interfaces span text instructions, reference images and multimodal instructions (Yu et al., 2025; Xia et al., 2025a; Xiao et al., 2025; Chen et al., 2025; He et al., 2025; Tian et al., 2025). Such controls have also been explored in 3D editing through text-image prompts, bounding boxes, and sketches (Zhuang et al., 2024; He et al., 2024; Mikaeili et al., 2023). Semantic components and local 3D regions can be localized with visual, linguistic, geometric, or interactive cues (Liu et al., 2025; Ma et al., 2025b; Simonelli et al., 2025; Lee et al., 2025b; Liang et al., 2026; Wu et al., 2024b). **UMAS3D** further unifies multimodal guidance and adaptive 3D edit localization, allowing Edit/Keep support to evolve with the target structure during generation.

3 Method

3.1 Preliminaries

TRELLIS. We employ the pre-trained image-conditioned TRELLIS (Xiang et al., 2025) as our 3D generation backbone. TRELLIS uses a two-stage process: Stage 1 produces a sparse 3D structure that determines active voxel locations and coarse geometry; Stage 2 generates Structured LATent (SLAT) features on these voxels to capture finer geometry and appearance. The resulting SLAT representation is decoded into the complete 3D asset. Throughout UMAS3D, all TRELLIS parameters remain frozen, and our editing method builds on this process without additional training.

3.2 Framework Overview

As shown in Figure 1, **UMAS3D** provides a unified pipeline from a source 3D asset and multimodal editing instructions to a target 3D asset. The source 3D object is first rendered into multi-view images, from which an editing-relevant source view is selected. A target image is then generated

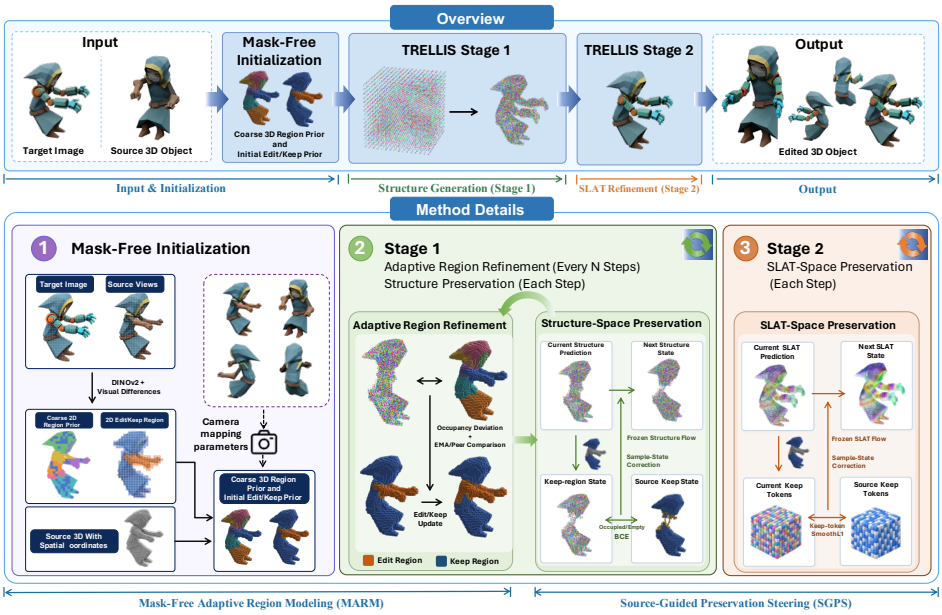

Figure 2: **Core 3D editing pipeline of UMAS3D.** MARM initializes and adaptively refines the Edit/Keep regions. SGPS preserves source-consistent content across structure and SLAT spaces.

from the selected source view and multimodal editing conditions to guide the subsequent 3D editing process. Addition, deletion, modification, appearance, and pose share the same pipeline.

As illustrated in Figure 2, to avoid external 3D masks while preserving unedited source content, we introduce **MARM** and **SGPS**. MARM constructs an initial 3D Edit/Keep prior and adaptively updates region assignments the target structure evolves (Section 3.4). SGPS preserves source consistency in the current Keep regions, maintaining source-consistent structure and source-aligned SLAT features (Section 3.5).

3.3 Unified Multimodal Editing Guidance

Multimodal Condition Unification. UMAS3D supports text instructions, bounding boxes, and reference images, denoted by T, B, and R, respectively. We consider five input combinations:

$$\mathcal{C} \in \{T,\ T+B,\ T+R,\ B+R,\ T+B+R\}. \tag{1}$$

Given $\mathcal{C}$, Gemini 3.6 Flash (Google DeepMind, 2026) interprets the available conditions and constructs a unified editing description E, while B and R provide spatial and visual guidance, respectively. The processing of different combinations is summarized in Table 1.

Source-View Selection and Target-Image Generation. Given source views $\mathbf{I}_s = \{\mathbf{I}_s^v\}_{v=1}^V$ rendered from the source asset $\mathbf{X}_s$, the source view associated with B is directly used when available; otherwise, Gemini 3.6 Flash selects the source view $\mathbf{I}_s^{v^*}$ most relevant to the intended edit. It then constructs the image-editing prompt from the selected view, E, and the available conditions. Gemini 3 Pro Image (Google DeepMind, 2025) then generates the target image $\mathbf{I}_t$:

$$E = \mathcal{F}_{pre}(\mathcal{C}), \qquad \mathbf{I}_t = \mathcal{F}_{edit}\left(\mathbf{I}_s^{v^*}, E, \delta_B B, \delta_R R\right), \tag{2}$$

where $\delta_B, \delta_R \in \{0,1\}$ indicate whether the corresponding modalities are present. For $B+R$, the editing semantics and prompt are inferred from the spatial and visual cues.

Unified Downstream 3D Editing. The target image $\mathbf{I}_t$ serves as the common interface for 3D editing. Once obtained, addition, deletion, modification, appearance, and pose share the same pipeline, thereby achieving condition-level and task-level unification. Reference image conditions are not used for deletion and pose, as these edits are defined by content removal or articulation changes and do not require an external visual exemplar. The supported configurations are summarized in Table 2, with further rationale provided in the Appendix.

Table 1: Processing of multimodal conditions.

Condition	Semantic	Spatial	Visual	View Selection
T	T	–	–	VLM selection
$T+B$	T	B	–	BBox view
$T+R$	T	–	R	VLM selection
$B+R$	VLM-inferred	B	R	BBox view
$T+B+R$	T	B	R	BBox view

Table 2: Supported multimodal conditions.

Edit Type	T	T+B	T+R	B+R	T+B+R
Addition	✓	✓	✓	✓	✓
Deletion	✓	✓	×	×	×
Modification	✓	✓	✓	✓	✓
Appearance	✓	✓	✓	✓	✓
Pose	✓	✓	×	×	×

3.4 Mask-Free Adaptive Region Modeling

Overview. To avoid manual annotations and predefined 3D masks, we propose **Mask-Free Adaptive Region Modeling (MARM)**. As shown in Figure 2, **MARM** constructs a coarse 3D Edit/Keep prior from source-target visual differences and refines region states as the target structure evolves.

Coarse Edit/Keep Initialization. Given the target image $\mathbf{I}_t$ and corresponding source view $\mathbf{I}_s^{v^*}$ from Section 3.3, we reuse the frozen DINOv2 (Oquab et al., 2023) encoder $\phi(\cdot)$ in TRELLIS to extract patch-level features:

$$\mathbf{H}_t = \phi(\mathbf{I}_t), \qquad \mathbf{H}_s = \phi(\mathbf{I}_s^{v^*}), \tag{3}$$

We compare patch-level source-target DINOv2 correspondences at the same locations, together with foreground, appearance, and edge differences. Strong correspondence with small visual changes provides Keep evidence, whereas weakened correspondence with clear visual changes provides Edit evidence; target-only content without reliable source correspondence is treated as Edit. This yields

$$\mathbf{E}_{2D} = \left(\mathbf{E}_{2D}^{E}, \mathbf{E}_{2D}^{K}\right). \tag{4}$$

To construct complete 3D regions, we use the source 3D, source views, and calibrated cameras. For each source view v, we extract its DINOv2 feature map as

$$\mathbf{H}_s^{v} = \phi(\mathbf{I}_s^{v}), \qquad v = 1, \ldots, V, \tag{5}$$

where $\mathbf{H}_s = \mathbf{H}_s^{v^*}$ corresponds to the selected source view. For each occupied source Grid64 voxel x, we project its center to visible views and aggregate the corresponding DINOv2 features:

$$p_v(x) = \pi(\Pi_v x), \qquad \mathbf{f}_x = \mathrm{Agg}_{v \in \mathcal{V}(x)} \, \mathbf{H}_s^{v}(p_v(x)), \tag{6}$$

where Π_v is the camera projection of view v, $\pi(\cdot)$ denotes perspective projection, $\mathcal{V}(x)$ is the visible-view set, and $\mathbf{f}_x$ is the aggregated voxel feature. We then cluster voxels by DINOv2 feature similarity and 3D spatial distance, with connectivity regularization, to obtain 3D regions. Each region is projected through Π_{v^*} to the 2D Edit/Keep evidence map, where the aggregated evidence determines its initial state:

$$\mathcal{R} = \{R_i\}_{i=1}^{N_R}, \qquad \mathbf{M}_0 = \left(\mathbf{M}_0^{E}, \mathbf{M}_0^{K}\right). \tag{7}$$

R_i denotes a 3D region. Edit-dominant and Keep-dominant regions are initialized as Edit and Keep; source-empty locations with strong target-side Edit evidence are opened as editable space. Thus, $\mathbf{M}_0$ matches 2D source-target evidence with complete multi-view 3D regions rather than directly back-projecting single 2D masks.

Adaptive Region Refinement. The same 3D regions $\mathcal{R}$ serve as refinement units. Since $\mathbf{M}_0$ is estimated before the target structure is fully formed, we first allow the target structure to evolve during an initial warm-up period, after which the region states are re-evaluated every five Stage 1 steps. Let $\mathbf{O}_t$ and $\mathbf{O}_s$ denote the current target and aligned source Grid64 occupancy states used for region refinement. For each R_i, we compute its structural deviation and EMA:

$$\ell_i^t = \frac{1}{|R_i|} \sum_{x \in R_i} |\mathbf{O}_t(x) - \mathbf{O}_s(x)|, \qquad \bar{\ell}_i^t = \lambda \bar{\ell}_i^{t-1} + (1-\lambda)\ell_i^t. \tag{8}$$

We further compare each region with others sharing the same current state:

$$d_i^t = \bar{\ell}_i^t - \underset{j \in \mathcal{P}(i)}{\mathrm{Median}} \left(\bar{\ell}_j^t\right), \tag{9}$$

where $\mathcal{P}(i)$ denotes regions sharing the current Edit/Keep state of R_i. If an Edit region persistently shows smaller deviation, its structure remains close to the source and it is reassigned to Keep. Conversely, a Keep region with persistently larger target-driven deviation is released to Edit. Let t^+

denote the next refinement checkpoint:

$$\mathbf{M}_{t^+}(R_i) = \begin{cases} K, & \mathbf{M}_t(R_i) = E,\ d_i^t < -\tau^K, \\ E, & \mathbf{M}_t(R_i) = K,\ d_i^t > \tau^E, \\ \mathbf{M}_t(R_i), & \text{otherwise.} \end{cases} \tag{10}$$

Here, τ^K and τ^E are conservative transition margins. State changes require persistent evidence across consecutive refinement checkpoints, preventing transient fluctuations from causing unstable updates. **MARM** thus progressively updates the Edit/Keep partition as the target structure evolves and determines where SGPS applies source preservation.

3.5 Source-Guided Preservation Steering

Overview. To preserve unedited content without restricting the intended edit, we introduce **Source-Guided Preservation Steering (SGPS)**. As shown in Figure 2, **MARM** determines where source preservation applies, while **SGPS** maintains source consistency within the current Keep regions during inference. **SGPS** preserves occupancy consistency in structure space and SLAT features. All operations update only the current generative state, with pretrained model parameters kept frozen.

Structure-Space Preservation. In Stage 1, SGPS preserves source occupancy within the Keep regions. Let $\mathbf{x}_t$ denote the structure latent at timestep t, and $\hat{\mathbf{x}}_{0,t}$ the corresponding clean prediction, which is decoded into Grid64 occupancy logits $\mathbf{P}_t$. Given the source-occupied voxel set $\mathbf{S}_o$ and the valid non-editable empty support $\mathbf{S}_e$, the current Keep support is divided into

$$\mathbf{M}_t^{K_o} = \mathbf{M}_t^K \cap \mathbf{S}_o, \qquad \mathbf{M}_t^{K_e} = \mathbf{M}_t^K \cap \mathbf{S}_e. \tag{11}$$

$$\Delta\mathbf{x}_{0,t} = \mathbf{x}_{0,t}^s - \hat{\mathbf{x}}_{0,t}, \qquad \tilde{\mathbf{x}}_{0,t} = \hat{\mathbf{x}}_{0,t} + \mathbf{W}_t^K \odot \Delta\mathbf{x}_{0,t}. \tag{12}$$

The source-aligned clean state is then recombined with the corresponding noise endpoint under the original PNS rule, while non-Keep regions remain target-driven. In addition to the source-aligned clean-state correction above, the BCE term is applied only to source-occupancy violations within the current Keep regions:

$$\mathcal{L}_1^o = \mathrm{BCE}\left(\mathbf{P}_t[\mathcal{E}_t^o], 1\right), \qquad \mathcal{L}_1^e = \mathrm{BCE}\left(\mathbf{P}_t[\mathcal{E}_t^e], 0\right), \qquad \mathbf{x}_t \leftarrow \mathrm{AdamStep}\left(\mathbf{x}_t; \mathcal{L}_1^o, \mathcal{L}_1^e\right). \tag{13}$$

where $\mathcal{E}_t^o = \left\{ v \in \mathbf{M}_t^{K_o} \mid \mathbf{P}_t(v) \le 0 \right\}$ and $\mathcal{E}_t^e = \left\{ v \in \mathbf{M}_t^{K_e} \mid \mathbf{P}_t(v) > 0 \right\}$. These sets capture occupied-to-empty and empty-to-occupied errors, respectively. Here, $\mathrm{AdamStep}$ denotes a sample-level IAS update of the current structure state without modifying model parameters. The updated state is re-evaluated by the frozen structure flow and advanced using the original TRELLIS update rule. After Stage 1, a final Keep-region Grid64 composition is applied.

SLAT-Space Preservation. Occupancy preservation alone cannot fully retain local geometry, material, and appearance. In Stage 2, we therefore fix the final region partition from Stage 1 and extend source consistency to the SLAT feature space. Let Z_t denote the current SLAT latent at timestep t, Z_s the source SLAT feature set, and $\hat{Z}_{0,t}$ the corresponding clean SLAT prediction obtained from the frozen SLAT Flow. Source and current SLAT features are aligned through their 3D sparse coordinates. The final preserved region $\mathbf{M}_T^K$ is mapped to the Keep-token set Ω_{slat}^K, where

$$\tilde{Z}_{0,t}^i = Z_s^i, \qquad i \in \Omega_{\mathrm{slat}}^K. \tag{14}$$

$\tilde{Z}_{0,t}^i$ defines a clean state rather than directly overwriting the noisy SLAT state. It is then recombined with the fixed initial-noise endpoint; To smooth the transition between the final Keep and Edit regions, we derive a narrow boundary token set Ω_{slat}^B from the final partition $\mathbf{M}_T$. This boundary band is used only for soft blending and does not introduce an additional **MARM** region state. Keep tokens retain source consistency, tokens in Ω_{slat}^B are softly blended, and the remaining tokens stay target-driven. The recombined state is re-evaluated by the frozen SLAT flow, after which source supervision is applied only to Keep tokens:

$$\mathcal{L}_2 = \frac{1}{|\Omega_{\mathrm{slat}}^K|} \sum_{i \in \Omega_{\mathrm{slat}}^K} \mathrm{SmoothL1}\left(\hat{Z}_{0,t}^i, Z_s^i\right), \qquad Z_t \leftarrow \mathrm{AdamStep}\left(Z_t, \mathcal{L}_2\right). \tag{15}$$

The SmoothL1 gradient updates only the current Keep-token state, while the remaining tokens receive no source-feature supervision and remain target-driven. The updated SLAT state is re-evaluated by the frozen SLAT Flow and advanced using the original TRELLIS update rule. Before decoding, a final Source-SLAT composition is applied on the Keep tokens. Thus, **SGPS** preserves source structure and fine-grained SLAT features while leaving the edited regions target-driven.

Table 3: Quantitative comparison with existing 3D editing methods. Best results are highlighted in **bold** with a light red background.

		Unedited-region Preservation			Overall Target Quality			
Method	Venue	Pres-IoU↑	Pres-PSNR↑	Pres-SSIM↑	DINO-I↑	CD-L1↓	F@1%↑	NC↑
EditP23	*arXiv'25*	0.1745	7.8001	0.4363	0.5598	0.0612	0.3684	0.6905
Easy3E	*CVPR'26*	0.5643	17.0366	0.7149	0.8568	0.0190	0.7719	0.8502
VoxHammer	*3DV'26*	0.5691	16.1752	0.6056	0.8468	0.0186	0.7598	0.8521
Nano3D	*ICLR'26*	0.5580	16.0651	0.6176	0.8554	0.0182	0.7696	0.8540
Vinedresser3D	*CVPR'26*	0.6030	19.8795	0.7913	0.8797	0.0158	0.8031	0.8712
3DEditFormer	*ICML'26*	0.5881	20.0870	0.7399	0.8878	0.0149	0.8096	0.8838
UMAS3D	*Ours*	**0.7217**	**24.9473**	**0.8586**	**0.8936**	**0.0134**	**0.8194**	**0.8995**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets and Multimodal Instructions. We evaluate UMAS3D on 150 3D objects from Pxform (Weng et al., 2026) and Omni3DEdit (Fan et al., 2026), covering addition, deletion, ,odification, appearance, and pose. A small number of Objaverse (Deitke et al., 2023) assets are used only for qualitative evaluation and excluded from quantitative statistics. Since these datasets do not provide complete Text, Bounding Box, and Reference Image conditions, we preserve the original editing intent and construct missing modalities from source multi-view observations. Gemini 3.6 Flash (Google DeepMind, 2026) generates Bounding Boxes and modality-specific text variants, while Gemini 3 Pro Image (Google DeepMind, 2025) generates Reference Images, yielding five settings: T, $T + B$, $T + R$, $B + R$, and $T + B + R$. Some details are provided in the Appendix.

Baselines. We compare against training-based 3DEditFormer (Xia et al., 2025b), inversion-based VoxHammer (Li et al., 2026) and Vinedresser3D (Chi et al., 2026), training-free FlowEdit-based Nano3D (Ye et al., 2026), and feed-forward EditP23 (Bar-On et al., 2025) and Easy3E (Hu et al., 2026). Details are provided in the Appendix. For quantitative evaluation only, we render four canonical views from the ground-truth target 3D of each Pxform and Omni3DEdit sample and manually select one representative view as the target image for all methods; the resulting prediction is then evaluated against the corresponding ground-truth target 3D.

Evaluation Metrics. We evaluate *Unedited-region Preservation* using Pres-IoU, Pres-PSNR, and Pres-SSIM, which measure geometric and appearance preservation between the source and prediction within unedited regions. We evaluate *Overall Target Quality* using DINO-I, CD-L1, F@1%, and NC, measuring visual-semantic and geometric consistency between the prediction and the ground-truth target. Details are provided in the Appendix.

4.2 COMPARISON RESULTS

As shown in Table 3, compared with existing methods, UMAS3D achieves the best overall performance in both *Unedited-region Preservation* and *Overall Target Quality*, demonstrating its ability to preserve unedited source content while producing high-quality target edits. The results show that UMAS3D better preserves source geometry and appearance, while maintaining higher consistency with the benchmark target in terms of overall visual semantics, geometric structure, and surface quality, thereby effectively reducing unintended changes during editing.

As shown in Figure 3, UMAS3D accurately follows the specified editing intent while better preserving source structures and local details unrelated to the edit. Compared with existing methods, its results exhibit more complete edits, more stable geometric structures, and fewer unintended changes in unrelated regions. Together, the quantitative and qualitative results validate the complementary roles of **MARM** and **SGPS**, enabling **UMAS3D** to achieve the intended edits while preserving source content more consistently.

4.3 HUMAN EVALUATION OF EDITING QUALITY

To directly assess instruction fulfillment and perceived editing quality, we conduct a human evaluation with 40 participants. We compare UMAS3D with four representative baselines—Nano3D,

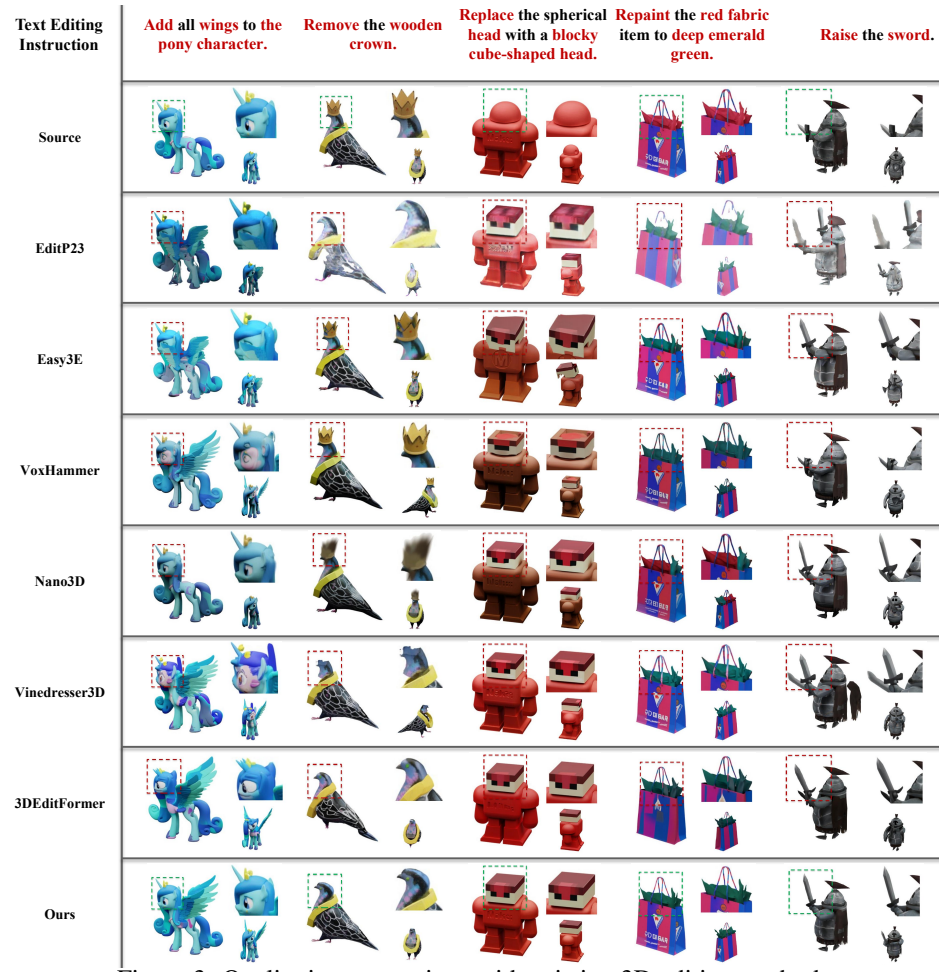

Figure 3: Qualitative comparison with existing 3D editing methods.

Table 4: Human evaluation results. Each value denotes the percentage of times a method is selected as the preferred result under the corresponding criterion.

Method	Instruction Alignment ↑	Visual Quality ↑	Source Preservation ↑
VoxHammer	7%	7%	8%
Nano3D	7%	9%	8%
Vinedresser3D	8%	11%	9%
3DEditFormer	21%	19%	14%
UMAS3D	**57%**	**54%**	**61%**

VoxHammer, Vinedresser3D and 3DEditFormer. For each case, anonymized results are shown in random order under identical rendering settings, together with the source 3D asset and editing instruction. Participants select the best result in terms of Instruction Alignment, Visual Quality and Source Preservation. As shown in Table 4, UMAS3D receives the highest preference.

4.4 Ablation Studies

As shown in Table 5, we construct five ablation variants, **A–E**. **A** denotes the original TRELLIS; **B** introduces the structure-space supervision of **SGPS**, while **C** further introduces SLAT-space supervision. **D** additionally enables adaptive Edit/Keep adjustment. **E** replaces the external 3D masks with mask-free initialization while retaining the two-stage **SGPS**.

The ablation results show that introducing Stage 1 **SGPS** noticeably improves the preservation of unedited structures and overall geometric quality, indicating that source-consistency constraints in structure space effectively suppress unintended geometric changes during generation. Building upon

Table 5: Quantitative ablation of UMAS3D components. *Ext.*: external 3D-mask initialization; *Mask-free*: MARM initialization without external 3D masks; *S1/S2 SGPS*: Stage 1/Stage 2 SGPS; *Adapt.*: adaptive Edit/Keep refinement. Best results are highlighted in **bold** with a light background.

	Components				Unedited-region Preservation			Overall Target Quality			
Model	Region Init.	S1 SGPS	S2 SGPS	Adapt.	Pres-IoU↑	Pres-PSNR↑	Pres-SSIM↑	DINO-I↑	CD-L1↓	F@1%↑	NC↑
A	–	–	–	–	0.1328	9.2586	0.4355	0.8644	0.0570	0.2861	0.6755
B	Ext.	✓	–	–	0.5437	16.0835	0.2619	0.8521	0.0162	0.7949	0.8633
C	Ext.	✓	✓	–	0.6438	17.2126	0.6770	0.8780	0.0153	0.8001	0.8800
D	Ext.	✓	✓	✓	0.6341	19.8069	0.6966	0.8798	0.0146	0.8048	0.8789
E	Mask-free	✓	✓	–	0.7094	18.6103	0.7737	0.8778	0.0152	0.8022	0.8959
UMAS3D	**Mask-free**	✓	✓	✓	**0.7217**	**24.9473**	**0.8586**	**0.8936**	**0.0134**	**0.8194**	**0.8995**

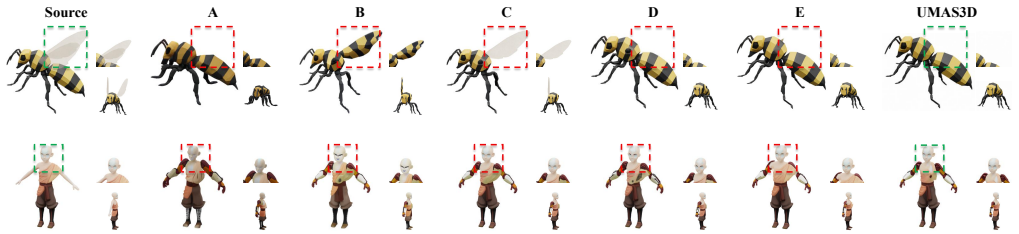

Figure 4: **Qualitative ablation of UMAS3D.** Variants follow Table 5. The first-row instruction is "Remove both bee wings.", and the second-row instruction is "Replace the human arm structure with robotic mechanical arms."

this, adding Stage 2 **SGPS** further improves the preservation of appearance and local details in Keep regions, while also enhancing overall visual consistency. This demonstrates that **SGPS** provides complementary source-preservation capabilities in structure space and SLAT feature space.

Further introducing adaptive Edit/Keep adjustment improves both *Unedited-region Preservation* and *Overall Target Quality*, showing that dynamically adjusting Edit/Keep regions during generation provides more accurate control over the Edit/Keep areas than using a fixed initial partition, thereby reducing unintended modifications to unrelated source content. Meanwhile, replacing the external 3D mask with mask-free initialization still maintains strong editing quality and source-preservation capability, validating the effectiveness of **MARM** in constructing a reliable Edit/Keep prior without requiring an external 3D mask.

As shown in Figure 4, the qualitative results exhibit a trend consistent with the quantitative evaluation. With the progressive introduction of the two-stage **SGPS** and adaptive Edit/Keep adjustment, source structure and local appearance are preserved more consistently, while the editing regions are controlled more accurately. The full **UMAS3D** therefore achieves reliable target editing and source-content preservation without relying on an external 3D mask, further validating the effectiveness and complementary roles of **MARM** and **SGPS**.

5 Conclusion

This paper presents UMAS3D, a unified 3D editing framework that is training-free, inversion-free, and free from external 3D masks. UMAS3D unifies diverse multimodal editing instructions into target-view guidance and handles multiple editing tasks through a shared 3D editing pipeline. We introduce Mask-Free Adaptive Region Modeling (MARM) to initialize and progressively refine the Edit/Keep regions during generation, and Source-Guided Preservation Steering (SGPS) to preserve source-consistent geometry and appearance in the Keep regions while leaving the Edit regions target-driven. Experiments across diverse multimodal conditions and editing tasks demonstrate effective target editing together with strong preservation of unedited source content.

Limitations. UMAS3D does not explicitly resolve conflicts among multimodal editing conditions. For example, if the text instruction asks to recolor a cup red while the reference image shows a blue cup, the framework lacks a dedicated mechanism to determine which condition should take priority.

AI USE STATEMENT

Generative AI tools were used to assist with parts of the research and writing process. In the experimental pipeline, we used Gemini models to generate or supplement multimodal editing conditions, including bounding boxes, textual conditions, and reference images, and Gemini 3 Pro Image was further used to generate the target-view guidance employed by UMAS3D; the corresponding usage is described in the main paper and Appendix. Generative AI tools were also used to assist with code development and debugging, as well as language editing of the manuscript. All AI-assisted code was executed and checked for correctness, the generated multimodal conditions and target views were manually reviewed by the authors, and the manuscript text was carefully inspected and verified by the authors.

REPRODUCIBILITY STATEMENT

To improve the reproducibility of our work, the main paper presents the complete UMAS3D pipeline together with the core methodology and mathematical formulations of MARM and SGPS, and details the datasets, baselines, and evaluation metrics used in our experiments. The Appendix further provides the construction of multimodal data, implementation settings, key hyperparameters, and additional evaluation details that are omitted from the main text due to space limitations. Together, these methodological and experimental specifications provide the key information needed to reproduce the main results reported in this work.

REFERENCES

Chong Bao, Yinda Zhang, Bangbang Yang, Tianxing Fan, Zesong Yang, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Sine: Semantic-driven image-based nerf editing with prior-guided editing field. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20919–20929. IEEE, 2023.

Roi Bar-On, Dana Cohen-Bar, and Daniel Cohen-Or. Editp23: 3d editing via propagation of image prompts to multi-view. *arXiv preprint arXiv:2506.20652*, 2025.

Amir Barda, Matheus Gadelha, Vladimir G Kim, Noam Aigerman, Amit H Bermano, and Thibault Groueix. Instant3dit: Multiview inpainting for fast editing of 3d objects. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16273–16282. IEEE, 2025.

Weiwei Cai, Shuangkang Fang, Weicai Ye, Xin Dong, Yunhan Yang, Xuanyang Zhang, Wei Cheng, Yanpei Cao, Gang Yu, and Tao Chen. Native 3d editing with full attention. *arXiv preprint arXiv:2511.17501*, 2025.

Hansheng Chen, Ruoxi Shi, Yulin Liu, Bokui Shen, Jiayuan Gu, Gordon Wetzstein, Hao Su, and Leonidas Guibas. Generic 3d diffusion adapter using controlled multi-view editing. *arXiv preprint arXiv:2403.12032*, 2024a.

Honghua Chen, Yushi Lan, Yongwei Chen, Yifan Zhou, and Xingang Pan. Mvdrag3d: Drag-based creative 3d editing via multi-view generation-reconstruction priors. *arXiv preprint arXiv:2410.16272*, 2024b.

Minghao Chen, Iro Laina, and Andrea Vedaldi. Dge: Direct gaussian 3d editing by consistent multi-view editing. In *European conference on computer vision*, pp. 74–92. Springer, 2024c.

Minghao Chen, Junyu Xie, Iro Laina, and Andrea Vedaldi. Shap-editor: Instruction-guided latent 3d editing in seconds. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26446–26456. IEEE, 2024d.

Xi Chen, Zhifei Zhang, He Zhang, Yuqian Zhou, Soo Ye Kim, Qing Liu, Yijun Li, Jianming Zhang, Nanxuan Zhao, Yilin Wang, et al. Unireal: Universal image generation and editing via learning real-world dynamics. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12501–12511. IEEE, 2025.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21476–21485. IEEE, 2024e.

Yankuan Chi, Xiang Li, Zixuan Huang, and James Matthew Rehg. Vinedresser3d: Towards agentic text-guided 3d editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12673–12683, 2026.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13142–13153, 2023.

Ziya Erkoç, Can Gümeli, Chaoyang Wang, Matthias Nießner, Angela Dai, Peter Wonka, Hsin-Ying Lee, and Peiye Zhuang. Preditor3d: Fast and precise 3d shape editing. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 640–649. IEEE, 2025.

Hongxing Fan, Haotian Lu, Rui Chen, Weibin Yun, Zehuan Huang, and Lu Sheng. Omni3dedit: A unified 3d editing benchmark with region annotations. `https://fanhongxing.github.io/Omni3DEdit/`, 2026. Project page, accessed August 19, 2026.

Will Gao, Dilin Wang, Yuchen Fan, Aljaz Bozic, Tuur Stuyck, Zhengqin Li, Zhao Dong, Rakesh Ranjan, and Nikolaos Sarafianos. 3d mesh editing using masked lrms. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7154–7165. IEEE, 2025.

Inbar Gat, Dana Cohen Bar, Guy Levy, Elad Richardson, and Daniel Cohen Or. Shapeup: Scalable image-conditioned 3d editing. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2026.

Google DeepMind. Gemini 3 pro image: Model card. `https://deepmind.google/models/model-cards/gemini-3-pro-image/`, 2025. Model card, published November 2025, accessed August 19, 2026.

Google DeepMind. Gemini 3.6 flash: Model card. `https://deepmind.google/models/model-cards/gemini-3-6-flash/`, 2026. Model card, published July 21, 2026, accessed August 19, 2026.

Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19683–19693. IEEE, 2023.

Runze He, Shaofei Huang, Xuecheng Nie, Tianrui Hui, Luoqi Liu, Jiao Dai, Jizhong Han, Guanbin Li, and Si Liu. Customize your nerf: Adaptive source driven 3d scene editing via local-global iterative training. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6966–6975. IEEE, 2024.

Runze He, Kai Ma, Linjiang Huang, Shaofei Huang, Jialin Gao, Xiaoming Wei, Jiao Dai, Jizhong Han, and Si Liu. Freeedit: Mask-free reference-based image editing with multi-modal instruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Susung Hong, Johanna Karras, Ricardo Martin-Brualla, and Ira Kemelmacher-Shlizerman. Perturb-and-revise: Flexible 3d editing with generative trajectories. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16293–16303. IEEE, 2025.

Shimin Hu, Yuanyi Wei, Fei Zha, Yudong Guo, and Juyong Zhang. Easy3e: Feed-forward 3d asset editing via rectified voxel flow. *arXiv preprint arXiv:2602.21499*, 2026.

Yufei Huang, Bangyan Liao, Yuqi Hu, Haitao Lin, Lirong Wu, Siyuan Li, Cheng Tan, Zicheng Liu, Yunfan Liu, Zelin Zang, et al. Dacapo: Score distillation as stacked bridge for fast and high-quality 3d editing. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16304–16313. IEEE, 2025.

Hayeon Kim, Ji Ha Jang, and Se Young Chun. Robust 3d-masked part-level editing in 3d gaussian splatting with regularized score distillation sampling. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5501–5510. IEEE, 2025a.

Jiwook Kim, Seonho Lee, Jaeyo Shin, Jiho Choi, and Hyunjung Shim. Dreamcatalyst: Fast and high-quality 3d editing via controlling editability and identity preservation. In *International Conference on Learning Representations*, volume 2025, pp. 55846–55872, 2025b.

Juil Koo, Chanho Park, and Minhyuk Sung. Posterior distillation sampling. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13352–13361. IEEE, 2024.

Dong In Lee, Hyeongcheol Park, Jiyoung Seo, Eunbyung Park, Hyunje Park, Ha Dam Baek, Sangheon Shin, Sangmin Kim, and Sangpil Kim. Editsplat: Multi-view fusion and attention-guided optimization for view-consistent 3d scene editing with 3d gaussian splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11135–11145. IEEE, 2025a.

Junha Lee, Chunghyun Park, Jaesung Choe, Yu-Chiang Frank Wang, Jan Kautz, Minsu Cho, and Chris Choy. Mosaic3d: Foundation dataset and model for open-vocabulary 3d segmentation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14089–14101. IEEE, 2025b.

Lin Li, Zehuan Huang, Haoran Feng, Gengxiong Zhuang, Rui Chen, Chunchao Guo, and Lu Sheng. Voxhammer: Training-free precise and coherent 3d editing in native 3d space. In *2026 International Conference on 3D Vision (3DV)*, pp. 1281–1292. IEEE, 2026.

Yuhan Li, Yishun Dou, Yue Shi, Yu Lei, Xuanhong Chen, Yi Zhang, Peng Zhou, and Bingbing Ni. Focaldreamer: Text-driven 3d editing via focal-fusion assembly. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 3279–3287, 2024.

Siyun Liang, Sen Wang, Kunyi Li, Michael Niemeyer, Stefano Gasperini, Hendrik PA Lensch, Nassir Navab, and Federico Tombari. Supergseg: Open-vocabulary 3d segmentation with structured super-gaussians. In *2026 International Conference on 3D Vision (3DV)*, pp. 1650–1659. IEEE, 2026.

Hao Liu, Yuxuan Lin, Jingfeng Guo, Ruihang Chu, Junjie Wang, Ruotong Li, and Yujiu Yang. Velocity-space 3d asset editing. *arXiv preprint arXiv:2605.07385*, 2026.

Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. Partfield: Learning 3d feature fields for part segmentation and beyond. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9704–9715. IEEE, 2025.

Guan Luo, Tian-Xing Xu, Ying-Tian Liu, Xiao-Xiong Fan, Fang-Lue Zhang, and Song-Hai Zhang. 3d gaussian editing with a single image. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 6627–6636, 2024.

Ziqi Ma, Hongqiao Chen, Yisong Yue, and Georgia Gkioxari. Feedforward 3d editing via text-steerable image-to-3d. *arXiv preprint arXiv:2512.13678*, 2025a.

Ziqi Ma, Yisong Yue, and Georgia Gkioxari. Find any part in 3d. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–10. IEEE, 2025b.

Aryan Mikaeili, Or Perel, Mehdi Safaee, Daniel Cohen-Or, and Ali Mahdavi-Amiri. Sked: Sketch-guided text-based 3d editing. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14561–14573. IEEE, 2023.

Rui Nie, Chuang Wang, Haitao Zhou, Jiahe Song, Buyu Li, Sheng Wang, and Qian Yu. Editflow3d: Automated local editing of 3d assets with trajectory preservation. *arXiv preprint arXiv:2608.03179*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Francesco Palandra, Andrea Sanchietti, Daniele Baieri, and Emanuele Rodola. Gsedit: Efficient text-guided editing of 3d objects via gaussian splatting. *arXiv preprint arXiv:2403.05154*, 2024.

Maria Parelli, Michael Oechsle, Michael Niemeyer, Federico Tombari, and Andreas Geiger. 3d-latte: Latent space 3d editing from textual instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14377–14386, 2026.

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

Yansong Qu, Dian Chen, Xinyang Li, Xiaofan Li, Shengchuan Zhang, Liujuan Cao, and Rongrong Ji. Drag your gaussian: Effective drag-based editing with score distillation for 3d gaussian splatting. In *Proceedings of the special interest group on computer graphics and interactive techniques conference conference papers*, pp. 1–12, 2025.

Etai Sella, Gal Fiebelman, Peter Hedman, and Hadar Averbuch-Elor. Vox-e: Text-guided voxel editing of 3d objects. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 430–440. IEEE, 2023.

Sitian Shen, Jing Xu, Yuheng Yuan, Xingyi Yang, Qiuhong Shen, and Xinchao Wang. Draggaussian: Enabling drag-style manipulation on 3d gaussian representation. *arXiv preprint arXiv:2405.05800*, 2024.

Andrea Simonelli, Norman Müller, and Peter Kontschieder. Easy3d: A simple yet effective method for 3d interactive segmentation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24707–24716. IEEE, 2025.

Hyeonseop Song, Seokhun Choi, Hoseok Do, Chul Lee, and Taehyeong Kim. Blending-nerf: Text-driven localized editing in neural radiance fields. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14337–14347. IEEE, 2023.

Yanhao Sun, Runze Tian, Xiao Han, XinYao Liu, Yan Zhang, and Kai Xu. Gseditpro: 3d gaussian splatting editing with attention-based progressive localization. In *Computer Graphics Forum*, volume 43, pp. e15215. Wiley Online Library, 2024.

Xueyun Tian, Wei Li, Bingbing Xu, Yige Yuan, Yuanzhuo Wang, and Huawei Shen. Mige: A unified framework for multimodal instruction-based image generation and editing. *arXiv preprint arXiv:2502.21291*, 3, 2025.

Junjie Wang, Jiemin Fang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. Gaussianeditor: Editing 3d gaussians delicately with text instructions. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20902–20911. IEEE, 2024.

Jiawei Weng, Saining Zhang, Zhenxin Diao, Peishuo Li, Henghaofan Zhang, Junhao Chen, and Hao Zhao. Feedforward 3d editing learns from semantic-part transformation. *arXiv preprint arXiv:2605.27351*, 2026.

Cunqi Wu, Peng Zhou, Jie Qin, and Qi Tian. 100editor: 100+ views per batch and minute-scale view-consistent 3d editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8449–8460, 2026.

Jing Wu, Jia-Wang Bian, Xinghui Li, Guangrun Wang, Ian Reid, Philip Torr, and Victor Adrian Prisacariu. Gaussctrl: Multi-view consistent text-driven 3d gaussian splatting editing. In *European conference on computer vision*, pp. 55–71. Springer, 2024a.

Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, et al. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. *Advances in Neural Information Processing Systems*, 37:19114–19138, 2024b.

Bin Xia, Yuechen Zhang, Jingyao Li, Chengyao Wang, Yitong Wang, Xinglong Wu, Bei Yu, and Jiaya Jia. Dreamomni: Unified image generation and editing. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28533–28543. IEEE, 2025a.

Ruihao Xia, Yang Tang, and Pan Zhou. Towards scalable and consistent 3d editing. *arXiv preprint arXiv:2510.02994*, 2025b.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480. IEEE, 2025.

Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13294–13304. IEEE, 2025.

Junliang Ye, Shenghao Xie, Ruowen Zhao, Zhengyi Wang, Hongyu Yan, Wenqiang Zu, Lei Ma, and Jun Zhu. Nano3d: A training-free approach for efficient 3d editing without masks. In *International Conference on Learning Representations*, volume 2026, pp. 46900–46922, 2026.

Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26125–26135. IEEE, 2025.

Chao Zhang, Fang Liu, Shuo Li, Yang Liu, Jiahao Wang, Xinyan Huang, Lingling Li, Puhua Chen, Xu Liu, Wenping Ma, et al. Vdfe: Difference-aware 3d scene editing with non-intrusive video diffusion priors for multi-view consistency and efficiency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18375–18385, 2026.

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 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*, pp. 14387–14397, 2026.

Jingyu Zhuang, Chen Wang, Liang Lin, Lingjie Liu, and Guanbin Li. Dreameditor: Text-driven 3d scene editing with neural fields. In *SIGGRAPH Asia 2023 conference papers*, pp. 1–10, 2023.

Jingyu Zhuang, Di Kang, Yan-Pei Cao, Guanbin Li, Liang Lin, and Ying Shan. Tip-editor: An accurate 3d editor following both text-prompts and image-prompts. *ACM Transactions on Graphics (ToG)*, 43(4):1–12, 2024.

A Appendix

A.1 Dataset Construction

Our quantitative benchmark is constructed from Pxform and Omni3DEdit and contains 150 source 3D assets spanning five editing tasks: Addition, Deletion, Modification, Appearance, and Pose, with 32, 29, 30, 29, and 30 samples, respectively. In addition, a small number of 3D assets from Objaverse are used for qualitative demonstrations on the project page and in the supplementary material to examine the applicability of UMAS3D to a broader range of object categories. These Objaverse samples are used only for additional qualitative visualization and are excluded from the quantitative benchmark, metric computation, and experimental statistics.

Table 6: Construction of the five multimodal editing conditions used in our benchmark. Bracketed terms denote sample-specific content.

Input	Text	Bounding Box	Reference Image
T	*[Original Instruction]* The original editing instruction is retained verbatim without rewriting.	–	–
$T + B$	*[Modality-Specific Text]* Since the Bounding Box provides **where** the edit occurs, redundant spatial descriptions already conveyed by the box are removed, while preserving the editing operation and target content or appearance. The original editing intent remains unchanged.	*[Four Source Views]* + *[Original Instruction]* Select the source view that best exposes the edit-relevant region, identify the **source-side edit anchor**, and return a tight 2D box $[x_1, y_1, x_2, y_2]$ normalized to $[0, 1000]$. For Addition, the box localizes the source-side attachment or placement region rather than future target content absent from the source.	–
$T + R$	*[Modality-Specific Text]* Since the Reference Image provides **what / appearance**, the text retains the editing operation, spatial location, and necessary relation while removing target visual details already represented by the reference.	–	*[Selected Source View]* + *[Original Instruction]* Generate a standalone visual reference that represents only the target visual information required by the edit. For Addition, the reference depicts the object to be added; for Modification, the desired replacement or modified part; and for Appearance, the target color, material, texture, or pattern.
$B + R$	–	Uses the same source-side Bounding Box as $T + B$.	Uses the same Reference Image as $T + R$.
$T{+}B{+}R$	*[Modality-Specific Text]* The Bounding Box provides **where**, while the Reference Image provides **what / appearance**. The text therefore retains only the editing operation and any relation still required to specify the intended edit.	Uses the same source-side Bounding Box as $T + B$.	Uses the same Reference Image as $T + R$.

Since the original datasets do not jointly provide complete Text, Bounding Box, and Reference Image conditions, we construct the missing modalities while preserving the original editing intent. For each sample, the source 3D asset is first rendered from four canonical views, i.e., front, right, back, and left, and the source view that best exposes the region relevant to the requested edit is selected according to the original instruction. The selection considers the visibility of the target part, occlusion, and its spatial relationship with surrounding structures. Based on the selected source view and the original instruction, we then construct the corresponding Bounding Box, Reference Image, and text variants required by different multimodal settings. These components are finally organized into five unified input combinations: T, $T + B$, $T + R$, $B + R$, and $T + B + R$.

The Text-only condition always retains the original editing instruction verbatim. For Bounding-Box-based conditions, Gemini 3.6 Flash analyzes the four source views together with the original editing semantics, selects the most suitable source view, and localizes the source-side edit anchor within that view to construct a normalized 2D box. The box therefore represents an edit-relevant spatial



Figure 5: Examples of the constructed multimodal editing conditions. Each dashed box shows the Bounding Box visualization, the generated Reference Image, and the original textual editing instruction. Modality-specific text variants are constructed according to Table 6.

anchor already present in the source rather than future target content. For reference-based conditions, Gemini 3 Pro Image generates a standalone visual reference from the selected source view and the original editing instruction. The Reference Image represents the object to be added for Addition, the desired replacement or modified part for Modification, and the target color, material, texture, or pattern for Appearance. Reference-based conditions are used only for Addition, Modification, and Appearance. For the same sample, all Bounding-Box-based combinations share the same Bounding Box, while all reference-based combinations share the same Reference Image, ensuring consistent editing semantics across different multimodal settings.

The construction rules for the five multimodal input combinations are summarized in Table 6. Bracketed terms denote sample-specific content inserted into the corresponding prompt.

Table 7: Prompt construction for target-view generation under the five multimodal input combinations. The five settings share the same generation pipeline but differ in the roles assigned to Text, Bounding Box, and Reference Image.

Input	Gemini 3 Pro Image Input	Prompt Construction and Modality Roles
T	*[Source View]*	Text is the only explicit editing condition. The textual instruction defines the requested operation and target semantics, while the spatial location is inferred jointly from the instruction and the source image. The prompt explicitly prohibits introducing content that is not requested by the instruction.
$T + B$	*[Source View] + [BBox Visualization]*	Text defines **what operation** should be performed, whereas the Bounding Box provides only a soft cue for **where** the relevant source region is located. The box is not treated as a hard mask, exact boundary, crop, or target silhouette, and the actual edited area may extend slightly beyond or remain within the box when required for coherent geometry and appearance. The BBox itself must not appear in the generated image.
$T + R$	*[Source View] + [Reference Image]*	Text is authoritative for the editing operation, while the Reference Image provides the requested visual target information. Depending on the editing intent inferred from the available conditions, the reference may provide target identity, shape, parts, color, material, or texture. Its framing, pose, background, shadows, and unrelated content are not directly copied; instead, the relevant visual attributes are adapted to the source image.
$B + R$	*[Source View] + [BBox Visualization] + [Reference Image]*	No user-provided Text condition, original textual instruction, or edit-type label is supplied. The Bounding Box provides approximate **where** information, while the Reference Image provides **what / appearance** information. The editing operation is inferred from the source image, the localized source region, and the visual relationship between the source and the reference, and is represented internally by the unified editing description E.
$T + B + R$	*[Source View] + [BBox Visualization] + [Reference Image]*	All three modalities jointly constrain the edit. Text is authoritative for the requested **operation / semantics**, the Bounding Box provides soft **where** guidance, and the Reference Image provides the desired **what / appearance**. The spatial and visual conditions complement the textual instruction without overriding its editing intent.

The source-side anchor is defined according to the editing task: the attachment or placement region for Addition, the source part to be removed for Deletion, the part to be replaced or modified for Modification, the region whose appearance changes for Appearance, and the articulated component involved in the pose change for Pose. The Reference Image encodes only the target visual information and does not provide spatial localization.

Figure 5 presents representative examples of the constructed multimodal editing conditions. Each dashed box contains the Bounding Box visualization, the corresponding Reference Image, and the original textual editing instruction. For clarity, the figure shows the original instruction rather than the modality-specific text variants, which are constructed according to Table 6.

A.2 Target-View Generation

For each sample, we first load the four canonical renderings of the source 3D asset, i.e., the front, right, back, and left views, together with the current multimodal editing condition. We use Gemini 3.6 Flash as a unified planning module to interpret the editing intent solely from the available input modalities and formulate a unified editing description E, including the relevant source part, target content, requested operation, spatial relationship, target geometry or appearance, and preservation requirements. No predefined edit-type or task-category label is provided during target-view generation; the required editing operation is inferred entirely from the current multimodal condition.

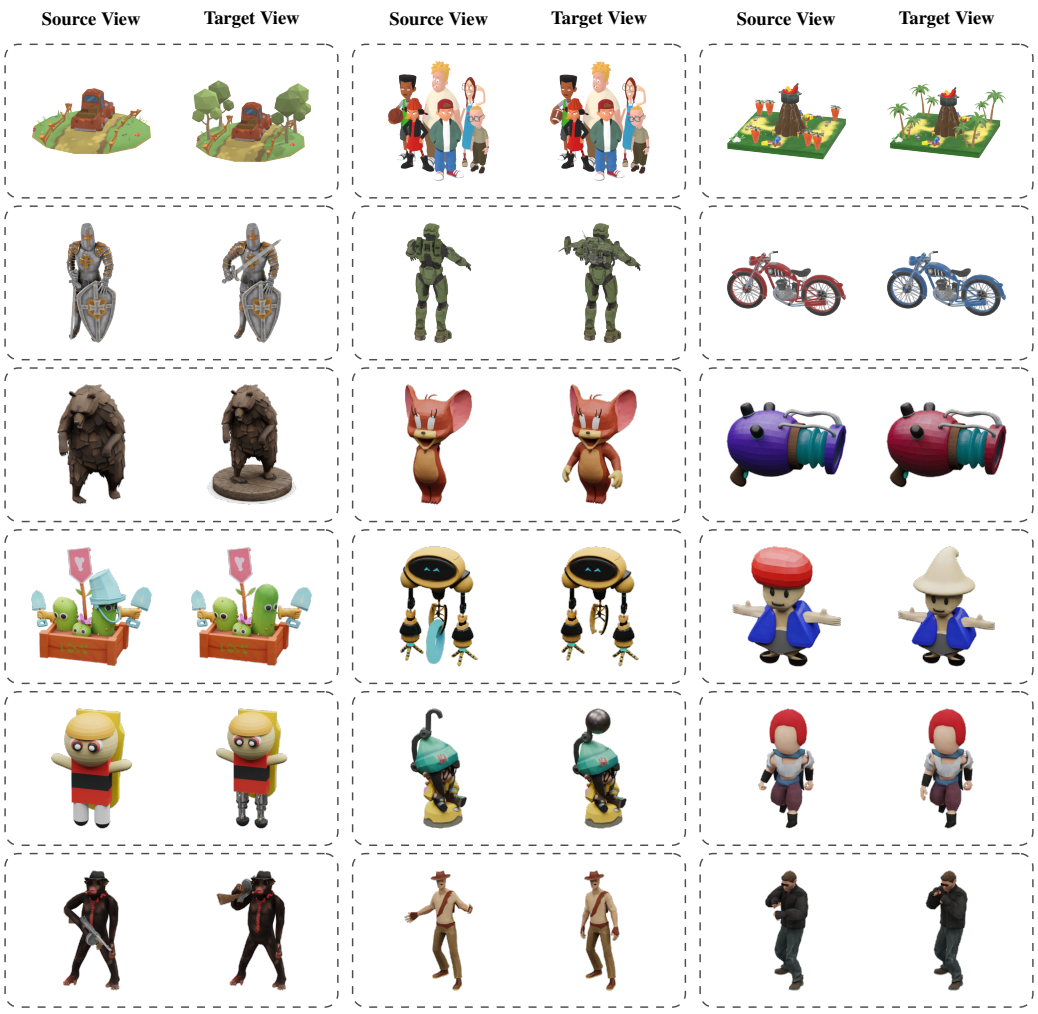

Figure 6: Examples of target-view generation. Each dashed box groups a selected source view and its corresponding target view generated by Gemini 3 Pro Image. The examples cover diverse objects and editing operations.

The source-view selection strategy depends on the available modalities. For T and $T + R$, Gemini 3.6 Flash selects the source view that is expected to provide the clearest evidence for the requested edit, considering the visibility of the target-relevant region, occlusion, local attachment or context, and preservation of unrelated geometry. For $T + B$, $B + R$, and $T + B + R$, the Bounding Box is already associated with a particular source view during multimodal condition construction. We therefore directly recover and lock this corresponding view rather than performing semantic view reselection.

Given the unified editing description E and the selected source view, we construct a modality-specific image-editing prompt and use Gemini 3 Pro Image to generate the target view I_t. The selected source view always serves as the editing canvas, whereas Text, Bounding Box, and Reference Image provide complementary editing guidance. Text primarily specifies the requested operation and target semantics; the Bounding Box provides only a soft spatial cue for the relevant source region; and the Reference Image provides the desired target identity, shape, part, color, material, or texture when applicable. For $B + R$, no user-provided Text condition, original textual instruction, or edit-type label is supplied; the required operation is inferred from the available visual and spatial cues and represented internally by E. In particular, the Bounding Box is explicitly treated as an approximate semantic localization cue rather than a pixel-level mask, exact editing boundary, crop, or target silhouette. Similarly, visual information from the Reference Image is adapted to the

viewpoint, scale, pose, lighting, and structural context of the source image rather than being directly copied.

Across all input settings, the generation prompt consistently requires preservation of unrelated object identity, unedited geometry and proportions, viewpoint, orientation, framing, lighting, shadows, and the original clean white background. The modality-specific prompt construction is summarized in Table 7.

Although no explicit task-category label is provided, the unified editing description E induces operation-specific constraints according to the editing intent inferred from the current multimodal condition. For an inferred addition operation, the requested content is inserted with appropriate scale and natural physical connectivity. For an inferred deletion operation, the target content is completely removed and the newly exposed surface is naturally reconstructed. For a local modification, only the relevant source part is modified while maintaining coherent geometry and intended connections. For an appearance change, the requested visual attributes are modified while preserving the geometry, size, pose, and placement of the corresponding part. For a pose change, the relevant articulation or part configuration is changed while preserving object identity and plausible structural connectivity. These constraints are determined from the inferred editing semantics rather than selected using an externally provided task label.

Figure 6 presents additional examples of target-view generation. Each pair shows the selected source view and the corresponding target view generated by Gemini 3 Pro Image. The examples cover diverse objects and editing operations and illustrate that the generated target views primarily modify the requested content while preserving the viewpoint, overall structure, appearance style, and unrelated regions of the source image.

A.3 Multimodal Condition–Task Coverage

UMAS3D supports five multimodal input combinations: T, $T+B$, $T+R$, $B+R$, and $T+B+R$. In our experimental setting, Addition, Modification, and Appearance cover all five combinations, whereas Deletion and Pose use T and $T+B$. This configuration is not obtained by mechanically applying every modality to every task, but is designed according to the information provided by each modality for different editing intents. Text primarily specifies the editing operation and semantics, the Bounding Box provides spatial localization of the edit-relevant source region, and the Reference Image provides complementary visual information about the target object, part, or appearance.

For **Addition**, **Modification**, and **Appearance**, the Reference Image provides explicit target-side visual information. For Addition, R can describe the identity, shape, or appearance of the object to be added. For Modification, R provides a visual reference for the desired target part or structure. For Appearance, R can specify target visual attributes such as color, material, texture, or pattern. These tasks therefore include the reference-based settings $T+R$, $B+R$, and $T+B+R$. Under $B+R$, no additional Text condition or task-category label is supplied; the editing semantics are inferred jointly from the source view, the spatial information provided by the Bounding Box, and the target visual information conveyed by the Reference Image.

For **Deletion** and **Pose**, we do not construct Reference-Image-based conditions and instead use T and $T+B$. Deletion is defined by removing existing source content and therefore primarily requires specifying what should be removed and where it is located. Pose changes the articulation, pose, or spatial configuration of existing parts, with the condition primarily describing motion semantics and spatial relations. Based on this condition design, we use Reference Images for editing scenarios that require explicit visual specification of newly added, replaced, or target appearance content, while not constructing Reference-Image-based conditions for Deletion and Pose. This setting also reflects how different editing intents can be expressed through complementary input modalities in practical editing scenarios.

Despite these task-dependent input combinations, the downstream processing remains unified. All available conditions are first converted into target-view guidance through the multimodal interface described in Section A.2. Once the target view is obtained, all five editing tasks share the same downstream 3D editing process without task-specific 3D editing pipelines. As illustrated in Figure 7, UMAS3D therefore achieves unification at two complementary levels: heterogeneous multimodal

conditions are converted into target-view guidance at the *condition level*, while the five editing tasks share the same downstream 3D editing process at the *task level*.

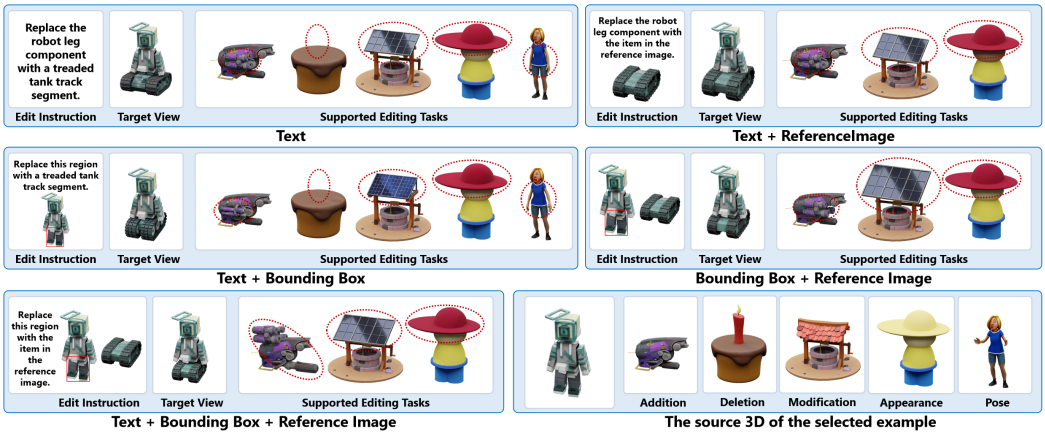

Figure 7: **Unified 3D editing across multimodal instruction combinations and editing tasks.** UMAS3D converts heterogeneous multimodal conditions into target-view guidance and handles Addition, Deletion, Modification, Appearance and Pose through the same 3D editing pipeline.

A.4 Visualization of MARM Initialization and Adaptive Region Refinement

UMAS3D does not require any externally provided or manually annotated 3D editing mask. Given the source 3D asset, its multi-view observations, and the generated target image, MARM first compares patch-level DINOv2 correspondences between the selected source view and the target image at the same and neighboring locations, together with foreground, appearance, and edge differences, to obtain 2D Edit/Keep evidence. Target-only content without reliable source correspondence provides additional target-side Edit evidence for opening editable space in source-empty regions.

To construct complete 3D regions, we do not directly back-project a single 2D mask. Instead, source-occupied Grid64 voxels are projected into the four source views and aggregate visible features from the frozen DINOv2 encoder. These voxel features are grouped according to DINOv2 feature similarity, 3D spatial distance, and connectivity to form complete 3D macro regions. Each 3D region is then projected to the selected-view Edit/Keep evidence map, where region-level evidence aggregation determines its initial state, yielding $\mathbf{M}_0 = (\mathbf{M}_0^E, \mathbf{M}_0^K)$.

The formal MARM partition contains only Edit and Keep states. Unknown and Target-New are used only as intermediate initialization evidence: Unknown denotes insufficient source-side evidence before region-level assignment, whereas Target-New denotes target-side content without reliable source correspondence and is ultimately incorporated into the editable support. They therefore do not introduce additional MARM region states.

Importantly, $\mathbf{M}_0$ serves only as the initial region prior rather than a fixed editing boundary. After an initial warm-up period, MARM periodically re-evaluates the current 3D regions according to the evolving target structure. A region is reassigned only when persistent evidence supports a transition. A Keep region can be released to Edit when its target-driven structural deviation remains consistently larger than that of comparable preserved regions. Conversely, an Edit region can be reassigned to Keep when its state remains consistently source-like. SGPS always follows the current region partition, so its source-preservation support evolves together with MARM refinement.

Figure 8 directly visualizes this adaptive process. For each example, we show the editing instruction, selected source view, generated target view, coarse 3D region prior, initial Edit/Keep partition $\mathbf{M}_0$, and final partition $\mathbf{M}_T$ obtained after Stage 1 refinement. Differences between $\mathbf{M}_0$ and $\mathbf{M}_T$ provide process-level evidence that the Edit/Keep support is actively reassigned during generation rather than remaining fixed after initialization.

At the end of Stage 1, the refined partition $\mathbf{M}_T$ is frozen and passed to Stage 2. No further MARM region reassignment is performed in Stage 2; instead, the final Keep support $\mathbf{M}_T^K$ is directly used by

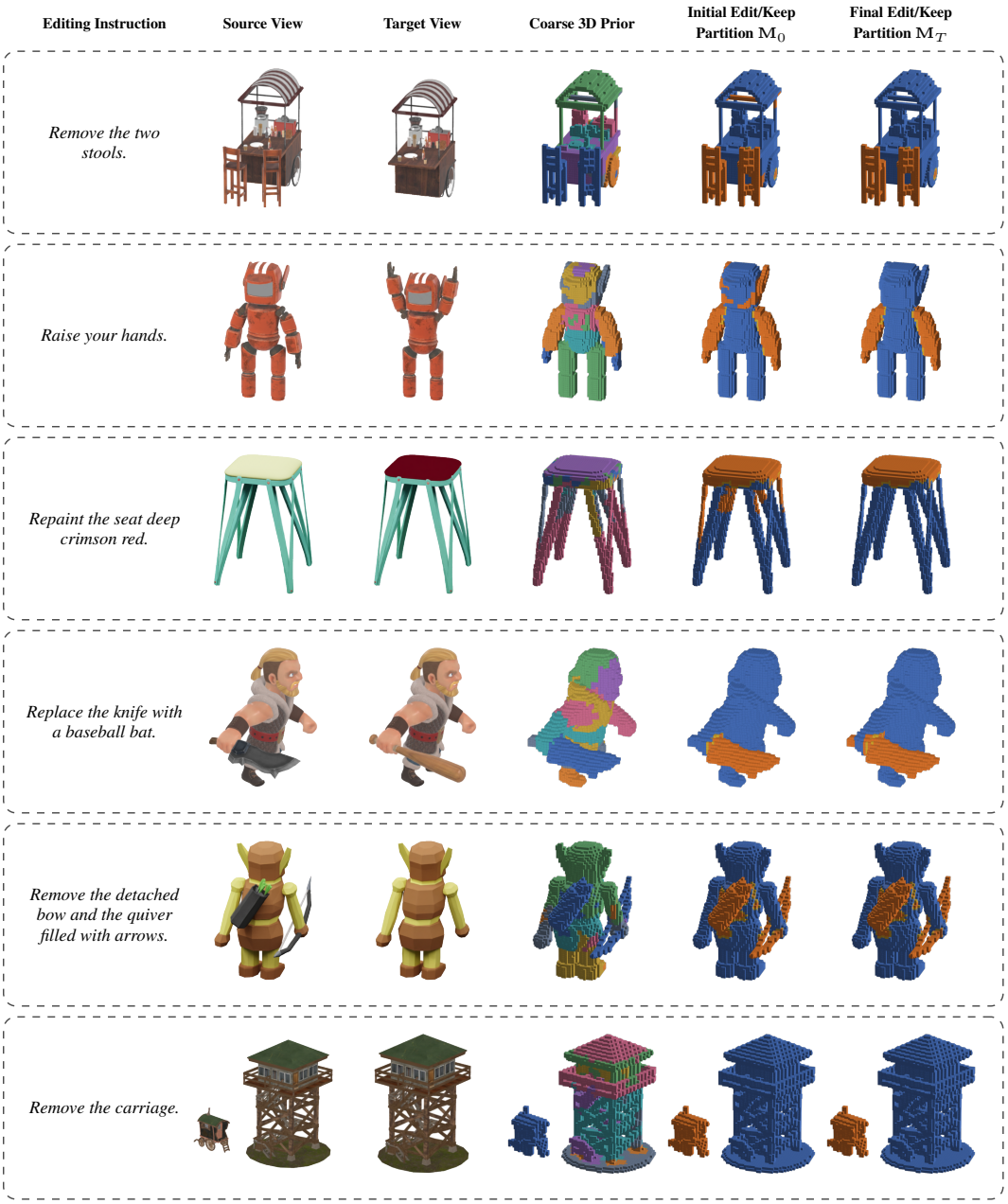

Figure 8: **Visualization of MARM initialization and adaptive region refinement.** Each row shows the editing instruction, selected source view, generated target view, coarse 3D region prior, initial Edit/Keep partition $\mathbf{M}_0$, and final partition $\mathbf{M}_T$ after Stage 1 refinement. Differences between $\mathbf{M}_0$ and $\mathbf{M}_T$ directly visualize the online reassignment of Edit/Keep regions within the single Stage 1 trajectory.

SGPS for source-feature preservation in SLAT space. Thus, Figure 8 complements the quantitative ablation in the main paper by explicitly visualizing how the initial partition evolves within the single Stage 1 generation trajectory.

A.5 Model and Inference Settings

Training-Free Setting. UMAS3D is fully training-free: all pretrained model parameters remain frozen throughout inference, with no additional training, fine-tuning, or test-time parameter adap-

tation. The optimization steps in SGPS act only on the current generation state and never update pretrained model parameters. TRELLIS Stage 1 and Stage 2 use 100 and 25 inference steps, respectively, with the same configuration for all editing tasks.

MARM Initialization Details. To complement Section 3.4, the selected source image and target image are resized to 518×518 and encoded by the same frozen DINOv2 frontend used by TRELLIS, yielding a 37×37 patch grid. MARM combines foreground presence, DINOv2 correspondence, Lab appearance difference, and edge difference into continuous Keep and Edit evidence. Target-only content additionally produces Target-New evidence for opening editable space in source-empty regions.

A source-foreground patch provides reliable Keep evidence when its Keep score is at least 0.55 and exceeds its Edit score by at least 0.08, and reliable Edit evidence when its Edit score is at least 0.55 and exceeds its Keep score by at least 0.10. Otherwise, the patch is marked Unknown at the evidence stage. Unknown and Target-New are internal initialization labels only: Unknown denotes insufficient evidence prior to region-level assignment, whereas Target-New denotes target-side content without reliable source correspondence. They do not introduce additional MARM region states; the formal MARM partition remains binary Edit/Keep.

The complete 3D region partition is constructed from the four source views rather than directly from single-target-image clustering. Source-occupied Grid64 voxels aggregate visible DINOv2 features from the four rendered source views. We first construct 16 provisional macro regions using deterministic clustering over DINO cosine distance and 3D spatial distance, followed by 3D-connectivity cleanup, and consolidate them into $K = 8$ adaptive regions. Each adaptive region is projected to the selected-view Edit/Keep evidence map, and region-level evidence aggregation determines the binary initial partition $\mathbf{M}_0$.

For target-only content, Target-New evidence is grouped into 2D candidate regions and projected through the calibrated selected-view camera into source-empty Grid64 candidate space. This candidate space permits new structure to emerge but does not directly enforce occupancy. Source-empty locations that are explicitly retained as non-editable support form the valid empty set $\mathbf{S}_e$ used by the Stage 1 SGPS empty-preservation branch.

Inference Parameters. The main inference parameters are summarized in Table 8. All settings are fixed across datasets and editing tasks.

Table 8: Main inference parameters of UMAS3D.

Component	Parameter	Value
TRELLIS	Stage 1 / Stage 2 steps	$100/25$
MARM	Adaptive regions K	8
	Provisional macro regions	16
	EMA coefficient λ	0.8
	Initial warm-up	first 35 Stage 1 steps
	Region update interval	every 5 steps
	Confirmation window / requirement	$3/2$
	Edit→Keep margin τ^K	0.04
	Keep→Edit margin τ^E	0.15
Stage 1 SGPS	Sample optimization steps / branch	1
	Base sample learning rate	1×10^{-5}
	Occupied / Empty update scale	$0.25/1.0$
	Empty constraint activation	$t \leq 0.5$
Stage 2 SGPS	Sample optimization steps	1
	Sample learning rate	1×10^{-5}
	SmoothL1 β	1.0

During the first 35 Stage 1 steps, the initial partition $\mathbf{M}_0$ is kept fixed to allow the target structure to establish a stable trajectory. Thereafter, MARM evaluates region states every five inference steps. To suppress transient switching, a transition is confirmed only when consistent evidence is observed in at least two of the most recent three refinement checkpoints. We use a larger margin for Keep→Edit transitions ($\tau^E = 0.15$) than for Edit→Keep transitions ($\tau^K = 0.04$), making the release of preserved source regions more conservative. The final partition obtained at the end of Stage 1 is frozen as $\mathbf{M}_T$, and its Keep support $\mathbf{M}_T^K$ is passed to Stage 2.

Sample-State Update and Sampling Order. Section 3.5 defines the Stage 1 objectives $\mathcal{L}_1^o$ and $\mathcal{L}_1^e$ and the Stage 2 objective $\mathcal{L}_2$. Here, we specify how these objectives are applied within the TRELLIS sampler.

Let $\mathbf{x}$ denote the current Stage 1 structure sample state and $\mathbf{Z}$ the current Stage 2 SLAT sample state. $\mathcal{T}_1$ and $\mathcal{T}_2$ denote the ordered sets of consecutive sampling-time pairs (t, t') in Stage 1 and Stage 2, respectively. $\mathcal{U}_{\mathrm{MARM}}$ denotes the Stage 1 refinement checkpoints after the initial warm-up. $\mathbf{M}$ denotes the current Edit/Keep partition and $\mathbf{M}^K$ its current Keep support.

We use $\hat{\mathbf{x}}_0$ and $\hat{\mathbf{Z}}_0$ to denote the predicted clean Structure and SLAT states. Superscript a denotes a source-aligned sample state, while $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{Z}}$ denote the SGPS-corrected states before frozen-flow re-evaluation. $\mathbf{G}_o$ and $\mathbf{G}_e$ denote the localized supports of the occupied and empty Stage 1 preservation branches. $\mathcal{A}_T$ denotes the final active voxel coordinate set produced by Stage 1.

The final Keep support $\mathbf{M}_T^K$ is mapped to the fixed Stage 2 Keep-token support $\mathbf{G}_K$ by exact sparse integer-coordinate correspondence. A narrow Boundary-token support $\mathbf{G}_B$ is additionally derived around the final Edit/Keep interface. $\mathbf{G}_B$ is used only for soft source–target blending and does not constitute an additional MARM state. Finally, $\boldsymbol{\epsilon}_{\mathrm{init}}$ denotes the fixed initial Stage 2 noise used for source-aligned Keep-token construction.

In Stage 1, source alignment first produces $\mathbf{x}^a$. The occupied and empty preservation branches then perform two independent Adam updates, both initialized from the same $\mathbf{x}^a$:

$$\Delta\mathbf{x}^o = \mathrm{AdamStep}(\mathbf{x}^a, \mathcal{L}_1^o, \eta\alpha_o) - \mathbf{x}^a, \qquad \Delta\mathbf{x}^e = \mathrm{AdamStep}(\mathbf{x}^a, \mathcal{L}_1^e, \eta\alpha_e) - \mathbf{x}^a. \tag{16}$$

The corrected state is

$$\tilde{\mathbf{x}} = \mathbf{x}^a + \mathbf{G}_o \odot \Delta\mathbf{x}^o + \mathbf{G}_e \odot \Delta\mathbf{x}^e, \tag{17}$$

where $\eta = 10^{-5}$, $\alpha_o = 0.25$, and $\alpha_e = 1.0$. Thus, the two BCE branches produce independent localized sample-state corrections from the same source-aligned state. The frozen TRELLIS Structure Flow is then re-evaluated on $\tilde{\mathbf{x}}$, followed by the original Euler sampling update.

Stage 2 follows the same sample-state correction principle. The source-aligned state $\mathbf{Z}^a$ is constructed using the source SLAT features and fixed initial-noise endpoint in $\mathbf{G}_K$, while tokens in $\mathbf{G}_B$ are softly blended between source-aligned and target-driven states. All remaining tokens stay target-driven. The SmoothL1 source supervision is applied only to the Keep-token support:

$$\Delta\mathbf{Z} = \mathrm{AdamStep}(\mathbf{Z}^a, \mathcal{L}_2, \eta) - \mathbf{Z}^a, \qquad \tilde{\mathbf{Z}} = \mathbf{Z}^a + \mathbf{G}_K \odot \Delta\mathbf{Z}. \tag{18}$$

The frozen SLAT Flow is subsequently re-evaluated on $\tilde{\mathbf{Z}}$, followed by the standard Euler update over all tokens. Boundary and Edit tokens receive no SmoothL1 source-feature gradient, although they continue to evolve through the normal target-conditioned SLAT Flow. A final exact Source-SLAT composition is applied only to the Keep tokens before decoding.

Algorithm 1 Inference Procedure of UMAS3D

Require: Source 3D $\mathbf{X}_s$; four source views and calibrated cameras $\{\mathbf{I}_s^v, \Pi_v\}_{v=1}^V$ with $V = 4$; selected source image $\mathbf{I}_s^{v^*}$; target image $\mathbf{I}_t$; frozen TRELLIS
Ensure: Edited 3D $\mathbf{X}_e$

$\mathbf{M}_0 \leftarrow \textsc{InitializeMARM}(\mathbf{X}_s, \{\mathbf{I}_s^v, \Pi_v\}_{v=1}^V, \mathbf{I}_s^{v^*}, \mathbf{I}_t)$
$\mathbf{M} \leftarrow \mathbf{M}_0$
$\mathbf{x} \leftarrow \textsc{InitStructureState}$
for $(t, t') \in \mathcal{T}_1$ **do**
  $\hat{\mathbf{x}}_0 \leftarrow \textsc{StructurePredict}(\mathbf{x}, t, \mathbf{I}_t)$
  if $t \in \mathcal{U}_{\text{MARM}}$ **then**
    $\mathbf{M} \leftarrow \textsc{MARMUpdate}(\mathbf{M}, \hat{\mathbf{x}}_0, \mathbf{X}_s)$
  end if
  $\mathbf{x}^a \leftarrow \textsc{SourceAlign}(\mathbf{x}, \hat{\mathbf{x}}_0, \mathbf{X}_s, \mathbf{M}^K)$
  $(\mathcal{L}_1^o, \mathcal{L}_1^e, \mathbf{G}_o, \mathbf{G}_e) \leftarrow \textsc{Stage1Loss}(\mathbf{x}^a, \mathbf{X}_s, \mathbf{M}^K)$
  $\Delta\mathbf{x}^o \leftarrow \textsc{AdamStep}(\mathbf{x}^a, \mathcal{L}_1^o, \eta\alpha_o) - \mathbf{x}^a$
  $\Delta\mathbf{x}^e \leftarrow \textsc{AdamStep}(\mathbf{x}^a, \mathcal{L}_1^e, \eta\alpha_e) - \mathbf{x}^a$
  $\tilde{\mathbf{x}} \leftarrow \mathbf{x}^a + \mathbf{G}_o \odot \Delta\mathbf{x}^o + \mathbf{G}_e \odot \Delta\mathbf{x}^e$
  $\mathbf{v} \leftarrow \textsc{StructureFlow}(\tilde{\mathbf{x}}, t, \mathbf{I}_t)$
  $\mathbf{x} \leftarrow \tilde{\mathbf{x}} - (t - t')\mathbf{v}$
end for
$\mathbf{M}_T \leftarrow \textsc{Freeze}(\mathbf{M})$
$\mathcal{A}_T \leftarrow \textsc{Stage1Finalize}(\mathbf{x}, \mathbf{X}_s, \mathbf{M}_T^K)$
$\mathbf{Z}_s \leftarrow \textsc{EncodeSourceSLAT}(\mathbf{X}_s)$
$(\mathbf{Z}, \boldsymbol{\epsilon}_{\text{init}}) \leftarrow \textsc{InitSLATState}(\mathcal{A}_T)$
$\mathbf{G}_K \leftarrow \textsc{KeepTokenSupport}(\mathbf{M}_T^K, \mathcal{A}_T)$
$\mathbf{G}_B \leftarrow \textsc{BoundaryTokenSupport}(\mathbf{M}_T, \mathcal{A}_T)$
for $(t, t') \in \mathcal{T}_2$ **do**
  $\hat{\mathbf{Z}}_0 \leftarrow \textsc{SLATPredict}(\mathbf{Z}, t, \mathbf{I}_t)$
  $\mathbf{Z}^a \leftarrow \textsc{SourceAlignSLAT}(\mathbf{Z}, \hat{\mathbf{Z}}_0, \mathbf{Z}_s, \boldsymbol{\epsilon}_{\text{init}}, \mathbf{G}_K, \mathbf{G}_B)$
  $\mathcal{L}_2 \leftarrow \textsc{KeepSmoothL1}(\mathbf{Z}^a, \mathbf{Z}_s, \mathbf{G}_K)$
  $\Delta\mathbf{Z} \leftarrow \textsc{AdamStep}(\mathbf{Z}^a, \mathcal{L}_2, \eta) - \mathbf{Z}^a$
  $\tilde{\mathbf{Z}} \leftarrow \mathbf{Z}^a + \mathbf{G}_K \odot \Delta\mathbf{Z}$
  $\mathbf{v} \leftarrow \textsc{SLATFlow}(\tilde{\mathbf{Z}}, t, \mathbf{I}_t)$
  $\mathbf{Z} \leftarrow \tilde{\mathbf{Z}} - (t - t')\mathbf{v}$
end for
$\mathbf{Z}_e \leftarrow \textsc{SourceCompose}(\mathbf{Z}, \mathbf{Z}_s, \mathbf{G}_K)$
$\mathbf{X}_e \leftarrow \textsc{Decode}(\mathbf{Z}_e)$
return $\mathbf{X}_e$

The entire procedure uses a single Stage 1 trajectory followed by a single Stage 2 trajectory, without an additional TRELLIS proposal pass.

Computational Cost. On 50 samples, with 10 samples from each editing task, using an NVIDIA A100-SXM4-40GB GPU and batch size 1, UMAS3D requires 245.80 s per sample on average and 11.55 GiB peak allocated GPU memory (16.93 GiB peak reserved memory). Runtime is measured from target-image preprocessing to final edited 3D GLB export, including mask-free region discovery, Stage 1 and Stage 2 generation, final decoding, and mesh/texture/GLB export, while excluding model and source-SLAT loading, four-view rendering, and GT evaluation.

A.6 Evaluation Metrics

We evaluate the final edited 3D results from two complementary aspects: *Unedited-region Preservation*, using Pres-IoU, Pres-PSNR, and Pres-SSIM, and *Overall Target Quality*, using DINO-I, CD-L1, F@1%, and NC.

Unedited-region Preservation. **Pres-IoU.** Let $\mathcal{O}_s$ and $\hat{\mathcal{O}}$ denote the surface-occupancy supports of the source asset $\mathbf{X}_s$ and prediction $\hat{\mathbf{X}}$ on a common 64^3 Grid64. Let $\mathbf{M}_{\text{gt}}^K$ denote the evaluation Keep region derived from the aligned source and benchmark target. The preserved surface supports are

$$\mathcal{O}_s^K = \mathcal{O}_s \cap \mathbf{M}_{\text{gt}}^K, \qquad \hat{\mathcal{O}}^K = \hat{\mathcal{O}} \cap \mathbf{M}_{\text{gt}}^K. \tag{19}$$

Pres-IoU is defined as

$$\text{Pres-IoU} = \frac{|\mathcal{O}_s^K \cap \hat{\mathcal{O}}^K|}{|\mathcal{O}_s^K \cup \hat{\mathcal{O}}^K|}. \tag{20}$$

A higher value indicates better preservation of source geometry in the unedited region.

Pres-PSNR. For canonical view v, let $\mathbf{I}_s^v$ and $\hat{\mathbf{I}}^v$ denote the source and predicted renderings, respectively, and let $\mathbf{M}_{\text{gt}}^{K,v}$ denote the corresponding evaluation Keep mask. Images are normalized to $[0, 1]$. The Keep-region MSE is

$$\text{MSE}_K^v = \frac{1}{3|\mathbf{M}_{\text{gt}}^{K,v}|} \sum_{p \in \mathbf{M}_{\text{gt}}^{K,v}} \sum_{c=1}^{3} \left(\mathbf{I}_{s,c}^v(p) - \hat{\mathbf{I}}_c^v(p) \right)^2, \tag{21}$$

and

$$\text{PSNR}_K^v = 10 \log_{10} \left(\frac{1}{\max(\text{MSE}_K^v, \epsilon)} \right), \qquad \epsilon = 10^{-12}. \tag{22}$$

Let

$$\mathcal{V}_K = \left\{ v \mid \mathbf{M}_{\text{gt}}^{K,v} \neq \varnothing \right\} \tag{23}$$

denote the valid Keep views. We compute

$$\text{Pres-PSNR} = \frac{1}{|\mathcal{V}_K|} \sum_{v \in \mathcal{V}_K} \text{PSNR}_K^v. \tag{24}$$

Pres-SSIM. We first compute the standard SSIM map between the complete source and predicted views:

$$\mathbf{S}^v(p) = \text{SSIMMap} \left(\mathbf{I}_s^v, \hat{\mathbf{I}}^v \right)_p, \tag{25}$$

where the standard constants for $[0, 1]$ images are

$$C_1 = (0.01)^2, \qquad C_2 = (0.03)^2. \tag{26}$$

To reduce contamination from SSIM windows crossing the Edit/Keep boundary, the evaluation Keep mask is slightly eroded before averaging. Let $\widetilde{\mathbf{M}}_{\text{gt}}^{K,v}$ denote the eroded Keep mask. Then

$$\text{SSIM}_K^v = \frac{\sum_p \widetilde{\mathbf{M}}_{\text{gt}}^{K,v}(p) \mathbf{S}^v(p)}{\sum_p \widetilde{\mathbf{M}}_{\text{gt}}^{K,v}(p)}, \tag{27}$$

and

$$\text{Pres-SSIM} = \frac{1}{|\mathcal{V}_K|} \sum_{v \in \mathcal{V}_K} \text{SSIM}_K^v. \tag{28}$$

A higher Pres-SSIM indicates better structural and visual preservation in the unedited region.

Overall Target Quality. DINO-I. Let $\mathbf{I}_{\text{gt}}^v$ denote the benchmark-target rendering at canonical view v. Using the same frozen DINOv2 backbone ϕ, let $g_\phi(\cdot)$ denote its image-level representation. We compute

$$\text{DINO-I} = \frac{1}{V} \sum_{v=1}^{V} \cos \left(g_\phi(\hat{\mathbf{I}}^v), g_\phi(\mathbf{I}_{\text{gt}}^v) \right), \qquad V = 4. \tag{29}$$

This metric is evaluated over the complete rendered object. A higher value indicates stronger visual-semantic consistency with the benchmark target.

CD-L1. Let $\mathcal{Q}_{\hat{X}}$ and $\mathcal{Q}_{\text{gt}}$ denote surface point sets sampled from $\hat{\mathbf{X}}$ and $\mathbf{X}_{\text{gt}}$, respectively. We use the symmetric Chamfer distance

$$\text{CD-L1} = \frac{1}{2} \left[\frac{1}{|\mathcal{Q}_{\hat{X}}|} \sum_{\hat{q} \in \mathcal{Q}_{\hat{X}}} \min_{q \in \mathcal{Q}_{\text{gt}}} \|\hat{q} - q\|_2 + \frac{1}{|\mathcal{Q}_{\text{gt}}|} \sum_{q \in \mathcal{Q}_{\text{gt}}} \min_{\hat{q} \in \mathcal{Q}_{\hat{X}}} \|q - \hat{q}\|_2 \right]. \tag{30}$$

A lower value indicates better overall geometric consistency with the benchmark target.

F@1%. Let D_{gt} denote the 3D bounding-box diagonal of the benchmark target and define

$$\tau = 0.01 D_{\mathrm{gt}}. \tag{31}$$

The surface precision and recall are

$$\mathrm{Prec}_\tau = \frac{\left|\left\{\hat{q} \in \mathcal{Q}_{\hat{X}} : \min_{q \in \mathcal{Q}_{\mathrm{gt}}} \|\hat{q} - q\|_2 \le \tau\right\}\right|}{|\mathcal{Q}_{\hat{X}}|}, \tag{32}$$

and

$$\mathrm{Rec}_\tau = \frac{\left|\left\{q \in \mathcal{Q}_{\mathrm{gt}} : \min_{\hat{q} \in \mathcal{Q}_{\hat{X}}} \|q - \hat{q}\|_2 \le \tau\right\}\right|}{|\mathcal{Q}_{\mathrm{gt}}|}. \tag{33}$$

We then compute

$$\mathrm{F@1\%} = \frac{2\,\mathrm{Prec}_\tau \mathrm{Rec}_\tau}{\mathrm{Prec}_\tau + \mathrm{Rec}_\tau}. \tag{34}$$

NC. Normal Consistency measures agreement between the predicted and benchmark-target surface normals under bidirectional nearest-neighbor correspondence. Define

$$\pi_{\mathrm{gt}}(\hat{q}) = \arg\min_{q \in \mathcal{Q}_{\mathrm{gt}}} \|\hat{q} - q\|_2, \qquad \pi_{\mathrm{pred}}(q) = \arg\min_{\hat{q} \in \mathcal{Q}_{\hat{X}}} \|q - \hat{q}\|_2. \tag{35}$$

Let $\hat{\mathbf{n}}(\hat{q})$ and $\mathbf{n}_{\mathrm{gt}}(q)$ denote the corresponding unit normals. We compute

$$\mathrm{NC} = \frac{1}{2}\left[\frac{1}{|\mathcal{Q}_{\hat{X}}|}\sum_{\hat{q} \in \mathcal{Q}_{\hat{X}}} \left|\hat{\mathbf{n}}(\hat{q})^\top \mathbf{n}_{\mathrm{gt}}\big(\pi_{\mathrm{gt}}(\hat{q})\big)\right| + \frac{1}{|\mathcal{Q}_{\mathrm{gt}}|}\sum_{q \in \mathcal{Q}_{\mathrm{gt}}} \left|\mathbf{n}_{\mathrm{gt}}(q)^\top \hat{\mathbf{n}}\big(\pi_{\mathrm{pred}}(q)\big)\right|\right]. \tag{36}$$

A higher NC indicates better local surface-orientation consistency with the benchmark target.

The evaluation Keep regions $\mathbf{M}_{\mathrm{gt}}^{K}$ and $\mathbf{M}_{\mathrm{gt}}^{K,v}$ are used only for metric computation and are independent of the Edit/Keep partition estimated by MARM during inference. If a sample contains no valid Keep region for a preservation metric, that metric is excluded for the sample rather than assigned an artificial value. Dataset-level scores are therefore averaged independently over the valid samples of each metric.

A.7 Qualitative 3D Editing Results

To further demonstrate the editing capability of UMAS3D across different tasks and object categories, Figure 9 presents additional qualitative results covering Addition, Deletion, Modification, Appearance, and Pose. For each example, we show four corresponding views of the source 3D asset and the final edited 3D result, enabling direct inspection of both the requested modification and the preservation of unrelated regions.

The results show that UMAS3D can perform diverse structural and appearance edits while largely preserving the geometry and visual attributes of unrelated source regions. In addition to samples from Pxform and Omni3DEdit, we include several Objaverse assets to illustrate the applicability of the method to more diverse object categories. These additional samples are used only for qualitative visualization and are excluded from all quantitative evaluations.



Figure 9: Additional qualitative 3D editing results of UMAS3D. Each dashed box presents four views of the source 3D and four corresponding views of the edited 3D, separated by a vertical dashed line.