ROUTE, DIAGNOSE, REFINE: DISAGREEMENT-AWARE MULTI-VIEW EDITING FOR 3D ASSET COMPOSITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Composing existing 3D assets from language requires realizing the requested interaction while preserving source geometry, appearance, and identity wherever changes are unnecessary. Existing 3D editors typically rely on a single or fixed set of views, leaving contact, occluded, and unseen regions unresolved. We introduce an active multi-view framework that casts 3D composition as a closed-loop *route–diagnose–refine* process. Our key observation is that view-conditioned source–target differential velocities share the same native 3D action space, enabling both editing and conflict diagnosis. Geometry-derived support and priority-aware ownership route each view's flow to reliable regions, while cross-view disagreement localizes unresolved edits and a three-state memory preserves resolved content. The system then acquires an informative new view and refines only where needed. Our end-to-end pipeline requires no user-specified 3D edit mask. Our method improves CLIP-T by 10.5% and DINO-I by 11.4% on our custom benchmark, while reducing LPIPS by 41.5% relative to Easy3E.

1 INTRODUCTION

Many practical 3D workflows begin with assets that users want to preserve. A character may need to grasp a tool, wear an accessory, or ride a vehicle; multiple objects may need to be attached or assembled into a functional configuration. The goal is not to regenerate a new object that merely resembles the inputs, but to realize a new spatial or functional interaction while preserving each source asset wherever modification is unnecessary. Language-guided multi-asset composition is therefore a preservation-constrained editing problem: the system must make the changes needed to establish a plausible interaction, while retaining the geometry, texture, proportions, and identity of the original assets outside the affected regions.

Text- and image-conditioned 3D generators can synthesize high-quality individual objects from prompts or references (Poole et al., 2022; Lin et al., 2023; Hong et al., 2024; Li et al., 2024; Xiang et al., 2025). But they are designed mainly for single-object generation or reconstruction, not relational composition of multiple user-provided assets. Accordingly, regenerating the target often alters source geometry, texture, and proportions. Existing 3D editors provide stronger preservation and control (Haque et al., 2023; Zhuang et al., 2023; Sella et al., 2023; Chen et al., 2024; Kamata et al., 2023; Hu et al., 2026; Li et al., 2026), yet their guidance typically comes from a single or fixed set of views. This becomes limiting after coarse composition, where contact regions, occluded surfaces, back-side geometry, and thin structures may require view-dependent corrections, while unrelated regions must remain unchanged. Consequently, preserving source assets while achieving consistent multi-view refinement is a central challenge in language-guided 3D asset composition.

A natural solution is to render the composed asset, edit several renderings with strong 2D priors (Meng et al., 2022; Brooks et al., 2023; Zhang et al., 2023), and transfer their guidance back to 3D. Multiple views are necessary, but they are not automatically compatible. Each edited view supplies only partial evidence and is therefore informative only for some parts of the scene, while different views may prescribe conflicting local corrections around contact, occlusion, and silhouette boundaries. Symmetric averaging can attenuate a decisive edit or collapse incompatible directions into an unhelpful compromise; more fundamentally, no reweighting of the current views can recover evidence that none of them observes. The central challenge is therefore not simply to obtain or

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.

average more views, but to answer three coupled questions: *where should each view act, where does the current edit remain unresolved, and where should the system look next?*

Contributions. We address these questions with a closed-loop multi-view editor built on a simple observation: a flow-based 3D editor maps every edited view to a source–target differential velocity field in the same native 3D latent space. Unlike image-space signals from different cameras, these fields share a common operational meaning—they specify how each view proposes to change the same 3D location. This shared action space allows us not only to combine view-specific edits, but also to expose when the current views prescribe incompatible corrections. Our resulting *route–diagnose–refine* framework separates three roles that fixed-view fusion typically conflates.

Where should each view act? Different views are reliable over different parts of the object because of visibility, viewing angle, occlusion, and edit-mask boundaries. So we estimate geometry-derived confidence regions and use them to define local view ownership rather than as coefficients for symmetric velocity averaging. Each differential flow acts primarily where its view provides reliable spatial support; when ownership regions overlap, a newly acquired view receives local priority because it was selected specifically to repair a conflict identified in the preceding round. This priority-aware routing preserves decisive local edits instead of diluting them through indiscriminate averaging.

Where is the current edit still unresolved? Rather than averaging cross-view inconsistency away, we retain disagreement among the view-conditioned differential velocities as an internal diagnosis signal. Large disagreement means that the current views prescribe incompatible changes at a 3D location and that the edit remains unresolved. After each editing round, we further distinguish three states: unresolved voxels requiring more evidence, edited-stable voxels that should retain their current state, and quiescent voxels that should remain anchored to the pre-round state. This state-aware diagnosis prevents later refinement from repeatedly modifying regions that have been resolved.

Where should the system look next? For the remaining unresolved region, we select an additional camera that increases its visibility while avoiding redundant viewpoints, synthesize a consistent target view, and continue editing only there. The complete sparse structure and feature states are carried across rounds, so newly acquired evidence can repair residual geometry without decoding and reconstructing the object or disturbing resolved content. So the editor turns a fixed set of view conditions into a closed loop that actively acquires evidence only where the current views are insufficient.

Finally, we instantiate this framework in an end-to-end language-guided composition system which takes only the source 3D assets and a natural-language instruction, without requiring a user-specified 3D edit mask. On 194 multi-asset compositions, our method improves CLIP-T by 10.5% and DINO-I by 11.4%, while reducing LPIPS by 41.5% over Easy3E, and receives 59.4% preference in a 130-participant user study.

2 Related Work

3D Generation and Editing. Recent 3D generation has shifted from optimization with 2D diffusion priors to feed-forward native-3D diffusion and rectified-flow models (Poole et al., 2022; Hong et al., 2024; Tencent Hunyuan3D Team, 2025; Li et al., 2025b;a; Wu et al., 2025; Xiang et al., 2026). TRELLIS (Xiang et al., 2025) represents geometry and appearance with structured native-3D latents, enabling direct generation and editing. Meanwhile, 3D editing has progressed from per-scene optimization to multi-view inpainting and learned native-3D editors (Haque et al., 2023; Sella et al., 2023; Chen et al., 2024; Barda et al., 2024; Xu et al., 2026; Yin et al., 2026); VoxHammer (Li et al., 2026) further preserves unedited regions directly in TRELLIS latents. Our setting instead requires coordinating view-dependent edits while preserving the identities of multiple composed assets.

Flow-based and Multi-view 3D Editing. Recent systems automate different parts of iterative 3D editing: Vinedresser3D (Chi et al., 2026) combines agentic localization, informative-view selection, and native-3D editing; EditFlow3D (Nie et al., 2026) uses VLM-generated guidance and masks for local differential-flow editing; Interact3D (Shan et al., 2026) introduces collision-aware composition with VLM-guided refinement; and Thoughtful3D (Liu et al., 2026) iteratively corrects cross-view inconsistencies in 3D generation. Unlike these methods, we use disagreement among view-conditioned differential flows as an internal signal that localizes unresolved edits and actively determines where new view evidence is needed.

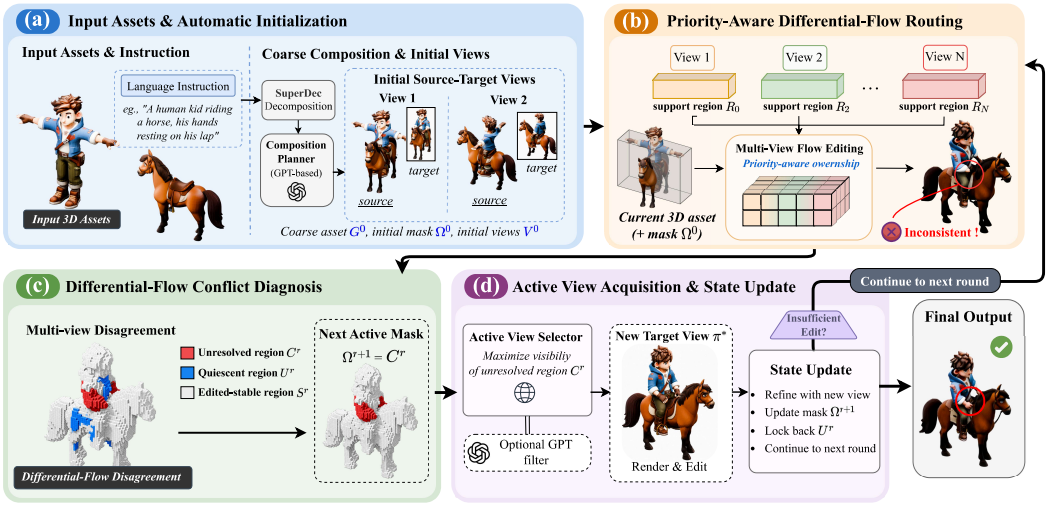

Figure 1: Our route–diagnose–refine framework. (a) Input and automatic initialization produce a coarse composition, initial edit mask, and initial views. (b) Priority-aware differential-flow routing assigns each view's flow to its supported 3D regions. (c) Differential-flow conflict diagnosis localizes unresolved regions from cross-view disagreement. (d) Active view acquisition selects a new informative view and updates the 3D state. The loop continues until a stopping criterion is met.

3 METHODOLOGY

Given source assets $\mathcal{A} = \{A_k\}_{k=1}^{K}$ and instruction y, we seek a composed asset that realizes the requested interaction while preserving content that need not change. Fig. 1(a) automatically initializes a coarse asset $\mathcal{G}^0$, edit mask Ω^0, and source–target views. Each round r routes view-conditioned differential flows to supported regions within Ω^r (b), diagnoses the unresolved region C^r from their disagreement (c), and acquires an informative view for localized refinement when $C^r \neq \emptyset$ (d). The updated views, mask, and native 3D state return to the next routing round, subject to the stopping criteria below.

3.1 PROBLEM SETUP AND AUTOMATIC COMPOSITION INITIALIZATION

Round-wise notation. At editing round r, $\Omega^r \subset \mathbb{R}^3$ is the active edit region and $\mathcal{V}^r = (\pi_1, \ldots, \pi_{N_r})$ is an ordered list of N_r calibrated cameras; π_i also denotes the associated 3D-to-image projection. Cameras are indexed by acquisition time, with π_1 as the primary/base view. For camera i, $I_i^{s,r}$ and $I_i^{t,r}$ are the current source rendering and its edited target. The native 3D state consists of a round-input sparse structure latent $\boldsymbol{x}_s^r$, a sparse latent feature (SLAT) field $\boldsymbol{z}_s^r$, and their coordinate set $\mathcal{P}^r$; $\boldsymbol{x}_t^r$ is the structure latent at noise time t. The cumulative source-preservation mask is M_{preserve}^r, initialized outside Ω^0. In voxelwise expressions, Ω^r also denotes its support on $\mathcal{P}^r$. We omit the round superscript when the round is clear.

Automatic coarse composition. As shown in Fig. 1(a), the source assets and instruction are first converted into a coarse composition and initial multi-view guidance. Fig. 2 shows the composition and refinement pathway: source objects, SuperDec decomposition, part-level planning, and the final refined asset. SuperDec (Fedele et al., 2025) first decomposes each source asset in Fig. 2 (a) into manipulable superquadrics in Fig. 2 (b). A GPT-based language-guided planner then receives the superquadric parameters and multi-view renderings of both the decompositions and original assets, providing explicit transformation handles together with semantic and appearance cues in Fig. 2 (c). Conditioned on y, it predicts part translations, rotations, and scales to construct a coarse arrangement. Applying these transformations to the source assets yields the coarse composition $\mathcal{G}^0$ and a loose and high-recall edit region Ω^0; subsequent native-3D refinement produces the final asset in Fig. 2 (d). We then render the initial camera set $\mathcal{V}^0$, and Step1X-Edit (Liu et al., 2025) generates the corresponding target views. For later cameras, target synthesis additionally conditions on the original primary-view source–target pair to preserve the same edit intent.

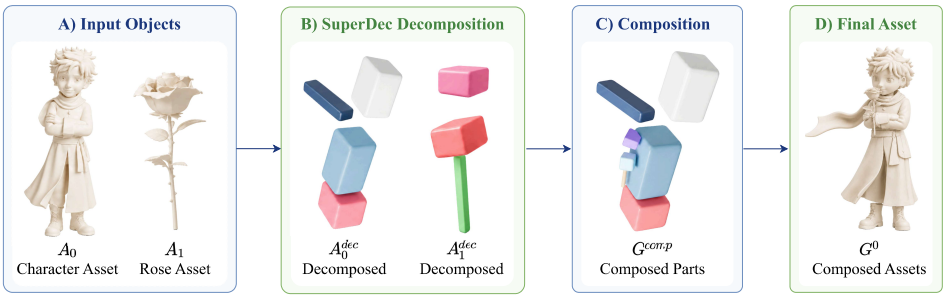

Figure 2: Automatic composition and refinement pathway. From left to right, the figure shows the input assets, their SuperDec decomposition, the coarse part arrangement predicted by the language-guided planner, and the composed asset after native-3D refinement.

3.2 Priority-Aware Differential-Flow Routing

Here we answer the first question: *where should each view act?* As visibility, viewing angle, occlusion, and mask boundaries make each edited view reliable only locally, we route its edit proposal to a geometry-supported region instead of symmetrically averaging all views. Fig. 1(b) illustrates this routing process, where view-specific differential flows are assigned to their supported 3D regions.

View-conditioned differential flows. We build on Voxel FlowEdit in Easy3E (Hu et al., 2026), which adapts inversion-free rectified-flow editing (Kulikov et al., 2025) to structured 3D latents. Let $\{t_m\}_{m=1}^{T}$ be the T denoising times and consider a step $t \to t' < t$. From the round-input structure latent $\boldsymbol{x}_s$ and the current latent $\boldsymbol{x}_t$ with noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, the source trajectory is

$$\tilde{\boldsymbol{x}}_t^s = (1-t)\boldsymbol{x}_s + \sigma(t)\boldsymbol{\epsilon}, \qquad \sigma(t) = \sigma_{\min} + (1-\sigma_{\min})t, \tag{1}$$

and the current edited latent is evaluated under the same source perturbation:

$$\tilde{\boldsymbol{x}}_t^e = \boldsymbol{x}_t + \big(\tilde{\boldsymbol{x}}_t^s - \boldsymbol{x}_s\big). \tag{2}$$

Here $\sigma_{\min}$ is the minimum noise scale. Let $v_\theta(\boldsymbol{x}, t, c)$ be the frozen image-conditioned rectified-flow network, which returns a velocity field over $\mathcal{P}^r$; $v_\theta(\cdot)(\boldsymbol{p})$ is its value at coordinate $\boldsymbol{p}$. Encoding the view pair (I_i^s, I_i^t) from Sec. 3.1 as (c_i^s, c_i^t) gives

$$\Delta \boldsymbol{v}_i(\boldsymbol{p}, t) = v_\theta(\tilde{\boldsymbol{x}}_t^e, t, c_i^t)(\boldsymbol{p}) - \lambda v_\theta(\tilde{\boldsymbol{x}}_t^s, t, c_i^s)(\boldsymbol{p}), \tag{3}$$

where $\lambda \geq 0$ controls the source reconstruction anchor. Following FlowEdit and Easy3E, the source-conditioned term anchors reconstruction, while their contrast provides an edit-oriented velocity. Crucially, every view expresses its proposed edit over the same 3D latent coordinates, enabling both local routing during the current round and direct cross-view comparison after it. We use the same formulation (3) for sparse structure and an analogous target-only velocity for SLAT repainting.

Geometry-derived view support. For each view, we estimate a soft 3D support field rather than requiring a user-specified trust region. For $\boldsymbol{p} \in \Omega^r$, let $\pi_i(\boldsymbol{p})$ denote its image projection. The raw support combines source-depth visibility, agreement with the instruction-induced 2D edit region, view-facing quality, and distance from the uncertain boundary of the active mask:

$$r_i(\boldsymbol{p}) = V_i(\boldsymbol{p})\, B_i(\pi_i(\boldsymbol{p}))\, \big(\max\big(\boldsymbol{n}(\boldsymbol{p})^\top \boldsymbol{d}_i(\boldsymbol{p}), 0\big) + \varepsilon\big)^\gamma\, \psi(d_{\Omega^r}(\boldsymbol{p}))\,. \tag{4}$$

Here $V_i \in [0,1]$ is a binary or soft z-buffer visibility term, $B_i \in [0,1]$ is the soft support of the instruction-induced 2D edit region, and $\boldsymbol{n}(\boldsymbol{p})$ is the local surface normal. For perspective camera i with center $\mathbf{c}_i$, the viewing direction is point-dependent, $\boldsymbol{d}_i(\boldsymbol{p}) = (\mathbf{c}_i - \boldsymbol{p})/\|\mathbf{c}_i - \boldsymbol{p}\|_2$, i.e., it points from voxel $\boldsymbol{p}$ toward the camera center. Finally, d_{Ω^r} is the signed distance to the mask boundary. The function $\psi : \mathbb{R} \to [0,1]$ increases toward the mask interior, preventing boundary voxels from being assigned to one view with excessive support. $\gamma > 0$ controls angular selectivity, and $\varepsilon > 0$ is a stabilizer.

To fill fragmented visible support without crossing strong boundaries, we solve

$$u_i = \arg\min_u \sum\nolimits_{\boldsymbol{p} \in \Omega^r} \big(u(\boldsymbol{p}) - r_i(\boldsymbol{p})\big)^2 + \mu \sum\nolimits_{(\boldsymbol{p},\boldsymbol{q}) \in \mathcal{E}^r} w_{\boldsymbol{pq}} \big(u(\boldsymbol{p}) - u(\boldsymbol{q})\big)^2, \tag{5}$$

where $\mathcal{E}^r$ contains neighboring active coordinates, edge-aware affinity $w_{\boldsymbol{pq}} \in [0,1]$ suppresses propagation across large geometric or mask-boundary changes, and $\mu \geq 0$ controls smoothness. Then with $\eta > 0$, the normalized support is

$$\chi_i(\boldsymbol{p}) = u_i(\boldsymbol{p}) / \big(\sum_{j=1}^{N_r} u_j(\boldsymbol{p}) + \eta\big). \tag{6}$$

These fields are recomputed at the beginning of every round from the current geometry and source/target renderings. Importantly, $\chi_i(\boldsymbol{p})$ represents spatial applicability rather than a calibrated probability that the corresponding velocity should be linearly averaged with other views. Manual trust regions are retained only as an optional diagnostic or oracle variant.

Priority-aware spatial ownership. We convert support fields $\{\chi_i\}$ into ownership weights $\{\alpha_i\}$ in acquisition order. We use π_1 as the base view and let later-acquired views take precedence only within their supported regions. The base view receives the residual ownership not claimed by views $2, \ldots, N_r$:

$$\alpha_1(\boldsymbol{p}) = \prod_{j=2}^{N_r} \big(1 - \chi_j(\boldsymbol{p})\big), \qquad \alpha_i(\boldsymbol{p}) = \chi_i(\boldsymbol{p}) \prod_{j=i+1}^{N_r} \big(1 - \chi_j(\boldsymbol{p})\big), \quad i = 2, \ldots, N_r. \tag{7}$$

Thus $\sum_i \alpha_i(\boldsymbol{p}) = 1$, with the base view receiving unclaimed support. A later view is a targeted intervention for a previously detected conflict, so it takes local precedence only where its support overlaps earlier guidance. The resulting routed differential velocity is

$$\Delta \boldsymbol{v}_{\text{route}}(\boldsymbol{p}, t) = \sum_{i=1}^{N_r} \alpha_i(\boldsymbol{p}) \Delta \boldsymbol{v}_i(\boldsymbol{p}, t). \tag{8}$$

Let M_{Ω^r} be the binary latent-resolution mask corresponding to Ω^r. A step $t \to t'$ updates only this region:

$$\boldsymbol{x}_{t'} = \boldsymbol{x}_t - (t - t') M_{\Omega^r} \odot \Delta \boldsymbol{v}_{\text{route}}(t), \tag{9}$$

where $\odot$ is element-wise masking. Following Easy3E, source lock-back anchors latent values outside the mask to the round input. After structure editing, generated coordinates inside Ω^r are merged with $\mathcal{P}^r$ outside it, and $\boldsymbol{z}$ is repainted on the merged set using the same ownership map. Features outside the active region are restored from the round-input SLAT.

3.3 Differential-Flow Conflict Diagnosis

Now we answer the second question: *where is the edit still unresolved?* Routing assigns local responsibility but cannot ensure compatible corrections. We therefore retain disagreement among the differential fields in Eq. (3) as a post-round conflict signal. Fig. 1(c) summarizes this diagnosis, while Fig. 3 shows how it progressively localizes unresolved regions across rounds.

Differential-flow disagreement. Let $N_r = |\mathcal{V}^r| \geq 2$ be the number of active views in round r, and $\{t_m\}_{m=1}^T$ the denoising steps. We quantify local cross-view conflict by averaging the pairwise discrepancy between the per-view differential fields in Eq. (3) over views and denoising steps:

$$D^r(\boldsymbol{p}) = \frac{1}{T} \sum_{m=1}^{T} \frac{2}{N_r(N_r - 1)} \sum_{1 \leq i < j \leq N_r} \left\| \Delta \boldsymbol{v}_i(\boldsymbol{p}, t_m) - \Delta \boldsymbol{v}_j(\boldsymbol{p}, t_m) \right\|_2 . \tag{10}$$

A large $D^r(\boldsymbol{p})$ indicates that the active views prescribe incompatible local corrections, making it an intrinsic conflict signal rather than a calibrated uncertainty probability. Low disagreement alone, however, does not imply that editing is complete: the views may agree on a coherent non-zero update. We therefore also measure the magnitude of the routed edit in Eq. (8):

$$E^r(\boldsymbol{p}) = \frac{1}{T} \sum_{m=1}^{T} \left\| \Delta \boldsymbol{v}_{\text{route}}(\boldsymbol{p}, t_m) \right\|_2 . \tag{11}$$

Thus, D^r measures *whether the views agree*, while E^r asks *whether the agreed edit remains active*.

Three-state post-round diagnosis. Diagnosis is performed only after the full denoising trajectory and therefore affects round $r+1$, never the completed round r. We combine relative statistics with absolute guards. The high-disagreement threshold is

$$\tau_C^r = \min\Big(\text{mean}_{\Omega^r}(D^r) + \text{std}_{\Omega^r}(D^r), \kappa_D^{\text{high}}\Big), \tag{12}$$

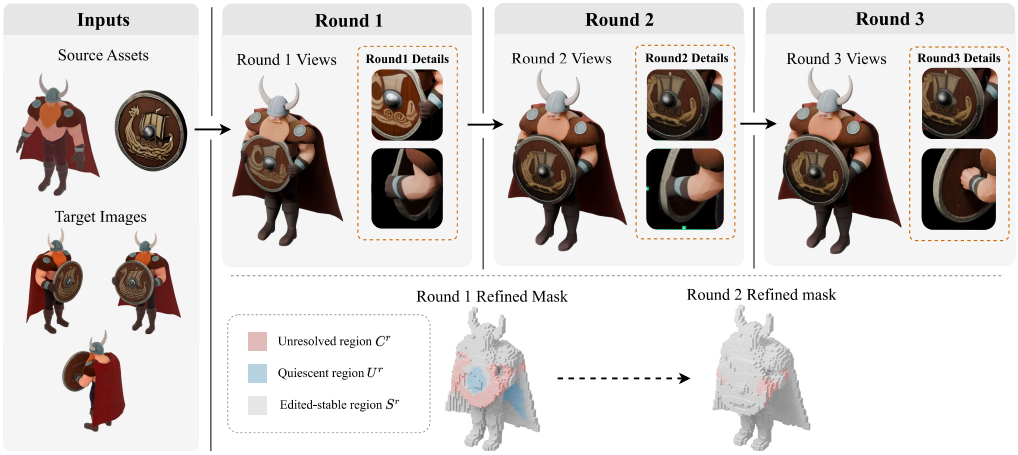

Figure 3: Conflict diagnosis and active refinement. After round r, differential- flow disagreement identifies C^r and hence Ω^{r+1}; resolved voxels enter the edited-stable state S^r or source-stable state U^r. The next camera $\pi^\star$ then focuses on the highlighted unresolved region, and the loop repeats without disturbing completed edits.

and the low-activity thresholds are

$$\tau_D^{\text{low},r} = \min\left(q_\ell(D^r), \kappa_D^{\text{low}}\right), \qquad \tau_E^{\text{low},r} = \min\left(q_\ell(E^r), \kappa_E^{\text{low}}\right), \tag{13}$$

where q_ℓ is a low quantile, set to 30% by default. The absolute guards κ prevent a globally difficult round from labeling a fixed fraction of its voxels stable solely, as their scores are relatively smaller.

A resolved voxel may be unchanged or may contain a completed edit; these cases must be preserved differently. We therefore partition Ω^r into three mutually exclusive states, including the conflict/unresolved set C^r, the unchanged/source-stable set U^r (voxels with low disagreement & edit activity),

$$C^r = \{\boldsymbol{p} \in \Omega^r : D^r(\boldsymbol{p}) > \tau_C^r\}, \quad U^r = \left\{\boldsymbol{p} \in \Omega^r \setminus C^r : D^r(\boldsymbol{p}) < \tau_D^{\text{low},r},\ E^r(\boldsymbol{p}) < \tau_E^{\text{low},r}\right\}, \tag{14}$$

and the edited-stable remainder S^r

$$S^r = \Omega^r \setminus \left(C^r \cup U^r\right). \tag{15}$$

So C^r continues editing, S^r retains its completed edit, and U^r reuses the round-input source anchor and joins M^r_{preserve} from Sec. 3.1. Only C^r defines the next active region. Then with morphological dilation for a safety margin and closing to remove holes,

$$\Omega^{r+1} = \Omega^r \cap \text{Close}(\text{Dilate}(C^r)) = \Omega^r \cap \text{Close}(\text{Dilate}(\Omega^r \setminus (S^r \cup U^r))). \tag{16}$$

This is the shrinking active mask visualized across rounds in Fig. 3.

3.4 Active View Acquisition and Stateful Refinement

Finally, we answer the third question: *where should the system look next?* Given the conflict set C^r in Eq. (14), we decide whether another view is needed and, if so, choose one that exposes the next-round mask Ω^{r+1} in Eq. (16). As illustrated in Fig. 1(d), the selected view provides new evidence for the unresolved region and the updated state is returned to the next routing round.

Stopping and next-view selection. We stop if $C^r = \varnothing$, if the unresolved fraction $|C^r|/|\Omega^r|$ falls to or below $\delta_{\text{stop}} \in [0, 1]$, or if the round budget $R_{\max}$ is exhausted. Otherwise, given a trade-off constant $\rho \geq 0$, from a discrete candidate camera set Π, we select

$$\pi^\star = \arg\max_{\pi \in \Pi} \ \text{Vis}_\pi(\Omega^{r+1}) - \rho \max_{1 \leq i \leq N_r} \text{Overlap}(\pi, \pi_i), \tag{17}$$

where Vis_π measures visibility of the unresolved mask, Overlap measures redundancy with an existing camera. An optional GPT filter removes visually redundant or clearly uninformative views from the top-ranked candidates; disabling it simply selects the best valid geometric candidate.

Table 1: End-to-end quantitative comparison on the 3D composition benchmark. A check mark indicates that a manually supplied 3D edit mask is required. All methods are evaluated over three reference views and averaged across them; ↑/↓ indicate that higher/lower values are better.

Method	Manual Mask	Condition Alignment CLIP-T ↑	DINO-I ↑	Paired-view Fidelity LPIPS ↓	PSNR ↑	SSIM ↑	Distribution Quality FID ↓
TRELLIS	✓	0.284	0.606	0.288	15.31	0.759	56.9
TRELLIS.2	✓	0.295	0.729	0.229	16.01	0.795	44.0
VoxHammer	✓	0.268	0.672	0.257	16.38	0.805	37.5
Instant3dit	×	0.198	0.523	0.438	12.89	0.709	79.2
Vox-E	×	0.234	0.670	0.521	14.11	0.692	103.0
Easy3E	✓	0.287	0.769	0.272	17.93	0.783	48.5
Ours	×	**0.317**	**0.857**	**0.159**	**21.98**	**0.904**	**27.1**

Target synthesis and state continuation. We render the current asset from $\pi^\star$ and synthesize its target with Step1X-Edit, conditioned additionally on the original primary-view source–target pair. We append the selected camera, $\mathcal{V}^{r+1} = (\pi_1, \ldots, \pi_{N_r}, \pi^\star)$. Through Eq. (7), the new targeted view receives local priority wherever its support overlaps earlier views. Let $(\boldsymbol{x}^r_{\text{edit}}, \boldsymbol{z}^r_{\text{edit}}, \mathcal{P}^r_{\text{edit}})$ be the routed output of round r. Rather than decode and re-encode the asset, the next round starts from

$$\boldsymbol{x}_s^{r+1} \leftarrow \boldsymbol{x}^r_{\text{edit}}, \qquad \boldsymbol{z}_s^{r+1} \leftarrow \boldsymbol{z}^r_{\text{edit}}, \qquad \mathcal{P}^{r+1} \leftarrow \mathcal{P}^r_{\text{edit}}. \tag{18}$$

Inside the new mask Ω^{r+1} in Eq. (16) the state remains editable; outside it, S^r in Eq. (15) retains its edited values and U^r reuses its round-input values. We update $M^{r+1}_{\text{preserve}} = M^r_{\text{preserve}} \cup U^r$ and recompute visibility, projections, normals, and the support fields χ_i in Eq. (6). The horizontal arrows in Fig. 3 visualize this carry-over: later rounds can repair geometry through $\boldsymbol{x}$, rather than only repaint appearance through $\boldsymbol{z}$, without disturbing resolved content.

4 Experiments

We evaluate language-guided composition on 194 cases built from Adobe Mixamo (Adobe, 2021) characters and PartObjaverse-Tiny (Yang et al., 2024) objects, covering interactions such as wearing, holding, attaching, and supporting. Given only the source assets and instruction, our system automatically generates the coarse composition, edit mask, and target views, starting from two views and adding at most one per refinement round. We compare against TRELLIS (Xiang et al., 2025), TRELLIS.2 (Xiang et al., 2026), Easy3E (Hu et al., 2026), VoxHammer (Li et al., 2026), Instant3dit (Barda et al., 2024), and Vox-E (Sella et al., 2023). Inputs are adapted to each method's interface, while all outputs use the same evaluation cameras. See additional benchmark, baseline, implementation, rendering, and runtime details in Appendix B.

Our evaluation has two levels. The end-to-end comparison evaluates complete composition systems under a common rendering protocol, while controlled studies fix the coarse source asset, target views, edit mask, denoising schedule, and inference budget to isolate priority-aware routing, disagreement-based diagnosis, and active view acquisition. We report CLIP-T (Radford et al., 2021), DINO-I (Oquab et al., 2023), LPIPS (Zhang et al., 2018), PSNR, SSIM (Wang et al., 2004), and FID (Heusel et al., 2017), together with a user study and targeted spatial analyses. Table 1 averages over three views; DINO-I, LPIPS, PSNR, and SSIM use the corresponding Step1X-Edit targets as references (Appendix B.3).

4.1 End-to-End Composition Quality

Quantitative comparison. Table 1 compares the final outputs under the common multi-view evaluation protocol. Our method performs best across all reported metrics. Relative to its Easy3E backbone, it improves CLIP-T by 10.5% and DINO-I by 11.4%, while reducing LPIPS by 41.5%. These gains indicate improved instruction alignment and fidelity to the edited image references.

Qualitative comparison. Fig. 4 compares representative results from identical rendering viewpoints. The baselines often improve the conditioned appearance but leave contact regions, rear surfaces, or source identity inconsistent, whereas our method preserves the supplied assets while producing more coherent interactions across views.

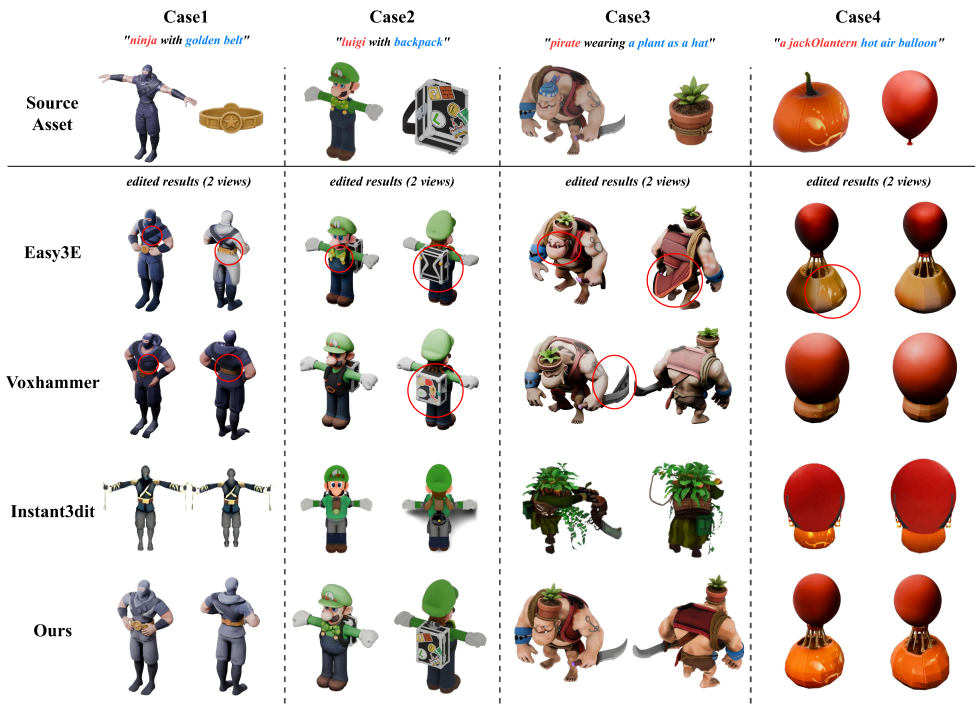

Figure 4: Qualitative comparison from common evaluation viewpoints. Our method better preserves source identity and resolves local interactions, particularly in contact and partially occluded regions.

Human evaluation. We additionally conduct a randomized user study with 130 participants. Each questionnaire randomly samples four benchmark cases. For every sampled case, participants are shown anonymized renderings from the compared methods in randomized order and are not told which rendering is produced by which method. They then select the rendering that best satisfies the composition instruction. Table 2 shows that our method receives 59.4% of valid preferences, substantially exceeding Easy3E (21.9%), VoxHammer (14.0%), and Instant3dit (4.7%).

Table 2: User preference (%) in the 130-participant study.

	Easy3E	VoxHammer	Instant3dit	**Ours**
Preference ↑	21.9	14.0	4.7	**59.4**

4.2 Ablation Study

The following controlled studies mirror the three questions that organize our method: where each view should act, where the current edit remains unresolved, and where the system should look next. Unless noted otherwise, all variants use the same coarse asset, target images, edit mask, random seeds, denoising steps, and total model calls.

Q1: Does priority-aware routing improve multi-view editing? We isolate the routing policy while holding the view set and all other settings fixed. We compare uniform fusion, average weighting and our priority-aware ownership. Fig. 5 shows representative cases in which symmetric fusion dilutes or compromises a decisive local edit, whereas priority-aware ownership retains the targeted correction. A supplementary quantitative comparison of Uniform, Average, and Priority-aware routing is reported in Appendix Table 7, using CLIP-T together with average and worst-view DINO-I and LPIPS.

Q2: Does differential-flow disagreement identify unresolved edits? On first-round outputs for all 194 assets, we compare native 3D predictions against shared per-asset references obtained by back-projecting multi-view error masks scored by Claude Sonnet 4.5 (Anthropic, 2025). All configurations use the same reference and evaluation settings. Table 3 compares D^r, routed-differential edit magnitude E^r, and visibility/support baselines; disagreement attains the highest AUPRC. The final column additionally reports Missed Edit Rate, the percentage of VLM-labeled voxels requir-

Table 3: First-round diagnosis on 194 assets against shared VLM-derived 3D references.

Signal	Spearman ↑	AUROC ↑	AUPRC ↑	Top-10% recall ↑	Missed Edit Rate (%) ↓
Visibility-based	0.0	0.49	0.27	0.01	98.5
Support-based	-0.1	0.46	0.26	0.05	89.2
Edit magnitude E^r	0.33	0.71	0.44	0.14	50.2
Disagreement D^r (ours)	**0.42**	**0.76**	**0.51**	**0.20**	**15.6**

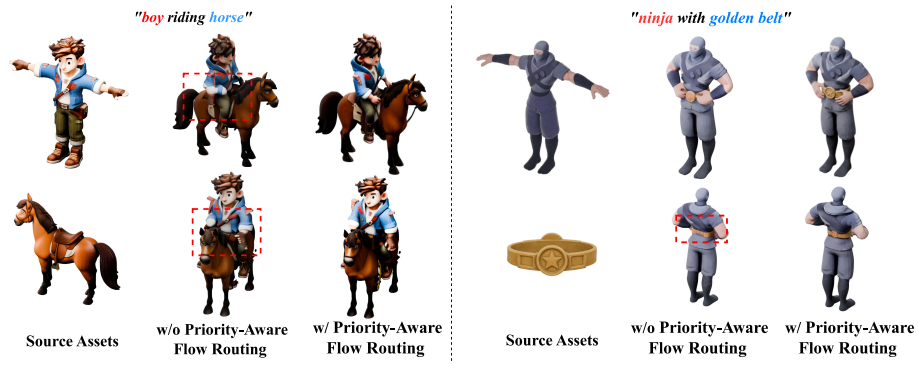

Figure 5: Comparison of multi-view routing policies under identical view set and edit budget.

Table 4: GPT-4 evaluation of next-view policies.

Policy	Highest-rated cases ↑	Average score ↑
Random	1	3.8
Farthest Camera	31	7.2
Object Coverage	27	7.5
Ours	**135**	**8.1**

Table 5: Preservation masks (50 assets) and round-wise rendered fidelity (Avg./Worst).

Round	Source preservation Prec. ↑	Rec. ↑	F1 ↑	PSNR ↑ Avg.	PSNR ↑ Worst	SSIM ↑ Avg.	SSIM ↑ Worst	LPIPS ↓ Avg.	LPIPS ↓ Worst
1	**1.00**	0.69	0.82	15.12	13.65	0.804	0.774	0.231	0.245
2	0.91	0.79	0.84	19.11	18.06	**0.903**	0.861	0.173	0.180
3	0.91	**0.80**	**0.85**	**19.36**	**19.19**	0.891	**0.874**	**0.165**	**0.172**

ing further editing that are assigned to the edited-stable state; our method achieves the lowest rate (15.6%). Appendix B.4 details the protocol.

Q3: Does unresolved-region coverage select more useful next views? We compare four next-view policies: random selection, the farthest camera, maximum whole-object coverage, and our unresolved-region coverage strategy. GPT-4 (OpenAI, 2023) evaluates anonymized outputs from the four policies for all 194 cases; we report highest-rated case counts and mean scores. Our policy ranks highest in 135 cases and achieves the best mean score (8.1; Table 4).

Round-wise progression. Table 5 evaluates source-preserved voxel sets against manually annotated 3D masks on 50 assets (Appendix B.5), alongside round-wise fidelity. These predictions denote source preservation, not the union of source-stable and edited-stable regions. Fidelity uses corresponding Step1X-Edit targets and the same sampled cameras, reporting view averages and worst-view scores (minimum PSNR/SSIM, maximum LPIPS). Table 4 separately isolates view selection.

5 Conclusion

We presented a language-guided framework for composing multiple 3D assets while preserving source identity and resolving view-dependent conflicts. Our *route–diagnose–refine* design uses priority-aware differential-flow routing, cross-view disagreement to localize unresolved edits, and active view acquisition to refine only where needed. Combined with automatic coarse composition, the system performs end-to-end multi-asset editing without user-specified 3D masks and consistently improves fidelity, preservation, and multi-view consistency over existing baselines.

AI USE STATEMENT

In this work, generative AI tools were used both for writing assistance and as components of the experimental pipeline. We used GPT models to generate free-form benchmark instructions, a GPT-based language-guided planner to predict coarse part transformations, and an optional GPT filter to remove visually redundant or clearly uninformative candidate views. GPT-4 was used to evaluate anonymized outputs in the next-view policy study, and Claude Sonnet 4.5 (`claude-sonnet-4-5`) was used to score residual editing errors and construct the VLM-derived reference masks used in the diagnosis evaluation. We also used generative AI tools for drafting assistance and iterative writing support. The authors take responsibility for the final methodology, experiments, claims, text, and artifacts produced with the aid of these tools.

ETHICS STATEMENT

Our work aims to assist 3D asset composition and editing. The benchmark uses publicly available or manually selected 3D assets and generated composition instructions. Our human evaluation was conducted through a low-risk questionnaire in which participants were shown randomized, anonymized renderings from different methods and asked only to select the output that best satisfies the corresponding composition instruction. Participation was voluntary, and participants were informed of the purpose and procedure of the questionnaire before taking part. The need for institutional ethics review was assessed according to our institution's applicable policy, and this type of anonymous, minimal-risk preference questionnaire was determined not to require formal ethics-board review. We did not collect sensitive personal information or personally identifying information, and only aggregate preference statistics are reported.

Potential misuse of our method includes creating misleading or unauthorized derivative 3D assets. We therefore emphasize that the method should be applied only to assets for which users have appropriate rights and permissions.

REPRODUCIBILITY STATEMENT

We describe benchmark construction, baseline adaptation, and evaluation metrics in the main paper and appendix. The method section specifies the view-conditioned latent editing objective, confidence-region construction, velocity fusion, disagreement-based mask filtering, and view-planning strategy. Appendix B.3 identifies the image references and view aggregation, and Appendix B.4 details residual-target construction, voxel assignment, label thresholds, and metric implementation. Additional implementation details, camera settings, inference-time breakdown, the coarse-composition planner prompt, and qualitative results are also provided in the appendix.

REFERENCES

Adobe. Mixamo. https://www.mixamo.com/, 2021. Adobe character and animation service.

Anthropic. Introducing claude sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5, 2025. Model identifier: `claude-sonnet-4-5`.

Amir Barda, Matheus Gadelha, Vladimir G. Kim, Noam Aigerman, Amit H. Bermano, and Thibault Groueix. Instant3dit: Multiview inpainting for fast editing of 3d objects. *arXiv preprint arXiv:2412.00518*, 2024.

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*, 2023.

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, 2024.

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.

Elisabetta Fedele, Boyang Sun, Leonidas Guibas, Marc Pollefeys, and Francis Engelmann. Superdec: 3d scene decomposition with superquadrics primitives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24625–24635, 2025.

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.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

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 *International Conference on Learning Representations*, 2024.

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*, pp. 12730–12740, 2026.

Hiromichi Kamata, Yuiko Sakuma, Akio Hayakawa, Masato Ishii, and Takuya Narihira. Instruct 3d-to-3d: Text instruction guided 3d-to-3d conversion. *arXiv preprint arXiv:2303.15780*, 2023.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, pp. 1097–1105, 2012.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19721–19730, 2025.

Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In *International Conference on Learning Representations*, 2024.

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 *International Conference on 3D Vision*, pp. 1281–1292, 2026. doi: 10.1109/3DV69130.2026.00126.

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

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. *arXiv preprint arXiv:2502.06608*, 2025b.

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.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, Guopeng Li, Yuang Peng, Quan Sun, Jingwei Wu, Yan Cai, Zheng Ge, Ranchen Ming, Lei Xia, Xianfang Zeng, Yibo Zhu, Binxing Jiao, Xiangyu Zhang, Gang Yu, and Daxin Jiang. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

Xinyue Liu, Jin Liu, Hongbo Wang, Ran He, and Huaibo Huang. Think-then-generate: Structural chain-of-thought reasoning for consistent 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34270–34280, 2026.

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 *International Conference on Learning Representations*, 2022.

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.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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 Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8748–8763, 2021.

Etai Sella, Gal Fiebelman, Peter Hedman, and Hadar Averbuch-Elor. Vox-e: Text-guided voxel editing of 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Hui Shan, Keyang Luo, Ming Li, Sizhe Zheng, Yanwei Fu, Zhen Chen, and Xiangru Huang. Interact3d: Compositional 3d generation of interactive objects. In *European Conference on Computer Vision*, 2026.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

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

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip H. S. Torr, and Yao Yao. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. In *Advances in Neural Information Processing Systems*, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21469–21480, 2025.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14419–14429, 2026.

Yizhao Xu, Hongyuan Zhu, Caiyun Liu, Tianfu Wang, Keyu Chen, Sicheng Xu, Jiaolong Yang, Nicholas Jing Yuan, and Qi Zhang. Beyond voxel 3d editing: Learning from 3d masks and self-constructed data. *arXiv preprint arXiv:2604.13688*, 2026.

Yunhan Yang, Yukun Huang, Yuan-Chen Guo, Liangjun Lu, Xiaoyang Wu, Edmund Y. Lam, Yan-Pei Cao, and Xihui Liu. SAMPart3D: Segment any part in 3d objects. *arXiv preprint arXiv:2411.07184*, 2024.

Youtan Yin, Yanning Zhou, Jiacheng Wei, Xiaofeng Yang, Jun Zhang, Jiayang Bai, Jingwen Ye, Weidong Zhang, and Guosheng Lin. Editverse3d: High-quality 3d object editing with region-aware learning. In *European Conference on Computer Vision*, 2026.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

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.

Algorithm 1 Route–Diagnose–Refine Multi-view Editing

Require: assets $\mathcal{A}$, instruction y, ordered initial cameras $\mathcal{V}^0$, candidate set Π, and round budget $R_{\max}$

Obtain $\mathcal{G}^0$, Ω^0, and the initial native-3D state (Sec. 3.1).
for $r = 0, \dots, R_{\max} - 1$ **do**
  Render $\{I_i^{s,r}\}$, synthesize $\{I_i^{t,r}\}$, and compute supports $\{\chi_i\}$.
  Route: compute $\{\Delta \boldsymbol{v}_i\}$ and $\{\alpha_i\}$, then edit inside Ω^r (Eqs. 3–9).
  Diagnose: compute D^r, E^r and partition Ω^r into C^r, S^r, U^r (Eqs. 10–15).
  if $C^r = \varnothing$ or $|C^r|/|\Omega^r| \le \delta_{\text{stop}}$ **then**
    return the current state.
  end if
  Refine: construct Ω^{r+1}, select $\pi^\star$, synthesize its target, and append the view.
  Continue the native-3D state using Eq. 18.
end for
return the final state.

A Detailed Algorithm for Route–Diagnose–Refine Multi-view Editing

See detailed algorithm in Alg. 1.

B Experimental Setup

B.1 Benchmark and Baseline Adaptation

Our benchmark contains 194 language-guided composition cases. Human characters are sampled from Adobe Mixamo (Adobe, 2021), while composable objects are drawn from PartObjaverse-Tiny (Yang et al., 2024), including wearables, handheld tools, decorations, and accessories. We manually select compatible asset pairs and use GPT models to generate free-form instructions describing spatial or functional interactions, such as wearing, holding, attaching, supporting, or leaning.

Our method starts directly from the original source assets and instruction. SuperDec (Fedele et al., 2025) and the language-guided planner automatically produce the coarse composition $\mathcal{G}^0$ and initial edit mask Ω^0, while Step1X-Edit (Liu et al., 2025) generates target images from rendered source views. Two edited views are used initially, and at most one view is added per subsequent round. Newly acquired targets additionally condition on the original primary-view source–target pair for consistency, and view-support fields are recomputed from the updated geometry each round.

Step1X-Edit provides the edited RGB target rather than the support mask used in Eq. (4). We obtain this 2D mask separately from the source–target image difference and use it as the soft instruction-induced edit support B_i; thus B_i denotes the edited region, not the full object foreground.

We compare with TRELLIS (Xiang et al., 2025), TRELLIS.2 (Xiang et al., 2026), Easy3E (Hu et al., 2026), VoxHammer (Li et al., 2026), Instant3dit (Barda et al., 2024), and Vox-E (Sella et al., 2023). Inputs are adapted to each method while keeping the underlying composition and guidance fixed. Methods requiring a single source asset receive the same automatically composed $\mathcal{G}^0$; image-conditioned methods receive source renderings and Step1X-Edit targets; and text-conditioned methods receive the original instruction. All assets are normalized to the same canonical coordinate system, with format conversion applied when required.

B.2 Editing Configuration

We implement our method on Easy3E with TRELLIS-image-large. Each source asset is rendered from 150 cameras sampled by a Hammersley sphere sequence using Blender Cycles at 512×512 resolution. DINOv2 ViT-L/14 (Oquab et al., 2023) features are extracted at 518×518 and cached together with sparse-structure and SLAT latents. Assets are normalized to $[-0.5, 0.5]^3$. The editable region is voxelized on a 64^3 grid and downsampled to a 16^3 latent mask with 8 channels.

Editing follows two stages: sparse-structure FlowEdit and SLAT repainting. Both use 25 sampling steps, source and target CFG scales of 5.0, random seed 1, and FlowEdit coefficient $\lambda = 1.0$. At the beginning of each refinement round, we recompute the view-support fields χ_i from the current geometry and source/target renderings as described in Eq. (6). We then convert these support fields into priority-aware ownership weights α_i using Eq. (7): later-acquired views take local precedence only within their supported regions, while the base view receives the residual ownership not claimed by later views. The same ownership weights are used for sparse-structure differential-flow routing and subsequent SLAT repainting. Final results are exported as textured GLBs without mesh simplification and with texture size 2048.

B.3 Evaluation Protocol

For quantitative evaluation, every method is rendered from the same fixed cameras at 512×512 using Blender Cycles with 32 samples. We use azimuths $0°$, $120°$, and $270°$. CLIP-T and the paired-image metrics are computed per view and averaged across views before benchmark-level aggregation. We report CLIP-T, DINO-I, LPIPS, FID, PSNR, and SSIM. CLIP-T (Radford et al., 2021) uses `openai/clip-vit-base-patch32`, DINO-I (Oquab et al., 2023) uses `facebook/dinov2-base`, LPIPS (Zhang et al., 2018) uses AlexNet (Krizhevsky et al., 2012), and FID (Heusel et al., 2017) uses ImageNet-pretrained InceptionV3 (Szegedy et al., 2016).

Reference images and aggregation. For case b and evaluation camera i, let $J_{b,i}^{\mathrm{ref}}$ be the edited RGB target generated by Step1X-Edit, and let $\widehat{J}_{b,i}$ be the rendering of the evaluated 3D output from the corresponding camera. DINO-I, LPIPS, PSNR, and SSIM compare this pair: their reference is neither the unedited source rendering nor a rendering of an independently constructed 3D target asset. For any of these four metrics m, the per-case and benchmark means are

$$\overline{m}_b = \frac{1}{n_b} \sum_{i=1}^{n_b} m\left(\widehat{J}_{b,i}, J_{b,i}^{\mathrm{ref}}\right), \qquad \overline{m} = \frac{1}{B} \sum_{b=1}^{B} \overline{m}_b, \tag{19}$$

where n_b is the number of evaluation views and B the number of cases; Table 1 uses $n_b = 3$ and $B = 194$. Each method is compared with the same target images. Where worst-view fidelity is reported, we take the minimum across views for DINO-I, PSNR, and SSIM and the maximum for LPIPS, rather than selecting the most favorable view. These measurements quantify agreement with the generated edited references. They do not, by themselves, measure source preservation in unchanged regions or correctness against independent 3D ground truth; imperfections in the Step1X-Edit targets can affect the scores.

B.4 Residual-Target Construction and Diagnosis Metrics

Reported experiment. Table 3 evaluates the first completed editing round for all 194 benchmark assets. The reference is constructed with Claude Sonnet 4.5 (`claude-sonnet-4-5`): the VLM scores residual editing errors in multiple views, producing 2D error masks that are projected into 3D. For each asset, the resulting reference is shared by every compared configuration, and the evaluation settings are identical across configurations. No later-round outputs enter this diagnosis table. The following description concerns this benchmark experiment, not alternative residual configurations from preliminary implementation examples.

VLM reference versus native 3D predictions. The VLM assesses differences between asset renderings and their corresponding Step1X-Edit target images. Let $r_{b,i}^{\mathrm{2D}}$ denote its scored error mask for asset b and view i. Projecting these view-specific annotations with the associated cameras and combining them in the asset's 3D coordinate frame produces a voxelwise reference score r_b^{ref} and an error-region reference mask G_b^{err}. The same r_b^{ref} and G_b^{err} are used for all configurations of asset b; the reference is not regenerated separately for each candidate method.

In contrast, let $\widehat{M}_{b,k}^{(1)}$ be the predicted 3D mask from configuration k after its first editing round, with native voxel coordinates $\mathcal{P}_{b,k}^{(1)}$. This prediction is the voxel set produced by the corresponding method itself. It is *not* reconstructed by projecting a 2D mask. Thus the two sides of the evaluation have different origins: projection constructs the VLM-derived reference, whereas the candidate masks

and their associated scores come from native 3D method outputs. The evaluated mask is not used as its own ground truth.

Shared evaluation protocol and reference labels. Each configuration's native output is compared with the same per-asset 3D reference using the same evaluation procedure. The shared reference must be distinguished from the method-specific predicted voxel sets: identical ground truth does not make those predictions identical. Let $\mathcal{E}_{b,k}$ denote the voxel coordinates compared for asset b and configuration k by this evaluator. The binary reference label is membership in the common error mask,

$$y_b(\mathbf{p}) = \mathbf{1}[\mathbf{p} \in G_b^{\mathrm{err}}], \qquad \mathbf{p} \in \mathcal{E}_{b,k}. \tag{20}$$

Positive labels therefore identify reference error regions; they are not created by thresholding the candidate disagreement score or its conflict mask. The reference-mask construction and labeling settings are shared across configurations. The binary prediction $\widehat{M}_{b,k}^{(1)}$, continuous candidate score, reference score r_b^{ref}, and reference label y_b are distinct quantities. A missing reference observation is not evidence that a voxel is error-free.

Candidate scores. The disagreement score is D^r from Eq. (10). The edit-magnitude baseline is *the same* E^r used by the method, computed from the routed *differential* velocity:

$$E^r(\mathbf{p}) = \frac{1}{T} \sum_{m=1}^{T} \|\Delta \mathbf{v}_{\mathrm{route}}(\mathbf{p}, t_m)\|_2. \tag{21}$$

This is Eq. (11), not a target-only velocity norm. For Table 3, r refers to the first completed round. The visibility- and support-based comparisons likewise use the outputs of their respective 3D configurations rather than projections of the reference masks. All candidate outputs are evaluated against the shared VLM-derived reference under the same settings.

Metric definitions. For a candidate score $s_{b,k}$ on $\mathcal{E}_{b,k}$, Spearman correlation measures rank agreement with the reference error scores:

$$\rho_{b,k} = \mathrm{Spearman}\left(\{s_{b,k}(\mathbf{p})\}_{\mathbf{p}\in\mathcal{E}_{b,k}}, \{r_b^{\mathrm{ref}}(\mathbf{p})\}_{\mathbf{p}\in\mathcal{E}_{b,k}}\right). \tag{22}$$

AUROC measures the ranking of reference-positive versus reference-negative voxels using y_b, while AUPRC summarizes precision–recall performance against the same labels. The reference label remains fixed as a candidate score threshold is swept; it is not redefined by the score being evaluated. The evaluator formulation expresses AUPRC as average precision,

$$\mathrm{AP}_{b,k} = \sum_{\ell} (R_\ell - R_{\ell-1}) P_\ell, \tag{23}$$

where P_ℓ and R_ℓ are precision and recall at the successive score thresholds. This is distinct from trapezoidal PR-curve integration.

For top-10% recall, set $q = \lceil 0.1|\mathcal{E}_{b,k}| \rceil$ and select the q highest-scoring voxels. Then

$$\mathrm{Recall@10\%}_{b,k} = \frac{\sum_{\mathbf{p}\in \mathrm{Top}_q(s_{b,k};\mathcal{E}_{b,k})} y_b(\mathbf{p})}{\sum_{\mathbf{p}\in\mathcal{E}_{b,k}} y_b(\mathbf{p})}. \tag{24}$$

The 10% is a *prediction-selection budget*, not the positive-label fraction or a prescribed threshold on the VLM score. Spearman correlation is undefined for constant arrays, AUROC requires both label classes, and top-10% recall has a zero denominator if no reference-positive voxel is present. These exceptional cases and tied scores require consistent handling across configurations.

Missed Edit Rate. We additionally measure how often voxels requiring further editing are assigned to the edited-stable state after the first completed editing round. For asset b and configuration k, let $\widehat{S}_{b,k}^{(1)}$ be the native 3D edited-stable set produced by that configuration's state partition, corresponding to S^r in Eq. (15). Using the same VLM-derived need-further-edit reference G_b^{err} and evaluation coordinates $\mathcal{E}_{b,k}$ defined above, the per-asset Missed Edit Rate (MER), in percent, is

$$\mathrm{MER}_{b,k} = 100\, \frac{\left|\widehat{S}_{b,k}^{(1)} \cap G_b^{\mathrm{err}} \cap \mathcal{E}_{b,k}\right|}{|G_b^{\mathrm{err}} \cap \mathcal{E}_{b,k}|}. \tag{25}$$

Concretely, the computation aligns the predicted state labels and reference labels on the evaluation coordinates, counts all valid reference-positive voxels, counts those also assigned to edited-stable, and divides the latter count by the former. Lower is better. The denominator is the number of voxels labeled as requiring further editing, *not* the size of the predicted edited-stable set; adding reference-negative voxels to that set therefore does not reduce this rate. The metric uses the predicted state partition rather than a stable-region reference constructed from the prediction itself, and requires no manual source-preservation annotation. It is undefined for an asset with no valid reference-positive voxel; such a case must not be interpreted as a measured zero error rate. Missing reference observations are not treated as error-free.

Scope of the reference. The diagnosis ground truth is a VLM-derived 3D annotation obtained from multi-view image evidence, not an independently reconstructed target 3D asset. Its construction is separate from the candidate native 3D masks, although it shares the Step1X-Edit target images with the editing task. It identifies remaining editing errors rather than regions that must retain the source state. Source-preservation evaluation therefore uses a different, manually annotated 3D reference, described next.

The VLM-derived reference can contain annotation errors, and the projection of 2D labels into 3D can introduce localization uncertainty. Missed Edit Rate is therefore interpreted relative to this proxy reference, rather than as a noise-free estimate of true editing failures; a nonzero rate cannot be attributed solely to annotation noise. Under the shared protocol in Table 3, our rate of 15.6% is lower than the visibility-, support-, and edit-magnitude-based settings (98.5%, 89.2%, and 50.2%, respectively). This provides evidence that voxels judged to need further editing are less often prematurely assigned to edited-stable by our method. Together with the diagnosis and source-preservation results, this supports the rationale of our three-state diagnosis. It does not certify that every edited-stable voxel is correct or that all remaining reference-positive voxels are assigned to the conflict state rather than the source-stable state.

B.5 Source-Preservation Mask Evaluation

Annotated subset and reference. The precision, recall, and F1 columns of Table 5 are evaluated on a subset of 50 assets from the 194-asset benchmark. One author manually annotates a 3D preservation mask for each asset in this subset. Denote this annotation by $M^{\mathrm{ref}}_{\mathrm{preserve},b}$: it marks the 3D regions to be source-preserved. These are direct manual 3D annotations, not back-projected VLM error masks and not complements of residual-positive labels. The 50-asset subset applies to the preservation-mask statistics; the first-round diagnosis in Table 3 uses all 194 assets.

Predicted source-preservation mask. Let $\widehat{M}^{r}_{\mathrm{preserve},b}$ be the voxel set marked for source preservation in the evaluated output of asset b after r rounds. At the final output this is the source-preserved set recorded when editing finishes; the same semantic definition applies to each reported round-wise output. It is the algorithm's source-preservation collection, corresponding to M_{preserve} in Sec. 3.1, rather than all voxels excluded from further editing. In particular, edited-stable voxels retained in their edited state are not counted as source-preserved merely because they are frozen. This distinction separates preserving source content from retaining an already completed edit.

Set-based mask metrics. For notation, let $\mathcal{H}^{r}_{b}$ denote the evaluation coordinates at which the prediction and manual reference are compared. Write

$$A^{r}_{b} = \widehat{M}^{r}_{\mathrm{preserve},b} \cap \mathcal{H}^{r}_{b}, \qquad B^{r}_{b} = M^{\mathrm{ref}}_{\mathrm{preserve},b} \cap \mathcal{H}^{r}_{b}. \tag{26}$$

Treat source-preservation membership as the positive class. The counts are

$$\mathrm{TP}^{r}_{b} = |A^{r}_{b} \cap B^{r}_{b}|, \qquad \mathrm{FP}^{r}_{b} = |A^{r}_{b} \setminus B^{r}_{b}|, \qquad \mathrm{FN}^{r}_{b} = |B^{r}_{b} \setminus A^{r}_{b}|. \tag{27}$$

Precision, recall, and F1 are consequently

$$\begin{aligned} \mathrm{Prec}^{r}_{b} &= \frac{\mathrm{TP}^{r}_{b}}{\mathrm{TP}^{r}_{b} + \mathrm{FP}^{r}_{b}}, & \mathrm{Rec}^{r}_{b} &= \frac{\mathrm{TP}^{r}_{b}}{\mathrm{TP}^{r}_{b} + \mathrm{FN}^{r}_{b}}, \\ \mathrm{F1}^{r}_{b} &= \frac{2\mathrm{TP}^{r}_{b}}{2\mathrm{TP}^{r}_{b} + \mathrm{FP}^{r}_{b} + \mathrm{FN}^{r}_{b}}. \end{aligned} \tag{28}$$

Thus precision measures how much of the predicted source-preserved set agrees with the manual annotation, while recall measures how much of the annotated preservation region is covered. These mask metrics are distinct from the rendered-image agreement metrics in Appendix B.3.

B.6 Inference Time

Our method uses 3.4 editing rounds on average. Each refinement round takes approximately 79 s: 46 s for native-3D editing and 33 s for Step1X-Edit target generation.

Table 6: Average inference-time breakdown per refinement round.

Component	Time
3D latent editing	46 s
Step1X-Edit target generation	33 s
Total per round	79 s
Average number of rounds	3.4

C Language-guided Coarse Composition Prompt

As described in Sec. 3, we use a language model to reason over the original asset renderings and their SuperDec abstractions, and to predict a coarse composition by modifying the corresponding superquadric parameters. The prompt used for this stage is shown below.

```
You are a 3D coarse composition assistant. You will be shown:
1. Renderings of multiple original 3D assets
2. Multi-view renderings of their abstracted versions made of
    superquadrics
3. JSON files describing each superquadric's parameters
   (asset ID, index, translation, rotation, scale, ...)
4. A user instruction describing the desired composition

Your task is to coarsely compose these assets according to the user's
instruction while preserving the original shapes as much as possible.

Please respond in the following format:
## Analysis
Briefly describe the assets and identify the parts relevant to the
requested interaction.

## Composition Plan
Explain which asset(s) or superquadrics should be transformed and how
(e.g., translation, rotation, scale). Prefer coherent asset-level
transformations and only modify individual parts when necessary.

## Updated JSON
Provide the complete updated JSON containing all assets and their
superquadrics:
[
 {
   "asset_id": 0,
   "superquadrics": [
     {
       "index": 0,
       "scale": [...],
       "translation": [...],
       "rotation": [...],
       ...
     },
     ...
   ]
 },
 ...
]
```

```
Important:
- Keep ALL assets and superquadrics in the output unless removal is
    necessary.
- Preserve the identity and geometry of each source asset as much as
    possible.
- Prefer translation and rotation over shape deformation.
- Modify only the assets or parts necessary for establishing the
    requested
  spatial or functional relation.
- Avoid severe penetration, floating, or disconnected interacting
    parts.
- Make the composition as MINIMAL as possible; fine geometric details
    will
  be refined in later stages.
```

D Supplementary Priority-Aware Routing Ablation

We provide an additional quantitative comparison of three representative routing strategies under the same editing setup. DINO-I and LPIPS are reported for both the average view and the worst view to expose failures that may be hidden by view-averaged scores.

Table 7: Supplementary quantitative comparison of routing strategies. DINO-I and LPIPS are reported as average and worst-view performance.

Method	CLIP-T ↑	DINO-I ↑		LPIPS ↓	
		Avg.	Worst	Avg.	Worst
Uniform	0.312	0.813	0.762	0.172	0.234
Average	0.308	0.824	0.798	0.164	0.218
Priority-aware (Ours)	**0.317**	**0.857**	**0.826**	**0.159**	**0.172**

E More Results

Figure 6 presents additional language-guided multi-asset composition results produced by our full method.

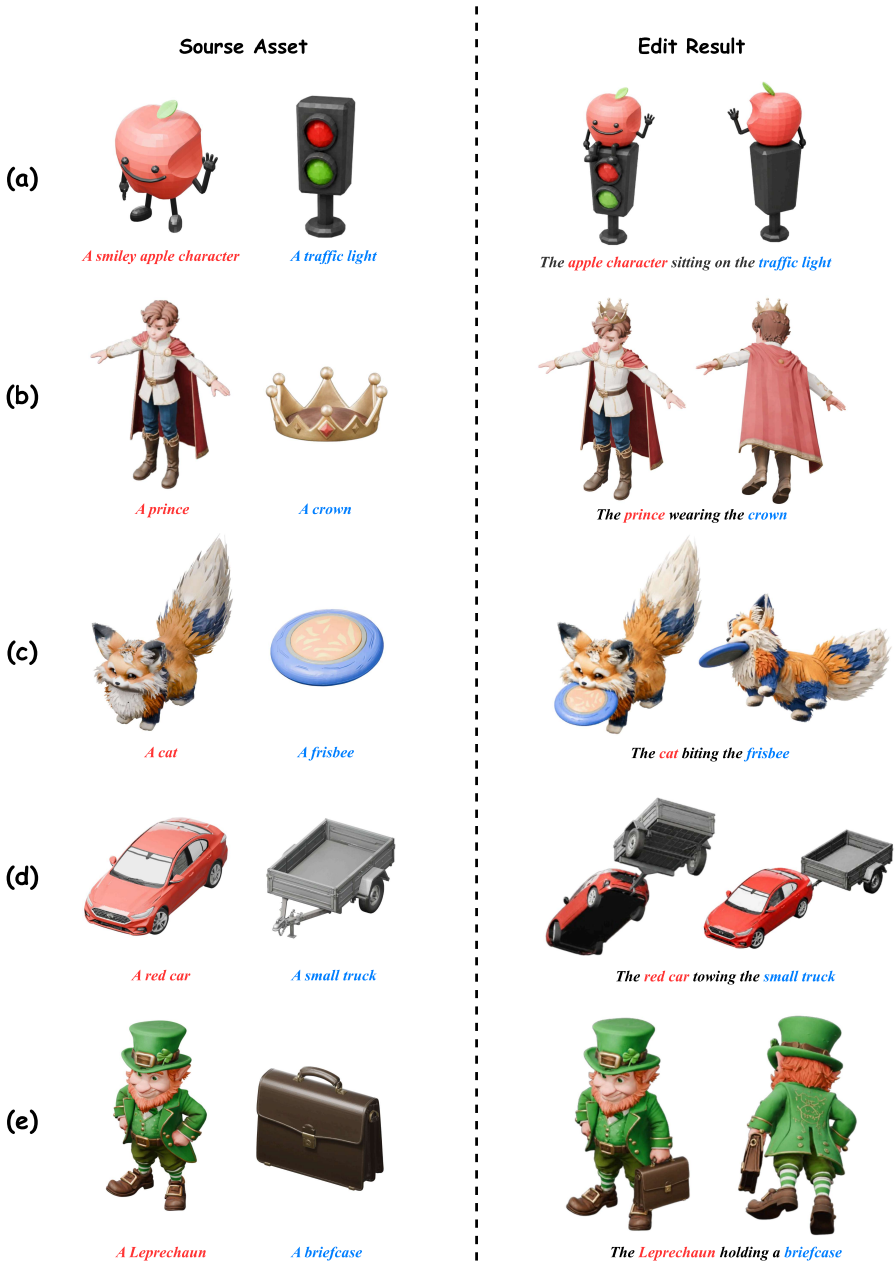

Figure 6: Additional results of our language-guided multi-asset 3D composition method.