SAP3D: SPATIALLY ADAPTIVE PERSONALIZATION FOR CONSISTENT 3D GENERATIVE EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Image-to-3D foundation models produce clean, detailed 3D assets, but offer no direct way to edit an existing one, be it generated or manually crafted. A natural route is to edit a 2D render of the asset and regenerate from it. Existing editors either hold part of the source's latent representation fixed, as frozen latent tokens or cached voxel features, which produces geometric artifacts and preserves nothing under a global edit, or couple the generation to the source's trajectory, which holds the edit back. We instead give the generator a prior for the specific asset: a low-rank adapter is fitted to the source object and applied at a variable strength per latent token, so that tokens in the edit region run without the adapter and follow the edited image, while every other token runs with it and reproduces the source. To decide which tokens belong to the edit, we use two probes: a 2D probe on the transformer's image cross-attention, which selects tokens by image content, and a 3D probe on the shape decoder's cross-attention, which grounds the region in 3D and reaches geometry the edited view does not show. Since VecSet tokens are not bound to fixed locations and drift during denoising, we re-estimate the assignment at every step. Because the adapter carries the asset's identity rather than any one region of it, a uniform strength also turns it into a control for global edits. On Edit3D-Bench, our method achieves the highest edit fidelity of all compared methods at numerically competitive preservation, and participants in a user study preferred its results in 88% of the comparisons that were not ties.

1 INTRODUCTION

Recent 3D generative models produce detailed shapes from a single image (Tencent Hunyuan3D Team, 2025b;c; Xiang et al., 2025; Li et al., 2025b). Artistic workflows, however, consistently require further refinement, through local edits such as replacing a wheel or adding a strap, and global ones, changing, for example, overall material, style, or pose. To tackle this challenge, a recent line of generative shape editing works has been proposed, where a given 3D shape, possibly one produced by an image-to-3D model, is edited using such a backbone, guided by a target image that depicts an edited shape from a single view. While different approaches have been proposed, all of them seek to retain identity by reusing latents of the source shape, obtained by encoding or inverting it. Some approaches (Hsiao et al., 2026; Li et al., 2026a; Hu et al., 2026b) explicitly retain latent values that correspond to regions that should not change, which occasionally results in visual artifacts, and does not induce any preservation for global edits. Other approaches (Zhou et al., 2026; Lim et al., 2026) couple the source and target generation trajectories over the whole shape, without a mask. This encourages some identity preservation but, as we demonstrate, it also dampens editing extent.

In this paper, we present **SAP3D**, a spatially adaptive personalization framework for consistent 3D generative editing. Rather than relying on source latents, which entangle appearance and pose with identity, we encode the source asset in the prior of the generator. Inspired by personalization in the image domain, SAP3D uses a simple-to-train Low-Rank Adaptation (LoRA (Hu et al., 2022)) to personalize a VecSet backbone to a single input asset. The adapter is simply trained to reconstruct the source shape from renderings of it with varying camera and lighting conditions. As we demonstrate, a personalized model robustly preserves identity under global editing, accurately reconstructs unedited regions, and naturally allows preservation strength control through the LoRA strength parameter. For strong editing, the personalized and base generators are combined with spa-

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.

tially varying strength, allowing greater source preservation for unedited regions and more freedom for edited ones.

Achieving this behavior requires *localization* during generation, indicating which latent values correspond to which region in the resulting shape. For voxel-based approaches (Li et al., 2026a; Hu et al., 2026b), localization is natively encoded in the data structure. Instead, we focus on vector-set (VecSet) based generators, which include leading open models (Tencent Hunyuan3D Team, 2025a; Li et al., 2025b), but represent shapes as an unordered collection of latent vectors with no explicit spatial address, requiring more careful consideration for localization. Among prior VecSet-based editors, only VecSet-Edit (Hsiao et al., 2026) localizes explicitly, selecting edit-relevant tokens according to their attention to the input image. VecSet-Edit prunes this selection once, early in denoising, since tokens drift, meaning their role and localization change during the diffusion process; after that the selection is fixed and can point to incorrect final regions. Moreover, the 2D image-based signal is partial, leaving the identity of unseen regions under-specified. In contrast, **SAP3D** localizes with complementary 2D and 3D cues and tracks them throughout denoising, providing a sharp and reliable estimate of where personalization should be retained and where the model should be given more freedom.

Our localization mechanism pairs a 2D probe of the image cross-attention, read from attention heads selected for their spatial fidelity, with a 3D probe that leverages the cross-attention values in the backbone's decoder between 3D query points and the generated set. We repeatedly re-evaluate these correspondences during denoising to account for their evolution, and take the union of the two selections, each thresholded strictly, to decide where personalization is applied.

We implement SAP3D on Hunyuan3D 2.1 (Tencent Hunyuan3D Team, 2025a) and evaluate it on Edit3D-Bench (Li et al., 2026a), considering both local and global editing scenarios. Our experiments show that spatially adaptive personalization delivers high-quality edits at a preservation that is numerically competitive and visually accurate. We further demonstrate that the continuous adaptation strength provides a unified mechanism spanning conservative preservation to substantial global and local transformations. These results suggest a step forward in bringing 3D generation closer to artist-friendly workflows.

2 Related Work

3D foundation models and the locality of their latents Image-to-3D generation is led by two families. VecSet models (Zhang et al., 2023a; Zhao et al., 2023; Zhang et al., 2024; Chen et al., 2025a; Li et al., 2025b; Tencent Hunyuan3D Team, 2025b; Li et al., 2025a) encode a shape as a set of hundreds to thousands of latent vectors, denoise the set with a diffusion or flow-matching transformer under image cross-attention, and decode it with a network that cross-attends from query points to the set to produce an occupancy or signed distance field. Structured-latent models (Xiang et al., 2025; 2026; Wu et al., 2026; Li et al., 2026b; Chen et al., 2025b) generate a sparse voxel structure first and a latent per occupied voxel second. The grid gives every latent an address at the cost of its resolution; the set gives no address but underlies leading open generators (Tencent Hunyuan3D Team, 2025a; Li et al., 2025b;a), which is the setting we work in.

Structured-latent papers describe the set as global (Li et al., 2026b; Ye et al., 2026b), and Hunyuan3D states that a token "does not correspond to a fixed location" (Tencent Hunyuan3D Team, 2025b). The decoder's attention says otherwise. FlashVDM found that each query point attends to a handful of tokens and used this to speed up decoding (Lai et al., 2025); TanGO (Lim et al., 2026) measured the same locality on Hunyuan3D 2.1 (Tencent Hunyuan3D Team, 2025a); and LATTICE notes that position information is "secretly encoded" in the set but that no model uses it, since query positions are unknown for a generated latent (Lai et al., 2026). Reading location from attention is established in 2D diffusion (Hertz et al., 2023; Tang et al., 2023; Cao et al., 2023; Tian et al., 2024), where recent work shows the signal concentrates in specific layers and is most reliable at intermediate noise (Helbling et al., 2025; Kim et al., 2026), and the layers that matter most for editing are spread across depth rather than grouped by position (Avrahami et al., 2025). In 3D, the decoder's attention has been read only offline, on clean latents (Lai et al., 2025; Lim et al., 2026); no method selects tokens from a user's 3D region while sampling, or tracks how the assignment moves along the trajectory.

Editing 3D assets with 2D priors Early methods refit the asset into a neural field and optimize it under a 2D diffusion prior (Haque et al., 2023; Sella et al., 2023; Zhuang et al., 2023; Chen et al., 2024b; Zhuang et al., 2024; Miao et al., 2025), losing topology and UVs. Other methods edit rendered views and lift them back to 3D, by optimization or with a feed-forward reconstructor (Chen et al., 2024a; Qi et al., 2024; Huang et al., 2025; Erkoç et al., 2025; Barda et al., 2025). EditP23 (Bar-On et al., 2025) takes the same signal we do, a render and its edited version, without a mask, but the whole object passes through the reconstructor. Unless the original is merged back afterwards (Erkoç et al., 2025; Barda et al., 2025), the unedited region is regenerated rather than kept, and what the views do not show is guessed.

Editing inside 3D foundation models Trained editors learn from synthetic before-and-after pairs (Xia et al., 2026; Ma et al., 2025; Yin et al., 2026; Weng et al., 2026; Gat et al., 2026; Ye et al., 2026a), from about 10^5 to 10^7 of them. Most give the user no way to say where to edit; EditVerse3D (Yin et al., 2026) takes a coarse 3D box, but makes only local edits, adding or replacing a part as its pairs, made by deleting parts, teach it. Where pairs are made by regenerating a box around each edit, unedited content inside the box can change, and such inconsistencies "propagate into the end-to-end model during training" (Ye et al., 2026a).

Training-free methods on grid latents invert or re-noise the asset and write its latents back into the preserved voxels, during sampling (Li et al., 2026a; Chi et al., 2026; Sella et al., 2026; Hu et al., 2026b) or once per stage (Ye et al., 2026b). The recipe is already softening. EditFlow3D (Nie et al., 2026) finds hard replacement "incompatible with the evolving edited region" and turns it into a guidance loss, and VS3D (Liu et al., 2026a) blends source and edited latents with a per-token weight set by how much they agree. All of it presumes a fixed address per latent, and Nano3D reports that its editing collapses or under-edits when moved to Hunyuan3D 2.1 (Ye et al., 2026b).

On set latents, editing has been mostly mask-free. AnchorFlow (Zhou et al., 2026) shares a latent anchor between source and target flows and applies it to all tokens alike, while TanGO (Lim et al., 2026) scales each token's update by how much the velocities predicted under the source and edited images disagree. VecSet-Edit (Hsiao et al., 2026) is the one method that also takes a 2D mask. It selects the tokens to edit from image cross-attention before sampling, expands the selection through self-attention, re-noises the source values of the rest, and prunes the selection once, early, since tokens are "spatially mobile rather than fixed on a grid" and early drift causes "irreversible interference" between the two groups. In all of these set-latent editors, identity is carried by the source latents or the conditioning image; on a set latent, the first moves and the second is partial. None carries a prior for the specific asset, and none lets the user reach what the image does not show. For a whole-object edit, a mask leaves nothing to preserve, and coupling to the source trajectory holds the edit back.

Personalization and localized adapters Fitting a generator to a subject is standard in 2D (Gal et al., 2023; Ruiz et al., 2023; Hu et al., 2022), and single-image editors fit the model to the image before editing it (Kawar et al., 2023; Zhang et al., 2023b); SINE interpolates fine-tuned and pretrained scores with a global weight, an early form of the knob we obtain per token. In 3D, personalization has mostly gone through 2D or multi-view priors (Raj et al., 2023; Liu et al., 2024; Zhuang et al., 2024), most recently DreamEdit3D (Ai et al., 2026), which fits a multi-view model to the asset's renders, edits, and lifts with a reconstructor, so identity passes through a low-resolution 2D model. PEGAsus (Hu et al., 2026a) fine-tunes TRELLIS itself on a single shape, but to learn its geometry or appearance as a concept for generating new shapes, not to edit it. LoRA on native 3D generators has served only as a dataset-level task adapter (Gat et al., 2026), and per-subject adapters exist only on a 3D-aware GAN (Qi et al., 2025). No prior work fits a native 3D generator to an asset in order to edit it.

Routing an adapter to some tokens and not others has been used in 2D to compose several subjects, each LoRA gated to its region by a binary mask (Liu et al., 2026b; Dalva et al., 2026; Yang et al., 2025), after earlier methods that regionalized prompts or blended latents rather than weights (Gu et al., 2023; Kong et al., 2024). InstructMoLE (Xiao et al., 2025) reports that learned per-token routing fragments the image; our routing follows the geometry of the edit and applies one adapter against the base model. Adapter strength has served as a semantic slider (Gandikota et al., 2024), and T-LoRA schedules the adapter's rank over timesteps (Soboleva et al., 2026), but no method varies the strength continuously over space. To our knowledge, no prior work applies an adapter

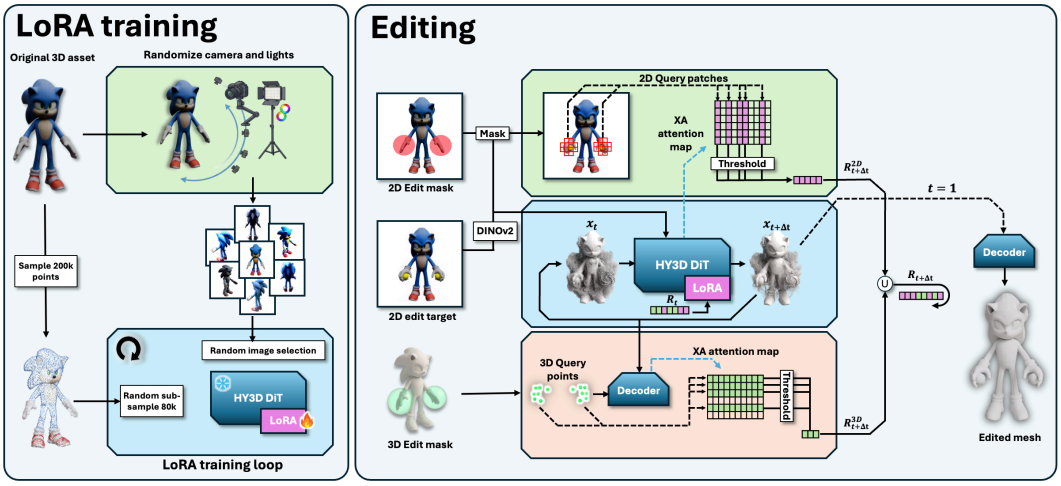

Figure 1: **Overview of SAP3D.** *Left:* we fit a LoRA to the source asset, pairing renders under random cameras and lighting with points sampled from its surface. *Right:* at each denoising step, the 2D probe reads the DiT's cross-attention to the image patches under the 2D edit mask, and the 3D probe reads the VAE decoder's cross-attention at query points inside the 3D edit region. Each probe thresholds its own scores, and their union $R_{t+\Delta t}$ determines which tokens are generated without the adapter at the next step. At $t = 1$ the decoder extracts the edited mesh.

per token in a 3D generator, or uses a continuous per-token scale to decide, step by step, where an asset's identity should hold.

3 METHOD

Given a source asset, an edited view of it and a region delimiting the edit, SAP3D synthesizes the edited asset in a single generation from noise (Figure 1). We first personalize the generator to the source asset with a low-rank adapter, so that the asset is encoded as a prior of the model (Section 3.2). During editing, the adapter's strength is modulated per latent token and per denoising step: tokens that realize the edited part are generated by the base model under the guidance of the edited view, while the remaining tokens are generated by the personalized model and reproduce the asset (Section 3.3). The assignment of tokens to the edit is re-estimated at every step from the model itself, through a 2D probe of the denoiser's image cross-attention and a 3D probe of the shape decoder (Section 3.4).

3.1 PRELIMINARIES

VecSet generation. We build on Hunyuan3D 2.1 (Tencent Hunyuan3D Team, 2025a), which represents a shape as a set of $N = 4096$ latent tokens $x = \{x^i\}_{i=1}^N$, $x^i \in \mathbb{R}^{64}$ (Zhang et al., 2023a). The VAE encoder samples points on the surface, selects N of them by farthest-point sampling, and lets each selected point cross-attend to the full point cloud, yielding one token per selected point. The decoder $\mathcal{D}$ mixes the set through self-attention and predicts the occupancy $o(p) = \mathcal{D}(p, x)$ of an arbitrary query point $p \in \mathbb{R}^3$ by cross-attending from p to the mixed set; the mesh is extracted from o on a grid. This cross-attention is local, each query point drawing on a small number of tokens (Lai et al., 2025; Lim et al., 2026).

The generator is a diffusion transformer v_θ trained with flow matching. Given noise $x_0 \sim \mathcal{N}(0, I)$ and a clean latent x_1, the path is $x_t = t\,x_1 + (1-t)\,x_0$, and the model predicts the velocity $v_\theta(x_t, t, c) \approx x_1 - x_0$, where c denotes the DINOv2 tokens of the conditioning image, attended to by cross-attention in every block. Sampling integrates from $t = 0$ to $t = 1$ with Euler steps, and at each step the current prediction of the clean latent is

$$\hat{x}_1 = x_t + (1-t)\,v_\theta(x_t, t, c). \tag{1}$$

Problem setting. The input comprises a source asset S, a view I' of S edited in 2D, and a region $\Omega \subset \mathbb{R}^3$ enclosing the part to be edited. A 2D mask M of the edit in I' is obtained by projecting Ω into the camera of I'. The goal is a mesh that follows I' within Ω and reproduces S elsewhere, including the parts of S that I' does not depict, with the edited and the preserved geometry integrating naturally where they meet.

3.2 Personalizing the Generator to the Asset

We personalize the generator to S with a low-rank adapter (Hu et al., 2022), analogously to subject-driven personalization of image models (Ruiz et al., 2023). We render S from randomly sampled cameras under randomized lighting, and sample a dense point cloud from its surface. Each training step pairs a randomly chosen render, used as the condition, with a latent representation of S encoded from a random subset of the surface points (Figure 1, left). With the base weights θ frozen, the adapter weights ϕ are optimized with the flow-matching objective

$$\mathcal{L}(\phi) = \mathbb{E}_{t,\,x_0,\,j} \left\| v_{\theta,\phi}(x_t, t, c_j) - (x_1 - x_0) \right\|^2, \tag{2}$$

where c_j denotes the DINOv2 tokens of render j and x_1 the VAE encoding of S. The adapter is restricted to the feed-forward networks of the denoiser, including those within its mixture-of-experts layers. Adapters are not applied to any of the attention projections, as we observed that adapting them destabilized training and degraded the reconstructions (Appendix C).

Having been fit to renders from a variety of views, the personalized model reproduces S from noise under any of its views. It therefore encodes the complete asset, including every part that a single edited view leaves unobserved, as well as properties that rendered images may leave ambiguous, such as orientation or scale. It is this prior, rather than a stored copy of the source latents, that preserves the asset during editing.

3.3 Spatially Adaptive Personalization

Per-token strength. Consider an adapted linear layer with base weight W and low-rank update ΔW, and let h^i denote the input of token i to that layer. Since the adapted layers act on each token independently, the update can be scaled separately for every token,

$$y^i = W h^i + s^i \, \Delta W h^i, \tag{3}$$

where s^i is the strength assigned to token i and is shared by all adapted layers. The attention layers carry no adapter and couple all tokens in every block. A token at strength 0 is thus updated by the pretrained weights alone, yet it attends to, and is attended by, the entire set, including the tokens that continue to reproduce the asset. The edited part is consequently synthesized jointly with the preserved part, in a shared frame and under mutual attention, rather than generated without the context of the asset and then merged into it, which can leave the two unnaturally combined or the edit forced into place. How the two parts meet is decided by the generator itself.

Editing is a single generation from noise conditioned on I', in which the strength of token i at time t is

$$s_t^i = \begin{cases} \alpha_{\text{pre}} & t < \tau_m, \\ 0 & \tau_m \le t < \tau_d \text{ and } i \in R_t, \\ \alpha_{\text{out}} & \tau_m \le t < \tau_d \text{ and } i \notin R_t, \\ 0 & t \ge \tau_d, \end{cases} \tag{4}$$

where R_t is the set of tokens assigned to the edit (Section 3.4). In the earliest steps the predicted shape is still coarse, and we found the token selections made there to be less reliable; all tokens therefore share a single strength α_{pre} until τ_m. From τ_m onward, tokens in R_t are generated by the base model and follow I', while the remaining tokens retain the adapter at strength α_{out} and reproduce S; we refer to $\tau_m \le t < \tau_d$, in which the adapter is selectively disabled, as the selective phase. The probes first run during the step at τ_m, so R_{τ_m} is empty and the first region applies from the following step. From τ_d onward, all tokens are generated by the base model. This final phase is optional: omitting it ($\tau_d = 1$) leaves both of our metrics unchanged within their noise, but we found it to improve the result visibly on several cases (Appendix C).

Global edits. Applying a single strength α to all tokens throughout sampling turns the same prior into a control for global edits, which require no region. The strength interpolates between generations that remain close to S ($\alpha = 1$) and generations that follow I' under the base model alone ($\alpha = 0$); intermediate values retain the identity of the asset while allowing the edited view to alter its style or pose throughout (Section 4.3).

3.4 Locating the Edit During Sampling

Unlike a voxel latent, a VecSet latent assigns no fixed location to a token: the part of the shape a token represents emerges during sampling and may shift from one step to the next (Hsiao et al., 2026); Appendix D measures this on 10 assets. An assignment of tokens to the edit made once therefore grows stale as sampling proceeds. VecSet-Edit (Hsiao et al., 2026) determines the editable set once and thereafter only removes tokens from it. We instead re-estimate the assignment at every step of the selective phase: two probes read the model's current state, each returns a set of tokens, and their union defines the region applied at the next step,

$$R_{t+\Delta t} = R^{\mathrm{2D}}_{t+\Delta t} \cup R^{\mathrm{3D}}_{t+\Delta t}. \tag{5}$$

2D probe: image patches under the mask. Let P denote the DINOv2 patches of I' covered by M over at least half their area. For token i and patch p, the cross-attention weight A^{2D}_{ip}, averaged over a subset of heads selected for their spatial fidelity (Appendix C.3), quantifies how strongly token i draws on patch p. Each patch in P nominates the tokens that attend to it most strongly, and R^{2D} is the set of nominated tokens. Because the model is conditioned on the edited view, the masked patches depict the target content itself, and the probe identifies the tokens that realize it.

3D probe: query points inside the region. We sample Q points uniformly within Ω, including empty space into which the edit may add geometry, and evaluate the decoder $\mathcal{D}$ at these points on the current prediction $\hat{x}_1$ (Equation (1)). Owing to the locality of the decoder's cross-attention, the attention of a query point concentrates on the tokens that determine the shape around it. Each query point therefore nominates the tokens it attends to most strongly, and R^{3D} is the set of nominated tokens. We also considered a gradient-based signal, the sensitivity of the decoded occupancy to each token, which performs similarly but requires a backward pass through the decoder at every probed step (Appendix C.3). As it is anchored in Ω rather than in the image, the 3D probe also reaches the parts of the edit that I' leaves unobserved.

Selective probes, combined by union. Each probe applies its own threshold to its own scores. For each nominating patch or query point, we sort its scores over the tokens, smooth them and scale them to $[0, 1]$, and locate the knee of the resulting curve, the point at which it departs furthest from the straight line joining its ends (Satopää et al., 2011). The nominator admits every token whose scaled score exceeds the value at the knee multiplied by a factor κ, set separately for each probe. Neither probe is confined by construction: through semantic and contextual association, the 2D probe can reach geometry hidden from view, and either probe can nominate tokens outside its mask. What restricts each to the most relevant tokens is its mask together with a strict threshold. The two probes rest on complementary evidence, the 2D probe on the appearance of the edit in I' and the 3D probe on the extent of Ω, including its occluded parts and empty space. Their union allows each to remain strict, since what one fails to select with confidence the other can supply.

Implementation details and all settings are given in Appendix A.

4 Experiments

4.1 Setup

Benchmark. We evaluate on Edit3D-Bench (Li et al., 2026a), which comprises 100 textured assets drawn from Google Scanned Objects (Downs et al., 2022) and PartObjaverse-Tiny (Yang et al., 2024), each with three edits, for 300 cases in total. Each case provides the source asset, a rendered view of it, an edited version of that view produced by text-guided 2D inpainting, a 3D region enclosing the part to be edited, given as one or more closed primitives (boxes, cylinders or spheres), and a 2D mask of the edit in the rendered view. The benchmark provides no ground-truth edited asset.

Algorithm 1 One edit with SAP3D

Require: adapted generator $(\theta, \Delta W)$, edited view I', mask M, region Ω, time grid $0 = t_0 < \cdots < t_K = 1$, schedule constants $\tau_m, \tau_d, \alpha_{\text{pre}}, \alpha_{\text{out}}$

$c \leftarrow \text{DINOv2}(I')$; $P \leftarrow$ patches covered by M; $\{p_q\}_{q=1}^{Q} \leftarrow$ points sampled in Ω
$x \leftarrow x_0 \sim \mathcal{N}(0, I)$; $R_{t_0} \leftarrow \emptyset$
for $k = 0$ **to** $K - 1$ **do**
  $t \leftarrow t_k$; set s_t^i for every token by Equation (4)
  $v \leftarrow v_{\theta,\, s_t \Delta W}(x, t, c)$ ▷ guided; records $A^{2\text{D}}$
  $\hat{x}_1 \leftarrow x + (1 - t)\, v$
  if $\tau_m \le t < \tau_d$ **then**
    $R_{t_{k+1}} \leftarrow \text{Probe2D}(A^{2\text{D}}, P) \cup \text{Probe3D}(\mathcal{D}, \hat{x}_1, \{p_q\})$
  end if
  $x \leftarrow x + (t_{k+1} - t_k)\, v$
end for
return mesh extracted from $\mathcal{D}(\cdot, x)$

Baselines. We compare against five methods: VoxHammer (Li et al., 2026a) and Easy3E (Hu et al., 2026b), which operate on the sparse voxel latents of TRELLIS (Xiang et al., 2025), and VecSet-Edit (Hsiao et al., 2026), TanGO (Lim et al., 2026) and AnchorFlow (Zhou et al., 2026), which operate on VecSet latents. Every baseline is run from its released code at its default settings, on our hardware and through the same evaluation as our method; the only changes are those needed to supply the benchmark's inputs in each method's expected format. VoxHammer's released code did not produce an output on 40 of the 300 cases, and its metrics are computed over the remaining 260. TanGO and AnchorFlow do not use a mask, and their released code produces untextured meshes; since nearly every metric requires a texture, we texture their geometry with the same pipeline as ours.

Implementation. SAP3D uses Hunyuan3D 2.1 with one adapter per source asset; all settings are given in Appendix A. Textures are produced by a pipeline we built on the Hunyuan3D-Paint model (Tencent Hunyuan3D Team, 2025a) (Appendix A.1). All methods run on a single NVIDIA A100 80 GB GPU.

Benchmark metrics. We report the benchmark's own metrics, computed with its released evaluation code without modification for every method: Chamfer distance (CD) to the source mesh outside the 3D region; PSNR, SSIM and LPIPS between renders of the prediction and of the source outside the 2D projection of the region, over 16 views; DINO-I, the similarity of DINOv2 features between whole renders of the prediction and of the source; and CLIP-T, the similarity between renders of the prediction and a caption of the edited object. The benchmark's FID, which its 100 source assets are too few to estimate reliably, is reported in Appendix B; we do not report its FVD.

Preservation. CD is computed from nearest-neighbor distances between points sampled at random on the two meshes, so it carries a noise floor set by the sampling density: at the benchmark's own sampling density, two independent samplings of the same mesh are at a CD of about 0.005 on average over the benchmark, and errors below that level cannot be resolved (Appendix B). We therefore also report an exact point-to-surface distance outside the 3D region: we sample $200{,}000$ points on each surface, discard those inside the region, measure the exact distance from each remaining point to the other surface, and average over both directions. Its floor, measured the same way, is below 10^{-8}.

Edit fidelity. None of the benchmark's metrics compares the prediction with the edited view that specifies the edit, so none can establish whether the requested edit was produced. We measure it directly: we render the prediction from the camera of the edited view, crop both the render and the edited view to the square bounding the 2D mask, set every pixel of the crop outside the mask to a uniform gray, and report one minus the cosine similarity of their DINOv2 class-token embeddings, and the same quantity computed with CLIP. The DINOv2 distance follows the measure of subject fidelity standard in personalization (Ruiz et al., 2023), here restricted to the edit (Appendix B).

Table 1: **Comparison on Edit3D-Bench** over all 300 cases. The first six columns are the benchmark's metrics, computed with its released code; the last three are our edit-fidelity and preservation (Pres.) metrics (Section 4.1). Best in bold, second underlined. †Over the 260 cases on which it produced an output; over the same cases, SAP3D reaches a CD of 0.0159 and a preservation of 0.0102.

	Benchmark metrics						Edit fidelity		
Method	CD ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DINO-I ↑	CLIP-T ↑	DINOv2 ↓	CLIP ↓	Pres. ↓
VoxHammer†	0.0182	27.13	0.949	0.050	0.905	0.281	<u>0.282</u>	<u>0.113</u>	0.0127
Easy3E	**0.0060**	**33.54**	**0.981**	**0.017**	**0.927**	0.281	0.351	0.120	**0.0012**
VecSet-Edit	<u>0.0175</u>	26.82	0.951	0.049	0.886	0.280	0.388	0.128	<u>0.0101</u>
TanGO	0.0519	22.60	0.911	0.118	0.836	<u>0.283</u>	0.328	0.120	0.0429
AnchorFlow	0.0378	25.01	0.931	0.085	0.841	0.280	0.347	0.125	0.0295
SAP3D (ours)	0.0183	<u>30.94</u>	<u>0.967</u>	<u>0.042</u>	<u>0.910</u>	**0.287**	**0.251**	**0.093**	0.0119

4.2 Main Results

Table 1 compares SAP3D with the baselines on all 300 cases; Figure 2 shows local and global edits, with more in Appendix F.

User study. We further conducted a blind pairwise study on 28 cases, 16 local and 12 global edits, each comparing our result with that of one baseline (Appendix E). For each case, participants were shown the source asset, its rendered view and the edit request, together with the two results on randomly assigned sides, and asked which result better applies the requested edit while otherwise staying faithful to the source, with the option of a tie. Eighteen participants cast 398 votes. Our result was preferred in 88.0% of the votes that were not ties (77.6% to 95.2%, a 95% bootstrap interval over participants and cases), and in 71.9% of those on the global edits; Figure 5 breaks the votes down by baseline.

4.3 Ablations

We ablate each component of SAP3D by changing it alone, holding every other setting at its value in the full method, with one seed on a random half of the benchmark (150 cases); Table 3 in Appendix C lists every variant, with differences paired over these cases.

Personalization. Without the adapter, the base model regenerates the entire asset from the edited view: preservation degrades from 0.012 to 0.042, and edit fidelity from 0.250 to 0.290. A uniform strength exposes the trade-off that spatial adaptivity resolves: at $s = 0.5$ both metrics are worse than those of SAP3D, while at $s = 0.75$ and $s = 1$ the asset is preserved more closely but the edit is largely suppressed (0.327 and 0.421). Every uniform strength edits worse than SAP3D by at least 2.5 paired standard errors.

Strength as a control. Over 12 global edits, where the region spans the entire asset, the distance to the source falls steadily as the strength rises, from 0.051 at $s = 0$ to 0.004 at $s = 1$, while edit fidelity is best at $s = 0.5$ (0.169, against 0.224 for the base model and 0.229 at $s = 1$).

Locating the edit and schedule. Using either probe alone, or fixing the region at τ_m, changes the averaged metrics less than personalization does, since the effect is concentrated on a minority of cases; the case-by-case comparison against a second run of the full method, and the sweep that sets τ_m, are given in Appendix C.

5 Conclusion

We presented SAP3D, which edits a 3D asset by personalizing the generator to it and modulating the adapter's strength per token and per step, so that the edit is generated by the base model while the rest of the asset is reproduced by the personalized prior. Where the edit lies is read from the

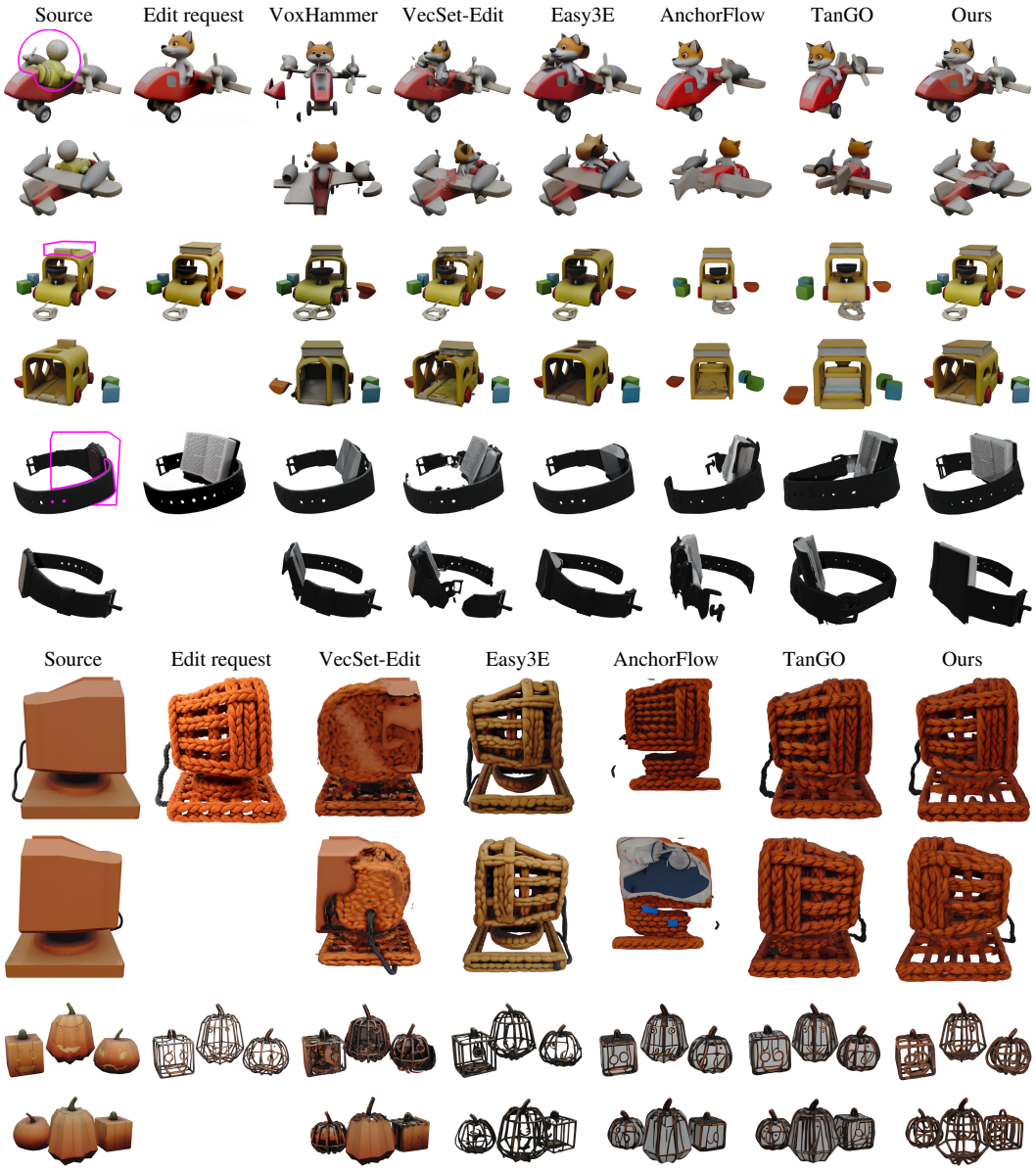

Figure 2: **Local (top) and global (bottom) edits.** Each case is shown from the camera of the edit request (upper row) and from the opposite side (lower row); for local edits, the outline on the source marks the edit region. The edit request is the single edited view that conditions every method. TanGO and AnchorFlow produce untextured meshes, which we texture with our pipeline. Global edits apply to the entire asset; we were unable to obtain results from VoxHammer on these, as on some of the local edits.

model itself, by a 2D probe of the image cross-attention and a 3D probe of the decoder's cross-attention, re-estimated throughout sampling as tokens drift. On Edit3D-Bench this yields the highest edit fidelity among the compared methods at numerically competitive preservation, and a uniform strength extends the same prior to global edits.

More broadly, treating an asset's identity as a prior that is applied to some tokens and withheld from others offers a route to editing in set-latent generators that does not depend on a fixed address per latent.

AI USE STATEMENT

We used generative AI tools to write code, collect and analyze experimental data, produce figures, and polish the writing. All AI-assisted work was reviewed by the authors, who take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our experiments use the public Edit3D-Bench benchmark and publicly released model weights. The user study asked participants only to compare pairs of rendered results. As with generative 3D models generally, an editing method could be applied to assets without their creators' consent; ours edits only an asset the user already holds.

REPRODUCIBILITY STATEMENT

The method is specified in Section 3 and Algorithm 1, with every setting in Appendix A; the evaluation protocol and metric definitions are in Appendix B, the ablation details in Appendix C, and the user study in Appendix E. All experiments use the public Edit3D-Bench benchmark and the public Hunyuan3D 2.1 weights.

REFERENCES

Jinxin Ai, Matthias Nießner, and Ziya Erkoç. Dreamedit3d: Personalization of multi-view diffusion models for 3d editing. *arXiv preprint arXiv:2605.16990*, 2026.

Omri Avrahami, Or Patashnik, Ohad Fried, Egor Nemchinov, Kfir Aberman, Dani Lischinski, and Daniel Cohen-Or. Stable flow: Vital layers for training-free image editing. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7877–7888. IEEE, 2025.

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.

Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22503–22513. IEEE, 2023.

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

Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 16251–16261, June 2025a.

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, 2024b.

Yiwen Chen, Zhihao Li, Yikai Wang, Hu Zhang, Qin Li, Chi Zhang, and Guosheng Lin. Ultra3d: Efficient and high-fidelity 3d generation with part attention, 2025b. URL https://arxiv.org/abs/2507.17745.

Yankuan Chi, Xiang Li, Zixuan Huang, and James M Rehg. Vinedresser3d: Agentic text-guided 3d editing. *arXiv preprint arXiv:2602.19542*, 2026.

Yusuf Dalva, Hidir Yesiltepe, and Pinar Yanardag. Lorashop: Training-free multi-concept image generation and editing with rectified flow transformers. *Advances in Neural Information Processing Systems*, 38:167841–167876, 2026.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

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.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NAQvF08TcyG.

Rohit Gandikota, Joanna Materzyńska, Tingrui Zhou, Antonio Torralba, and David Bau. Concept sliders: Lora adaptors for precise control in diffusion models. In *European Conference on Computer Vision*, pp. 172–188. Springer, 2024.

Inbar Gat, Dana Cohen Bar, Guy Levy, Elad Richardson, and Daniel Cohen Or. Shapeup: Scalable image-conditioned 3d editing. In *SIGGRAPH Conference Papers*, pp. 1–11, 2026.

Yuchao Gu, Xintao Wang, Jay Zhangjie Wu, Yujun Shi, Yunpeng Chen, Zihan Fan, Wuyou Xiao, Rui Zhao, Shuning Chang, Weijia Wu, et al. Mix-of-show: Decentralized low-rank adaptation for multi-concept customization of diffusion models. *Advances in neural information processing systems*, 36:15890–15902, 2023.

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.

Alec Helbling, Tuna Han Salih Meral, Benjamin Hoover, Pinar Yanardag, and Duen Horng Chau. ConceptAttention: Diffusion transformers learn highly interpretable features. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 22946–22963. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/helbling25a.html.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=_CDixzkzeyb.

Teng-Fang Hsiao, Bo-Kai Ruan, Yu-Lun Liu, and Hong-Han Shuai. Vecset-edit: Unleashing pre-trained lrm for mesh editing from single image. In *SIGGRAPH Conference Papers*, pp. 1–12, 2026.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Jingyu Hu, Bin Hu, Ka-Hei Hui, Haipeng Li, Zhengzhe Liu, Daniel Cohen-Or, and Chi-Wing Fu. Pegasus: 3d personalization of geometry and appearance. *arXiv preprint arXiv:2602.08198*, 2026a.

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 (CVPR)*, 2026b.

Junchao Huang, Xinting Hu, Shaoshuai Shi, Zhuotao Tian, and Li Jiang. Edit360: 2d image edits to 3d assets from any angle. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16618–16628. IEEE, 2025.

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Conference on Computer Vision and Pattern Recognition 2023*, 2023.

Chaehyun Kim, Heeseong Shin, Eunbeen Hong, Heeji Yoon, Anurag Arnab, Paul Hongsuck Seo, Sunghwan Hong, and Seungryong Kim. Seg4diff: Unveiling open-vocabulary semantic segmentation in text-to-image diffusion transformers. *Advances in Neural Information Processing Systems*, 38:71685–71724, 2026.

Zhe Kong, Yong Zhang, Tianyu Yang, Tao Wang, Kaihao Zhang, Bizhu Wu, Guanying Chen, Wei Liu, and Wenhan Luo. Omg: Occlusion-friendly personalized multi-concept generation in diffusion models. In *European conference on computer vision*, pp. 253–270. Springer, 2024.

Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu, Fuyun Wang, Huiwen Shi, Xianghui Yang, Qingxiang Lin, Jingwei Huang, Yuhong Liu, et al. Unleashing vecset diffusion model for fast shape generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2523–2533. IEEE, 2025.

Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu, Qingxiang Lin, Jingwei Huang, Chunchao Guo, and Xiangyu Yue. Lattice: Democratize high-fidelity 3d generation at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19982–19992, 2026.

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, 2026a.

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, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b.

Zhihao Li, Yufei Wang, Heliang Zheng, Yihao Luo, and Bihan Wen. Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling. *Advances in Neural Information Processing Systems*, 38:118582–118600, 2026b.

Siwoo Lim, Sunjae Yoon, Gwanhyeong Koo, Hyeonseo Yun, and Chang D Yoo. Tango: Training-free 3d editing via tangent-space guidance and optimization. *arXiv preprint arXiv:2607.14927*, 2026.

Fangfu Liu, Hanyang Wang, Weiliang Chen, Haowen Sun, and Yueqi Duan. Make-your-3d: Fast and consistent subject-driven 3d content generation. In *European Conference on Computer Vision*, pp. 389–406. Springer, 2024.

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*, 2026a.

Yaoli Liu, Yao-Xiang Ding, and Kun Zhou. Freefuse: Multi-subject loRA fusion via adaptive token-level routing at test time. *Transactions on Machine Learning Research*, 2026b. ISSN 2835-8856. URL https://openreview.net/forum?id=Bvela8OAMc.

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

Xingyu Miao, Haoran Duan, Yang Long, and Jungong Han. Rethinking score distilling sampling for 3d editing and generation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=1dZgzGTZEO.

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.

Luchao Qi, Jiaye Wu, Annie N Wang, Shengze Wang, and Roni Sengupta. My3dgen: A scalable personalized 3d generative model. In *2025 ieee/cvf winter conference on applications of computer vision (wacv)*, pp. 961–972. IEEE, 2025.

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.

Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Ben Mildenhall, Nataniel Ruiz, Shiran Zada, Kfir Aberman, Michael Rubenstein, Jonathan Barron, Yuanzhen Li, and Varun Jampani. Dreambooth3d: Subject-driven text-to-3d generation. *ICCV*, 2023.

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

Ville Satopää, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *International Conference on Distributed Computing Systems Workshops*, pp. 166–171, 2011.

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.

Etai Sella, Hao Phung, Nitay Amiel, Or Litany, Or Patashnik, and Hadar Averbuch-Elor. Prox-e: Fine-grained 3d shape editing via primitive-based abstractions. In *SIGGRAPH Conference Papers '26*, SIGGRAPH Conference Papers '26, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400725548. doi: 10.1145/3799902.3811141. URL https://doi.org/10.1145/3799902.3811141.

Vera Soboleva, Aibek Alanov, Andrey Kuznetsov, and Konstantin Sobolev. T-lora: Single image diffusion model customization without overfitting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 9051–9059, 2026.

Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Türe. What the daam: Interpreting stable diffusion using cross attention. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5644–5659, 2023.

Tencent Hunyuan3D Team. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material, 2025a.

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

Tencent Hunyuan3D Team. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details, 2025c. URL https://arxiv.org/abs/2506.16504.

Junjiao Tian, Lavisha Aggarwal, Andrea Colaco, Zsolt Kira, and Mar Gonzalez-Franco. Diffuse, attend, and segment: Unsupervised zero-shot segmentation using stable diffusion. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3554–3563. 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.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, Yao Yao, et al. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. *Advances in Neural Information Processing Systems*, 38:170778–170804, 2026.

Ruihao Xia, Yang Tang, and Pan Zhou. Towards scalable and consistent 3d editing. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=pXymBD1UZg.

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.

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, et al. 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.

Jinqi Xiao, Qing Yan, Liming Jiang, Zichuan Liu, Hao Kang, Shen Sang, Tiancheng Zhi, Jing Liu, Cheng Yang, Xin Lu, et al. Instructmole: Instruction-guided mixture of low-rank experts for multi-conditional image generation. *arXiv preprint arXiv:2512.21788*, 2025.

Yang Yang, Wen Wang, Liang Peng, Chaotian Song, Yao Chen, Hengjia Li, Xiaolong Yang, Qinglin Lu, Deng Cai, Xiaofei He, et al. Lora-composer: Leveraging low-rank adaptation for multi-concept customization in training-free diffusion models. *IEEE Transactions on Image Processing*, 34:8145–8158, 2025.

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.

Junliang Ye, Kenkun Liu, Guocun Wang, Yang Li, Yansong Qu, Chunshi Wang, Jingwei Xu, Yunhan Yang, Zibo Zhao, Jiachen Xu, et al. Hunyuan3d-buffalo 1.0: A unified multimodal model for scalable 3d generation, understanding, and editing. *arXiv preprint arXiv:2608.02711*, 2026a.

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, 2026b.

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 (ECCV)*, 2026.

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

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

Zhixing Zhang, Ligong Han, Arnab Ghosh, Dimitris N Metaxas, and Jian Ren. Sine: Single image editing with text-to-image diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6027–6037, 2023b.

Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, BIN FU, Tao Chen, Gang YU, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xmxgMij3LY.

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 (CVPR)*, pp. 14387–14397, June 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 Implementation Details

Sampling uses 30 Euler steps with classifier-free guidance at scale 5.0, and meshes are extracted at grid resolution 384. The adapter has rank 16 and scaling factor 1. It is trained with AdamW at learning rate 7×10^{-6} and batch size 4 for up to $5{,}000$ steps, retaining the checkpoint with the lowest validation Chamfer distance, on 250 renders per asset and 200k surface points, of which 80k are randomly selected and encoded at each step. The 2D probe averages head 6 of layer 15 and head 0 of layer 14. The schedule constants and probe thresholds were selected empirically: τ_m and τ_d fall at steps 8 and 16 of the 30, $\alpha_{\text{pre}} = 0.71$ and $\alpha_{\text{out}} = 1$, the 3D probe uses $Q = 1{,}000$ query points, and the knee factors are $\kappa^{\text{2D}} = 2.54$ and $\kappa^{\text{3D}} = 0.83$. Algorithm 1 summarizes one edit.

A.1 Texturing

The benchmark's appearance metrics require a textured mesh, while SAP3D, like the VecSet generator it builds on, produces geometry. We texture the edited mesh in two parts. Outside the 3D region, the source asset's texture is transferred onto the edited surface from the nearest point of the source surface; inside it, the texture is generated by Hunyuan3D-Paint (Tencent Hunyuan3D Team, 2025a), conditioned on the edited view, and its colors are corrected to match the edited view.

B Evaluation Protocol

B.1 What the Benchmark Measures

Edit3D-Bench was designed primarily to measure preservation, and it does so thoroughly: of its eight metrics, seven compare the prediction with the unedited source, and four of those are restricted to the region outside the edit. The edited view that specifies each edit is consumed by every method as its condition, but it is not referenced by any metric, and no ground-truth edited asset is provided. The one metric that relates the prediction to the requested edit, CLIP-T, compares it with a text caption of the edited object.

A consequence is that a prediction which performs no edit at all, returning the source asset unchanged, scores better than every published method on CD, PSNR, SSIM, LPIPS and DINO-I (0.0053, 84.55 dB, 1.000, 0.000 and 1.000 respectively), and removing the edit region from the source improves CD further: on one asset, deleting 51.2% of its faces lowers CD from 0.00331 to 0.00307. This means that the benchmark's metrics cannot be read as a measure of edit quality, and that a method is rewarded on most of them for editing less. We therefore complement them with an edit-fidelity metric and a preservation metric.

B.2 Chamfer Distance and Its Noise Floor

The benchmark's CD samples $20{,}480$ points on each surface and downsamples them to a 0.01 voxel grid, settings the evaluation code does not expose. At these settings, two independent samplings of the same source mesh give a mean CD of 0.005345 over all 300 cases, in the benchmark's normalized units: this is the value an exact copy of the source receives. The floor varies with the object's surface area, from 0.0028 to 0.0089 across the benchmark, and it combines with the true error in quadrature rather than adding to it, so it cannot be subtracted. Rescoring identical meshes moves CD by 0.42%

of its value. The sampling also compresses real differences between predictions: on three cases where the exact distance degrades by a factor of 3.5 to 4.2, CD reports a factor of 1.5 to 1.8.

The exact point-to-surface distance of Section 4.1 removes both effects. Its floor, measured the same way, is 7.2×10^{-9}, and resampling moves it by 0.16% to 0.99% of its value. We report CD for comparability with prior work and base our conclusions about preservation on the exact distance.

B.3 Properties of the Image Metrics

The following are properties of the released evaluation code. They apply equally to every method, including ours, and we retain the code unmodified so that our numbers remain comparable with published ones.

- **DINO-I is computed on whole renders**, including the edit region, and so decreases for any visible edit. 81% of the DINO-I difference between our unedited and edited predictions arises inside the edit region.
- **Masked LPIPS is influenced by the excluded region.** Its spatial map is computed from features with a receptive field of about 99 pixels, so content inside the edit region reaches scored pixels; 53% of the LPIPS difference between the same two predictions arises inside the edit region.
- **About 90% of the pixels averaged by the image metrics are background**, which matches trivially and compresses differences between methods.
- **CLIP-T barely separates an edit from its absence.** On the annotated views, the edited view itself scores 0.294 against its caption on average, where the unedited view scores 0.267, on a nominal scale of -1 to 1. On 63 of the 300 cases the unedited view scores higher than the edited view.

B.4 FID and FVD

The three edits of an asset share its 16 source renders as their reference, so the reference side of FID holds at most $1{,}600$ distinct images (100 assets $\times$ 16 views) against $4{,}800$ predicted renders, while the Inception features it compares have $2{,}048$ dimensions. The covariance estimate is therefore rank-deficient, and the value depends strongly on the number and choice of assets: between renders of two disjoint halves of the source assets, whose true distance is zero, FID measures 105 to 208 depending on the number of images, and scoring a method on a random half of the assets raises its FID by 50% to 60%. FVD compares one turntable video per asset in a $1{,}024$-dimensional feature space and is affected more severely; we do not report it. Table 2 reports FID, computed with the benchmark's code, for completeness. Although its values should not be read as absolute, its ranking is stable to the choice of assets: over 30 random halves of the assets, every pair of methods keeps its order in all 30, except TanGO and AnchorFlow, which keep theirs in 21.

Table 2: **FID on Edit3D-Bench** (↓), over all 300 cases. †Over the 260 cases on which it produced an output.

	VoxHammer†	Easy3E	VecSet-Edit	TanGO	AnchorFlow	SAP3D
FID	31.60	**22.45**	36.59	39.87	41.25	<u>28.04</u>

B.5 Our Edit-Fidelity Metric

Outside the mask, the edited view is nearly identical to the rendered source view: over the object's pixels, the two differ by 2.1% of the intensity range on average, against 24.2% inside the mask. The metric therefore compares the prediction with the edited view over the square bounding the mask, with every pixel outside the mask set to the same gray in both images, so that only the edited content affects the score.

We compare class-token embeddings of the masked images rather than patch tokens at masked positions. In DINOv2's final layer every patch token has attended to the entire image, so a token

at a masked position still reflects content outside the mask: over the unmasked area, where the two images agree closely, corresponding patch tokens reach a cosine similarity of only 0.75. Masking the input is what confines the comparison to the edit.

The metric observes the edit from the camera of the edited view only; like every metric on this benchmark, it does not observe the far side of the edit.

C Additional Ablations

Table 3: **Ablations** on 150 cases of Edit3D-Bench. Edit fidelity is the distance to the edited view under DINOv2 and CLIP; preservation is the exact point-to-surface distance outside the 3D region. Means over the same 150 cases; lower is better throughout.

	Edit fidelity			
Variant	DINOv2 ↓	CLIP ↓	Preservation ↓	CD ↓
SAP3D (full)	0.250	0.094	0.0119	0.0186
Personalization				
w/o personalization ($s = 0$)	0.290	0.111	0.0418	0.0493
uniform strength $s = 0.5$	0.273	0.100	0.0171	0.0237
uniform strength $s = 0.75$	0.327	0.109	0.0080	0.0144
uniform strength $s = 1$	0.421	0.133	0.0043	0.0105
Locating the edit				
2D probe only	0.264	0.094	0.0106	0.0175
3D probe only	0.260	0.098	0.0099	0.0165
region fixed at τ_m	0.243	0.091	0.0121	0.0188
Schedule				
adapter retained after τ_d	0.244	0.096	0.0115	0.0186

Locating the edit. Averaged over the benchmark, the localization choices change the metrics less than personalization does, since their effect is concentrated on a minority of cases. We therefore also compare each variant with the full method case by case, taking a second run of the full method as the reference for noise: between the two runs, edit fidelity differs by more than 0.1 on 6.7% of the cases. With the 2D probe alone, edit fidelity is worse than with the union by more than 0.1 on 8.7% of the cases and better on 4.7%; with the 3D probe alone, it is worse on 9.3% and better on 6.7%. Either probe alone selects fewer tokens than the union and preserves slightly better. Fixing the region at τ_m, rather than re-estimating it at every step of the selective phase, leaves both metrics within their noise on average; visually, however, we observed several cases in which re-estimation clearly improved the result.

Schedule. The step τ_m at which we begin to selectively disable the adapter governs a trade-off of its own, which we measured by sweeping it across configurations of the remaining settings and two seeds, on all 300 cases. The later the selective phase begins, the better the asset is preserved: the median preservation error falls steadily from 0.028 at step 2 to 0.006 at step 14. Edit fidelity is best, and nearly constant, for τ_m between steps 5 and 8 (medians 0.264 to 0.268); it declines gradually at steps 9 and 10 (0.277 and 0.290) and sharply thereafter, to 0.387 at step 14, suggesting that by then the coarse shape is largely settled and leaves the edit less room to form. We take τ_m at step 8, the latest before this decline. Retaining the adapter after τ_d likewise leaves both metrics within their noise; we keep the final phase as an option, since on several cases it visibly improved the result in ways the metrics do not capture.

C.1 Training the Adapter

Adapter placement. We trained adapters for all 100 assets under four placements, with otherwise identical training settings and, in this comparison, the posterior mean as the training target (Table 4). At each adapter's best validation checkpoint, the median validation CD over the 100 assets is 0.0164 with the feed-forward layers alone, 0.0208 when cross-attention is also adapted, 0.0247 with self-attention, and 0.0205 with both; with a posterior sample as the target, our final configuration reaches

0.0088. Adapting both attention types also destabilizes training: 69 of the 100 adapters produced completely broken geometry at some checkpoint during training, against 5 of 100 for the feed-forward layers alone.

Training target. Encoding each training shape to a latent drawn from the VAE posterior, rather than to its mean, lowers the best validation CD on 81 of the 100 assets and raises it on none. The median falls from 0.0164 to 0.0088, and the CD of a typical asset is roughly halved (median per-asset ratio 1.96).

Table 4: **Training ablations** over the 100 benchmark assets. Validation CD at each adapter's best checkpoint, median over assets; the last column counts adapters that produced completely broken geometry at some checkpoint during training.

Adapted layers	Training target	Median best val. CD ↓	Broken during training ↓
feed-forward (ours)	posterior sample	**0.0088**	**0/100**
feed-forward	posterior mean	0.0164	5/100
+ cross-attention	posterior mean	0.0208	7/100
+ self-attention	posterior mean	0.0247	3/100
+ both	posterior mean	0.0205	69/100

C.2 Uniform Adapter Strength on Local Edits

Figure 3 and Table 5 follow three of the local edits of Figure 2 across uniform adapter strengths. On each, lowering the strength frees the edit but releases the rest of the object with it, and raising it restores the object but suppresses the edit; SAP3D sits between these regimes. On the airplane and the watch strap its edit fidelity is better than at any uniform strength; on the figurine only the base model ($s = 0$) edits better, at 25 times its preservation error.

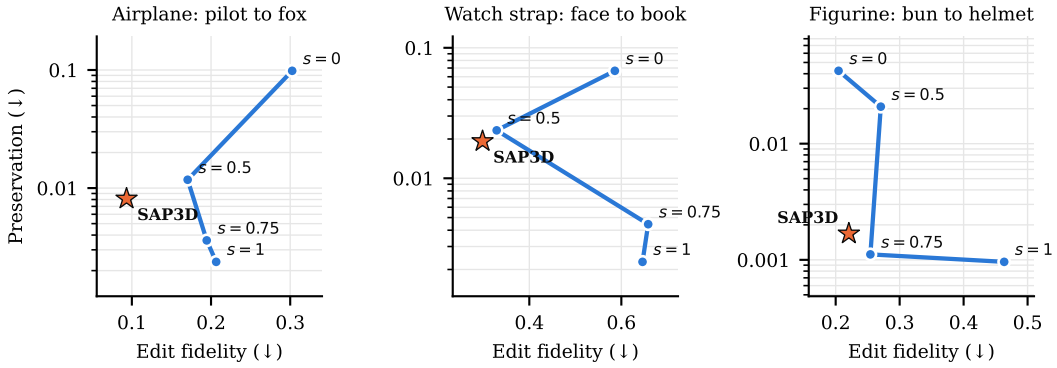

Figure 3: **Uniform adapter strength on three local edits.** Edit fidelity against preservation (model units, logarithmic scale) for the adapter held at strength s for every latent token at every step, and for SAP3D, which sets the strength per token.

Table 5: **Uniform adapter strength on three local edits**: edit fidelity / preservation, both lower is better.

Case	$s = 0$	$s = 0.5$	$s = 0.75$	$s = 1$	SAP3D
Airplane	0.302 / 0.0984	0.170 / 0.0118	0.195 / 0.0036	0.206 / 0.0024	0.093 / 0.0081
Watch strap	0.586 / 0.0666	0.329 / 0.0233	0.658 / 0.0044	0.646 / 0.0023	0.298 / 0.0192
Figurine	0.204 / 0.0424	0.271 / 0.0208	0.254 / 0.0011	0.463 / 0.0010	0.221 / 0.0017

C.3 The Probes

Heads of the 2D probe. We ranked the 336 cross-attention heads of the DiT (21 layers of 16 heads) by how closely their attention from tokens to image patches follows the object: the share of attention falling on the object's foreground patches and the spatial concentration of the patches each token attends to. Heads (15,6) and (14,0) ranked first and second across seeds, and the 2D probe averages the two. The 3D probe needs no such selection: the decoder's cross-attention from query points to tokens is a single layer, and the uniform average of its heads localized reliably without selection.

The signal of the 3D probe.

We compared the decoder's cross-attention (Section 3.4) with a gradient-based signal, the sensitivity of the decoded occupancy inside Ω to each token. Let $o(p)$ denote the occupancy logit that the VAE decoder assigns to a point p given the current prediction $\hat{x}_1$. The sensitivity of token i is the norm of the gradient of the total occupancy at the query points with respect to that token,

$$g^i = \Big\| \frac{\partial}{\partial \hat{x}_1^i} \sum_{q=1}^{Q} o(p_q) \Big\|_2, \tag{6}$$

where the derivative passes through the whole decoder, including the self-attention layers that mix the tokens before its cross-attention. It measures how strongly a change of token i would alter the shape inside Ω, and yields a single score per token, to which the knee rule of Section 3.4 is applied as to a single nominator; the cross-attention signal instead lets every query point nominate its own tokens. Computing g requires a backward pass through the decoder at every probed step, where cross-attention requires only the forward pass.

Sweeping each signal across configurations of the remaining settings and two seeds, on all 300 cases, with about 100 configurations per signal, cross-attention gives the better median edit fidelity (0.276 against 0.286) and the occupancy sensitivity the better median preservation (0.0096 against 0.0125). At each signal's best configuration for edit fidelity the two are level (0.257 against 0.258), and cross-attention preserves better (0.0123 against 0.0159).

D Token Drift

To measure how far tokens move during sampling, we generate each of 10 benchmark assets from its source render with its adapter at full strength, so that the generated shape is the source itself, and compute for every token the VAE decoder's cross-attention from $16{,}384$ surface points, on the clean latent predicted at each step. Over steps 11 to 20, a median of 24.4% of the tokens (18.2% to 29.0% across the assets) have their attention centroid in a different one of eight regions into which we partition the surface than at the final step, against 1.8% between two independent draws of the surface points. Figure 4 follows one such token in both of the model's spatial signals.

E User Study

The study comprised 28 cases, 16 local and 12 global edits, each pairing our result with that of one baseline. Every participant saw the cases in a random order, and the two results of each case on sides drawn at random; each result, like the source asset, was shown as a rotating render, beside the source view and the edit request (Figure 6). Participants answered the question "Which result better applies the requested edit, while otherwise staying faithful to the source?" by choosing either result or a tie. Eighteen participants took part, thirteen of whom answered all 28 cases, and 398 votes were kept after three duplicate submissions were removed. Left and right were each chosen 179 times among the votes that were not ties. The interval reported in Section 4.2 resamples participants and cases jointly, since the votes of one participant, or on one case, are not independent.

F Additional Results

Figures 7 to 10 show 18 further local edits and Figures 11 to 13 the remaining 10 global edits of our test set.

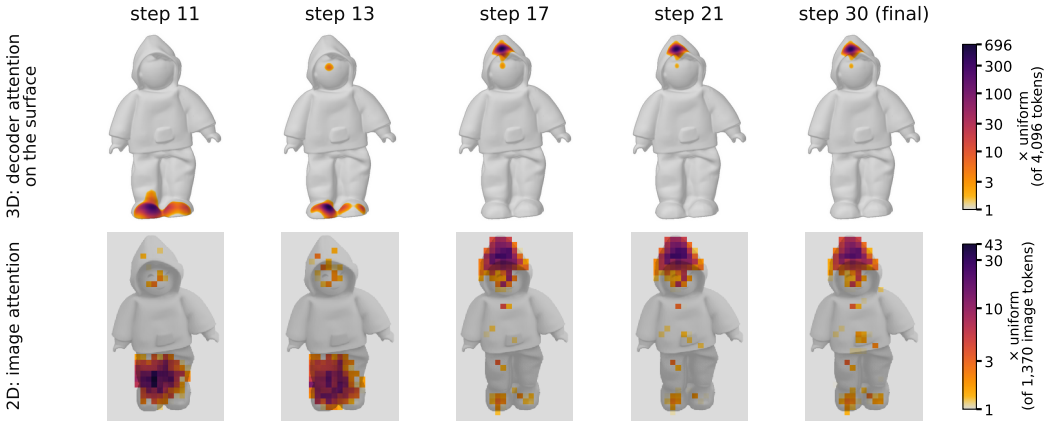

Figure 4: **Token drift.** Top: the VAE decoder's cross-attention to one tracked token, computed on the clean latent predicted at the indicated step (of 30) and drawn on the final generated surface. Bottom: the same token's cross-attention to the patches of the conditioning image in the denoiser, averaged over the two heads of the 2D probe; background patches are not drawn. Both are shown as multiples of the level under evenly spread attention. The token is sharply localized at step 11 (peak 360 times that level), yet between steps 13 and 17 both its surface attention and its image attention move from the feet to the hood, over 73% of the object's bounding-box diagonal (the 99.99th percentile among the 4,096 tokens of each of 10 objects).

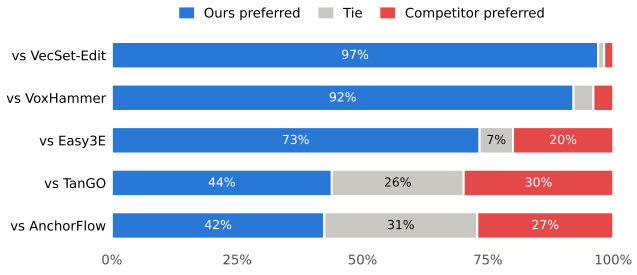

Figure 5: **User study.** Share of votes preferring our result, declaring a tie, or preferring the baseline, for each baseline.

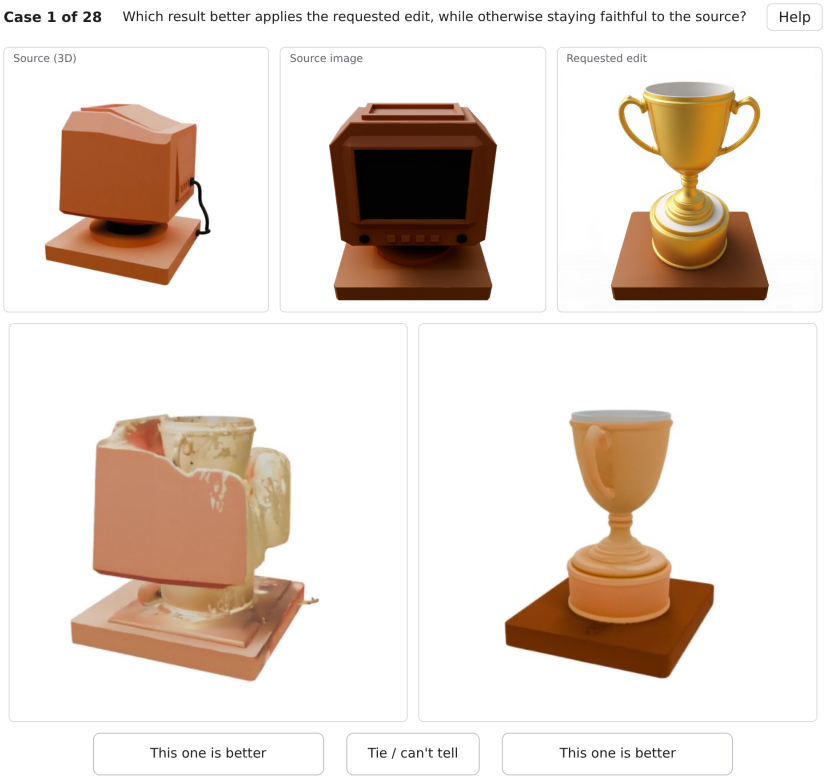

Figure 6: **The interface of the user study.** The source asset (rotating), its rendered view and the edit request above; the two results below, on randomly assigned sides.



Figure 7: **Additional local edits (1/4).** Layout as in Figure 2.



Figure 8: **Additional local edits (2/4).** Layout as in Figure 2.



Figure 9: **Additional local edits (3/4).** Layout as in Figure 2.



Figure 10: **Additional local edits (4/4).** Layout as in Figure 2.

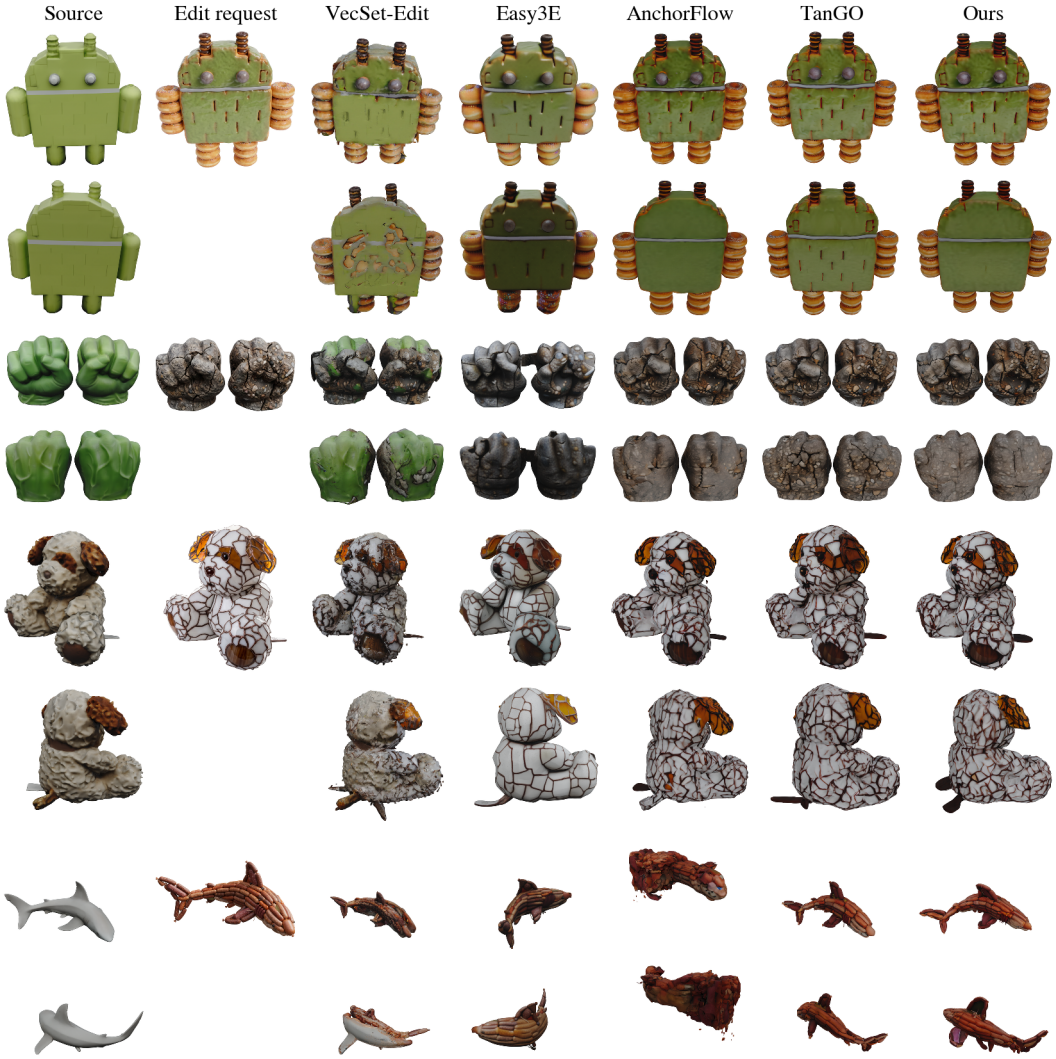

Figure 11: **Additional global edits (1/3).** Layout as in Figure 2.



Figure 12: **Additional global edits (2/3).** Layout as in Figure 2.



Figure 13: **Additional global edits (3/3).** Layout as in Figure 2.