This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

CraftMesh: High-Fidelity Generative Mesh Manipulation via Poisson Seamless Fusion

James Jincheng Hu[1] Yuxiao Wu[2] Youcheng Cai[2,*] Ligang Liu[2]
[1] Hefei Thomas School [2] University of Science and Technology of China
contact@jameshu.org wuyx2020@mail.ustc.edu.cn caiyoucheng@ustc.edu.cn lgliu@ustc.edu.cn

"Give the deer wings" *Drag-based editing*

Figure 1. Mesh editing results produced by CraftMesh. CraftMesh is a versatile 3D mesh editing framework that enables users to perform text-based and drag-based editing, and delivers high-quality outputs even in challenging editing scenarios.

Abstract

Controllable, high-fidelity mesh editing remains a significant challenge in the domain of 3D content creation. Existing generative methods often struggle with complex geometries and fail to preserve fine-scale details. We propose CraftMesh, a novel framework for high-fidelity generative mesh manipulation based on Poisson Seamless Fusion. We decompose mesh editing into a pipeline that leverages the strengths of 2D image editing and 3D generation models: we first edit a 2D reference image, then generate a 3D mesh corresponding to the edited region, and fuse it seamlessly into the original mesh through a Geometry and Texture Fusion method. We introduce two core techniques: Poisson Geometric Fusion, which utilizes a hybrid SDF/Mesh representation with normal blending to achieve harmonious geometric integration, and Poisson Texture Harmonization for visually consistent texture blending. We demonstrate state-of-the-art structural consistency, geometric fidelity, and texture quality in challenging editing scenarios.
Project page: https://jameshu.org/CraftMesh

* Corresponding author.

1. Introduction

In recent years, the rapid advancement of 3D generative models [20, 24, 46, 55] has allowed the generation of high-quality 3D content from text prompts or images. These advances have substantially accelerated downstream applications in video games, augmented and virtual reality (AR/VR) [48], robotics [27], and digital manufacturing [25]. Despite these notable achievements in 3D generation, the challenge of controllable 3D editing remains an open problem. Current 3D generation frameworks are designed to reconstruct 3D models from 2D images but provide limited flexibility for localized modifications.

Recent research has been done on neural field [34] editing, encompassing appearance-guided and text- or image-driven approaches [17, 50, 59]. These methods remain restricted to appearance-level modifications and cannot inherently support geometric manipulations of explicit surface meshes. In contrast to the rapidly expanding body of work on neural field editing, mesh-based generative editing has received considerably less attention, despite the fact that meshes remain the most widely adopted representation in professional 3D content creation pipelines. In practical design workflows, artists and engineers iteratively refine meshes with precise part-level control to satisfy both aesthetic and functional requirements while avoiding un-

intended alterations to unrelated geometry. This demand underscores the necessity for editing methods that enable fine-grained controllability while faithfully preserving the connectivity of the original model.

Existing generative mesh editing methods can be broadly categorized into two principal paradigms: score distillation sampling (SDS)-based approaches and multi-view diffusion (MVD)-based approaches. SDS-based methods [2, 26] optimize geometry using SDS losses and better retain the original structure, but they lack multi-view consistency, often producing oversimplified or distorted results. MVD-based methods [3, 5, 23] synthesize multi-view edits followed by a reconstruction step, but fail to preserve the geometry and texture of the original model.

To overcome these limitations, we harness the capabilities of generative models by reframing editing tasks as generative processes. We introduce an **image editing–mesh generation–seamless fusion** pipeline that fully leverages the strengths of 2D models for image editing and 3D models for high-quality mesh generation. The framework edits the rendered image of the target region, generates high-quality 3D content consistent with the edited view, and finally integrates the generated geometry and texture into the original mesh. However, a central challenge is achieving seamless geometric and textural consistency between the generated mesh and the source model.

Classical Poisson Mesh Editing [56] operates in the coordinate domain but suffers from two fundamental limitations. First, incompatible connectivity—when transferring gradients from another mesh, this requires an unrealistic one-to-one vertex correspondence. Second, incompatible gradients—gradients in the coordinate domain are discontinuous at boundaries. SeamlessNeRF [16] and GaussianStitching [15] blend radiance fields via Poisson-based optimization but cannot be directly applied to meshes with explicit surfaces. Directly solving the Poisson equation in 3D volumetric space incurs a computational complexity of $O(n^3)$, making it computationally expensive.

To this end, we present a novel high-fidelity generative mesh manipulation framework, termed **CraftMesh**, enabling complex and controllable editing while preserving geometry and texture (see Fig. 1). We employ a 2D image editing model to modify reference images derived from the original mesh, generate corresponding meshes and extract the edited regions as edited region meshes, which are then fused into the original mesh via our core method, **Geometry and Texture Fusion**, an SDF-domain Poisson fusion method. We adopt a hybrid SDF/Mesh representation, in which geometry and texture are optimized in implicit and explicit forms, respectively. Specifically, We propose a *Poisson Geometry Blending* strategy that enables natural gradient transitions and seamless geometric blending between the edited and original meshes. We further propose a *Poisson Texture Harmonization* strategy to enable seamless texture fusion between the edited and original meshes in texture space. Experimental results demonstrate the superiority of our approach in achieving high-fidelity mesh editing. Additionally, we conduct further experiments using a drag-based method for controlled image manipulation, demonstrating the versatility of our framework (see Fig. 5).

Our contributions are summarized as follows:

- We introduce a framework that reformulates mesh editing as an **image editing–mesh generation–seamless fusion** pipeline integrating 2D and 3D generative models.
- We propose Geometry and Texture Fusion, an SDF-domain Poisson fusion method, in which we introduce **Poisson Geometry Blending** to enable seamless geometric integration and **Poisson Texture Harmonization** to achieve seamless texture blending across edited regions.

2. Related Work

3D Generation Models. Recent advances in 2D diffusion models [19, 37, 41] have significantly accelerated modern 3D content creation.

SDS-based Approaches. Score Distillation Sampling (SDS) bridges 2D diffusion priors and 3D optimization. DreamFusion [39] first optimized NeRF under text-to-image diffusion guidance, followed by Magic3D [29], which introduced a two-stage low-to-high resolution refinement. Later, LucidDreamer [28] further improved stability and fidelity through interval score matching, whereas ProlificDreamer [51] incorporated a variational SDS formulation to improve diversity and quality. These methods successfully bridge 2D diffusion priors and 3D optimization, although they frequently remain computationally intensive.

MVD-based Approaches. Multi-view diffusion (MVD) enforces multi-view consistency during image synthesis to reconstruct 3D assets. SyncDreamer [31] and MVDream [44] exploit multi-view diffusion for geometrically coherent text-to-3D generation. Wonder3D [32] and One-2-3-45++ [30] further extend this paradigm to single-image 3D generation. SV3D [49] uses latent video diffusion. Instant3D [21] achieves fast reconstructions from sparse views.

3D Native Generation Approaches. More recently, researchers have shifted toward training generative models directly on large-scale 3D datasets, thereby overcoming the inherent limitations of 2D priors. Foundational resources such as Objaverse [11], Objaverse-XL [10], and OmniObject3D [53] provide millions of diverse, well-annotated 3D objects, enabling scalable learning of both geometry and texture. Clay [57] demonstrates controllable large-scale text-to-3D generation. 3DTopia-XL [6] scales primitive-based diffusion approaches, achieving improved generalization across diverse categories. Trellis [54] proposes structured 3D latent representations that improve scalability and versatility, making generative models more efficient at cap-

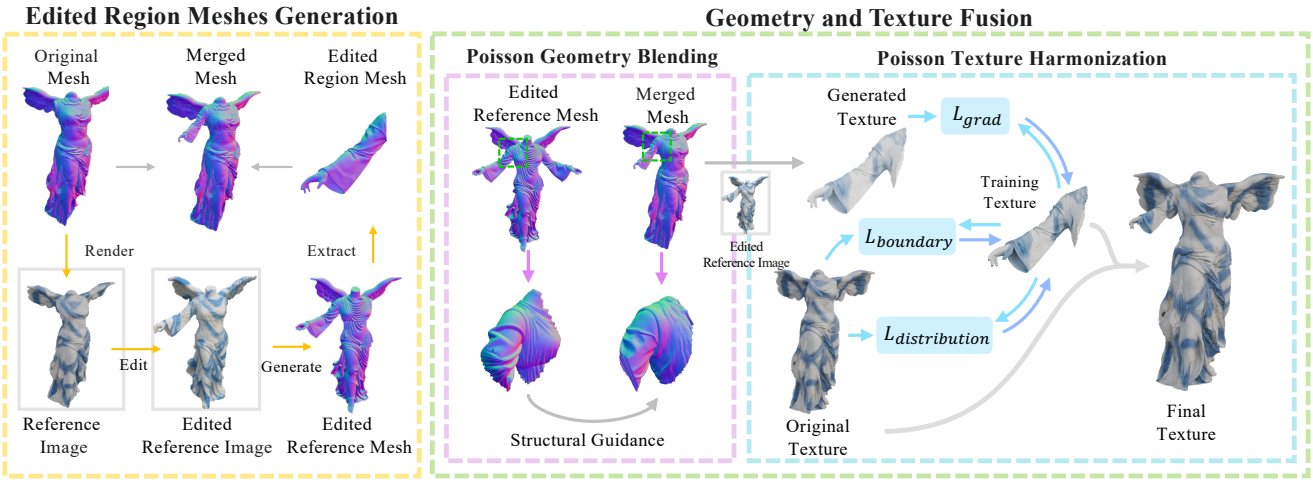


Figure 2. An overview of the CraftMesh framework. Our framework follows an **image editing–mesh generation–seamless fusion** pipeline that fully leverages the strengths of 2D models for image editing and 3D models for high-quality mesh generation. First, **Edited Region Mesh Generation** produces meshes for the editing region. Then, **Poisson Geometry Blending** achieves natural geometric transitions through normal blending. Finally, **Poisson Texture Harmonization** performs texture fusion to seamlessly color the edited regions.

turing complex shapes. Hunyuan3D 2.5 [20] achieves high-fidelity geometry and physical-based rendering (PBR) texture. These native 3D models mark a shift toward more direct, efficient, and realistic 3D generation.

Generative Mesh Editing. Most existing generative editing approaches mainly focus on implicit representations [7, 17, 42, 50, 59]. Although these methods achieve promising results, they are constrained by implicit representations and thus cannot be applied to mesh-level editing. In this paper, we focus on generative mesh editing, which can be broadly categorized into two paradigms: SDS-based editing and MVD-based editing.

SDS-based Editing. SDS-based editing approaches extend the concept of Score Distillation Sampling (SDS) loss to editing tasks by guiding mesh optimization using pre-trained diffusion priors. FocalDreamer [26] introduces focal-fusion assembly for localized text-driven 3D editing, thereby enabling controllable, region-specific modifications. MagicClay [2] first optimizes an SDF with Score Distillation Sampling. Then, the changes are lifted to the editing mesh through vertex optimization.

MVD-based Editing. MVD-based editing methods utilize multi-view diffusion (MVD) to maintain consistency across different views, thereby enhancing fidelity. MVEdit [5] adapts generic 3D diffusion priors for controlled multi-view editing. CMD [23] proposes CondMV, which takes a target image and multi-view conditions to generate multi-view consistent edits. Instant3dit [3] fine-tunes 2D diffusion models for multi-view consistent inpainting. MaskedLRM [14] leverages large reconstruction models with masked conditioning for efficient mesh editing.

However, these methods are unable to edit complex models or achieve high-fidelity mesh manipulation. In this paper, our method fully capitalizes on the strengths of 2D and 3D generative models. By employing a Poisson seamless fusion strategy, our approach merges generated edited region meshes with the original mesh, thereby achieving high-fidelity and structurally consistent mesh manipulation.

Seamless Editing. Pioneered by Poisson Image Editing [38], seamless editing is a classical topic in computer graphics and digital image processing. The primary goal is to achieve smooth and imperceptible transitions in images or textures, thus maintaining visual consistency [1, 4, 18]. Poisson Mesh Editing [56] applies the Poisson equation to mesh editing, enabling geometric merging through gradient-field manipulation. However, these methods solve the Poisson equation on the coordinate domain, which results in discontinuous gradients at the merging boundary that create artifacts. Our method solves the Poisson equation on the gradient domain.

Recently, SeamlessNeRF [16] achieves seamless blending of neural radiance fields through a Poisson based optimization, focusing on radiance field merging. Gao et al. [15] advances example-based 3D modeling by introducing 3D Gaussian stitching. While these works offer smooth merging in radiance fields or 3D Gaussians, explicit mesh geometry is not considered. In this paper, we consider both geometry and texture, ensuring seamless fusion between the edited region mesh and the original mesh.

3. Method

We propose CraftMesh, a high-fidelity generative mesh manipulation framework that integrates 2D image editing, 3D mesh generation, and Poisson-based fusion. Fig. 2 illustrates the overall workflow. We first edit reference images using 2D image editing models to achieve user-intent-consistent modifications, followed by generating edited region meshes with 3D generative models. Next, we employ a Poisson Seamless Fusion strategy to integrate the edited-region mesh into the original mesh, ensuring geometric and texture consistency through Poisson Geometry Blending and Poisson Texture Harmonization.

3.1. Edited Region Meshes Generation

Recent image generation models have demonstrated remarkable performance in controllable image editing, producing semantically aligned and globally consistent results. Representative examples include FLUX Kontext [19], Qwen-Image [52], and Gemini 2.5 [9], which can effectively preserve content structure while introducing new details. Compared with direct 3D editing, these 2D approaches are lightweight, controllable, and well-suited for performing high-quality object-centric image manipulation. On the other hand, recent progress in 3D mesh generation, such as Craftsman3d [24] and Hunyuan3D [20], has enabled the production of 3D meshes with unprecedented structural fidelity, intricate details and textural realism. However, existing 3D mesh editing methods lag significantly behind. For instance, Instant3dit [3] fine-tunes multi-view diffusion models to regenerate 3D content, but often struggles with consistency. Similarly, FocalDreamer [26] and MagicClay [2] are limited to basic tasks and frequently yield simple results in the edited region.

To bridge this gap, we leverage the complementary advantages of 2D image editing models and 3D mesh generation models. Specifically, we first generate **Edited Region Meshes**, which we later fuse with the original mesh (see Sec. 3.2.1, Sec. 3.2.2). The generation proceeds as follows: From the original mesh, we render the reference image, which is edited according to the user's intent of mesh editing. Users can employ a variety of tools to perform the edit, including image editing models [19], software suites, or other creative instruments. This flexibility in choosing the image-editing backend provides users with greater control and adaptability to achieve the desired outcome. Next, we lift the edited reference image to 3D using image-to-mesh generation models [20, 33, 54]. From this edited reference mesh, we extract the components within the editing region to form the edited region mesh. Optionally, we can perform an additional image editing step on the edited reference image to extract an image of the editing region, which is used by image-to-mesh models to generate the edited region mesh with higher fidelity.

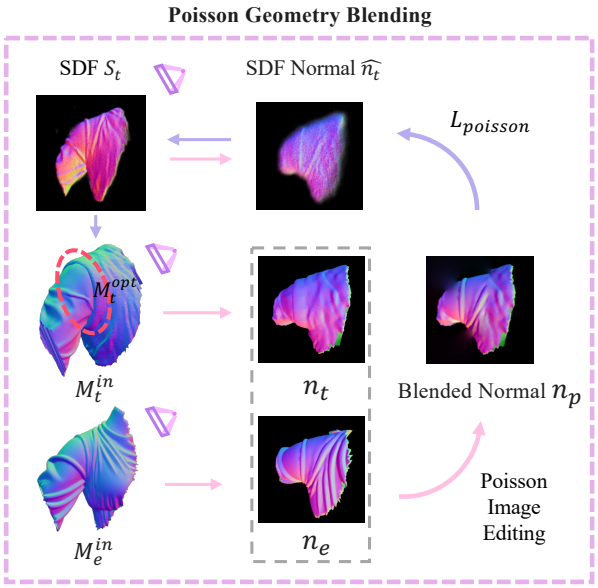


Figure 3. Details of Poisson Geometry Blending. We employ a hybrid SDF/Mesh representation, which is optimized to be harmonious using blended normals.

Our core idea is to seamlessly fuse the edited region mesh into the original mesh while using the edited reference mesh as geometric guidance. CraftMesh offers: (1) control of mesh editing through image editing, meaning no requirement of manually specifying editing locations in 3D, unlike FocalDreamer [26], MagicClay [2], and Instant3dit [3], making editing more controllable and user-friendly; (2) effective integration of 2D editing models with 3D mesh generation, achieving high-quality edited region meshes with structural harmony.

3.2. Geometry and Texture Fusion

We apply geometry and texture fusion on a hybrid SDF/Mesh representation [2]. For geometry, indirectly optimizing an SDF rather than directly optimizing mesh vertices yields more stable convergence and improved robustness to noise. For texture, we represent color through an implicit neural color field. Neural SDF fields provide several advantages over discrete meshes, including robustness to noise, improved convergence, differentiable rendering, analytical gradients, and inherent continuity. We leverage these properties to perform Poisson seamless fusion in the SDF domain, enabling simultaneous and coherent fusion of geometry and texture.

3.2.1. Poisson Geometry Blending

Based on the Poisson equation, we propose a Poisson Geometry Blending strategy to achieve seamless integration of the edited region into the original mesh while leveraging the edited reference mesh as structural guidance. We ensure

that the final mesh inherits the seamless transition and natural details of the reference mesh while retaining the fidelity of the original mesh and edited region mesh.

Fig. 3 gives an overview of the workflow. We first apply a mesh Boolean operation [8] between the original mesh and the edited region mesh to generate an initial merged mesh. We align the original mesh with the edited region mesh according to [58]. We then adopt a hybrid SDF/Mesh representation, which enables flexible refinement of mesh geometry by optimizing vertex positions, splitting triangles and collapsing edges. The refinement is guided by normal maps rendered from both the edited reference mesh and the edited region mesh, which are blended using a Poisson-based approach. This geometry blending strategy allows the edited region mesh to be naturally incorporated into the original mesh with harmonious boundary transitions.

Intersection Region Extraction. Given the original mesh M_o and the edited region mesh M_r, we first apply a mesh Boolean operation to obtain a merged mesh M_t. For insertion tasks, we use mesh Boolean union, and for deletion tasks, we use mesh Boolean difference. Geometric discontinuities occur at the transition boundary, so we extract and later refine it using a hybrid SDF/Mesh representation. The extraction process is as follows:

The Boolean operation produces a set of vertices V_{in} at the intersection between M_o and M_r. We align the edited reference mesh M_e with M_t. The corresponding intersection regions of M_t and M_e can be defined as:

$$M_t^{in} = \left\{ v \in M_t \mid \min_{u \in V_{in}} \|u - v\|_2 < \epsilon_0 \right\}, \tag{1}$$

$$M_e^{in} = \left\{ v \in M_e \mid \min_{u \in V_{in}} \|u - v\|_2 < \epsilon_0 \right\}, \tag{2}$$

where ϵ_0 controls the extent of the intersection. This focuses attention on the harsh transition. We define a smaller subset of M_t^{in} as its optimization region:

$$M_t^{opt} = \left\{ v \in M_t^{in} \mid \min_{u \in V_{in}} \|u - v\|_2 < \epsilon_1 \right\}, \quad \epsilon_1 < \epsilon_0. \tag{3}$$

Poisson Normal Blending Guidance. We train the mesh M_t^{in} indirectly by optimizing a neural SDF S_t it is bound to, changes in the SDF will be propagated to the mesh through vertex optimization by refining vertex positions, splitting triangles and collapsing edges. Vertices not in the optimization region M_t^{opt} are frozen during optimization. Indirectly optimizing an SDF instead of direct mesh optimization offers stable convergence and robustness against noise. Compared to voxel-based methods like DMTet [36], vertex optimization achieves precise and natural topology.

During optimization, we render multi-view supervision images: (1) a normal map of M_t^{in}, denoted n_t; (2) a binary mask of M_t^{opt}, denoted $mask^{opt}$; (3) a normal map rendered from the SDF S_t, denoted $\hat{n}_t$; (4) a normal map of M_e^{in}, denoted n_e. To enforce consistency, we apply the classical Poisson Image Editing (PIE) algorithm [38] $\Gamma(\cdot)$ to blend n_t and n_e under $mask^{opt}$:

$$n_p = \Gamma(n_t, n_e, mask^{opt}). \tag{4}$$

The blended normal map n_p preserves fine-grained details from n_e inside the mask while achieving a seamless transition to n_t along the mask boundary. Solving the Poisson equation on 2D images reduces computational complexity to $O(kn^2)$, offering significant efficiency over 3D volumetric methods ($O(n^3)$) while maintaining comparable fidelity.

We then minimize the discrepancy between the rendered normal $\hat{n}_t$ and the blended normal n_p:

$$\mathcal{L}_{\text{poisson}} = \sum_i \|\hat{n}_t^i - n_p^i\|_F^2, \tag{5}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and i indexes different camera viewpoints. Although the blended normal maps n_p^i are not strictly multi-view consistent, the implicit SDF effectively resolves inconsistencies and learns a coherent transition geometry. Following MagicClay [2], we further incorporate additional regularization terms, such as a smoothness loss $\mathcal{L}_{\text{smooth}}$ and an Eikonal loss $\mathcal{L}_{\text{eik}}$, to improve geometric fidelity and enforce implicit surface constraints. The loss for geometry blending is formulated as:

$$\mathcal{L}_{\text{geo}} = \mathcal{L}_{\text{poisson}} + \lambda_1 \mathcal{L}_{\text{smooth}} + \lambda_2 \mathcal{L}_{\text{eik}}, \tag{6}$$

where λ_1 and λ_2 are parameters.

3.2.2. Poisson Texture Harmonization

The newly synthesized regions of the merged mesh M_t lack texture information. A straightforward solution is to employ a texture generation model [33] conditioned on the edited reference image for color generation; however, the resulting textures often exhibit noticeable shifted colors compared to the original mesh and discontinuities along boundaries, as shown in Fig. 2. Therefore, we adopt the following Poisson-based strategies to improve overall visual harmony.

Gradient Propagation Let C_{new} denote the color field of M_t^{new}. Before optimization, we store a frozen copy C_{new}^{ori} for gradient reference. Since gradients encode details, we retain fine texture patterns by enforcing consistency between the gradients of the current and original color fields:

$$\mathcal{L}_{\text{grad}} = \text{MSE}\left(\sigma\left(\frac{\nabla C_{new}}{\gamma} \right), \sigma\left(\frac{\nabla C_{new}^{ori}}{\gamma} \right) \right), \tag{7}$$

where ∇ denotes numerical gradients, $\sigma(\cdot)$ is the sigmoid function, and γ is a gradient scaling constant. This term stabilizes and preserves high-frequency texture details.

Smooth Transition Refinement. To achieve seamless blending at intersection boundaries, we introduce a

distance-weighted soft boundary loss:

$$\mathcal{L}_{\text{boundary}} = \sum_{p_i^{new} \in M_t^{new}} w_i \| C_{new}(p_i^{new}) - C_{pr}(p_i^{pr}) \|_2^2, \tag{8}$$

where p_i^{pr} is the nearest point on M_t^{pr} to p_i^{new}, and

$$w_i = \left(1 - \frac{\delta}{\|p_i^{new} - p_i^{pr}\|_2}\right)^2 \tag{9}$$

attenuates the influence with distance. The parameter δ controls the effective boundary width. This boundary loss enforces smooth color transitions and prevents visible seams between fused regions.

Distribution-Aware Color Alignment. Theoretically, $\mathcal{L}_{\text{grad}}$ and $\mathcal{L}_{\text{boundary}}$ follow the formulation of Poisson Image Editing (PIE) [38]. However, the repetitive patterns inherent in textures interfere with gradient-domain color propagation, causing failure to normalize colors (Fig. 6e). Therefore, we propose an $\mathcal{L}_{\text{distribution}}$ loss that addresses this limitation by enforcing color distribution consistency between the generated and preserved regions.

Let M_t^{new} denote the newly synthesized geometry and M_t^{pr} the preserved geometry from the original mesh. The preserved mesh M_t^{pr} inherits textures directly from the original, while M_t^{new} is textured using a generative model [33]. The predicted colors form a probability density distribution in RGB space, which is obtained using kernel density estimation:

$$\rho(q) = \frac{1}{N} \sum_{i=1}^{N} \exp\left(-\frac{\|q - r_i\|^2}{2\sigma^2}\right), \tag{10}$$

where $\{r_i\}_{i=1}^N$ are sampled mesh colors and σ denotes the Gaussian kernel bandwidth.

We denote the color distributions of M_t^{new} and M_t^{pr} as ρ^{new} and ρ^{pr}, respectively. Distribution-aware alignment is achieved by minimizing their discrepancy:

$$\mathcal{L}_{\text{distribution}} = \frac{1}{N} \sum_{i=1}^{N} \|\rho^{new}(q_i) - \rho^{pr}(q_i)\|_2, \tag{11}$$

where $\{q_i\}_{i=1}^N$ are color samples from M_t^{new}.

The overall optimization objective for Poisson Texture Harmonization is:

$$\mathcal{L}_{\text{tex}} = \mathcal{L}_{\text{distribution}} + \theta_1 \mathcal{L}_{\text{grad}} + \theta_2 \mathcal{L}_{\text{boundary}}, \tag{12}$$

where θ_1 and θ_2 balance the gradient and boundary consistency terms. Unlike prior mesh editing pipelines that synthesize only albedo materials, our formulation directly extends to physically based rendering (PBR) materials.

Method	$\text{CLIP}_{\text{sim}} \uparrow$	$\text{CLIP}_{\text{dir}} \uparrow$	NIQE $\downarrow$	NIMA $\uparrow$
FocalDreamer	13.010	3.927	12.340	5.234
MagicClay	15.043	5.994	7.344	5.334
Instant3dit	14.108	4.326	7.390	5.288
VoxHammer	17.366	10.482	8.291	5.307
Ours (MeshyAI)	**20.801**	**18.479**	**4.710**	**5.928**

Table 1. Quantitative comparison with baselines. Our method consistently achieves the best performance across all metrics, demonstrating better semantic consistency and visual quality compared to existing methods.

4. Experiments

4.1. Experiment Setup

Implementation. We use FLUX Kontext [19] as the generative image-editing model and MeshyAI [33] as the image-to-mesh generator. It is worth noting that our framework is model-agnostic with respect to these components. In the future, more powerful models can be readily incorporated when conducting experiments. We adopt MagicClay [2] as the backbone for both the hybrid SDF–mesh representation and the implicit neural color field. On a single 4090 GPU, Poisson Geometry Blending requires 5 minutes for 1000 iterations, while Poisson Texture Harmonization requires 1 minute for 2000 iterations.

Dataset. We construct an evaluation dataset comprising 100 diverse and complex 3D models, selecting from Objaverse-XL [10], Google Scanned Objects [12] and the internet. For each model, we apply 3 distinct editing prompts. We evaluate these meshes through a series of complex editing tasks, including insertion, deletion, and localized region editing, to comprehensively assess our method's effectiveness in achieving global structural consistency and high-quality local details.

Baselines. We compare our method with recent state-of-the-art approaches for mesh editing, including SDS-based methods such as FocalDreamer [26] and MagicClay [2], the MVD-based method Instant3dit [3], and the Latent-based method Voxhammer [22]. All results were obtained using their official open-source implementations.

4.2. Quantitative Results

Metrics. Following prior work [26, 43], we adopt CLIP-based metrics for quantitative evaluation: **CLIP_{sim}** [40], which measures the semantic alignment between a rendered view of the edited mesh and the target text prompt; and **CLIP_{dir}**, which quantifies editing effectiveness by computing the directional CLIP similarity [13] between the original and edited meshes with respect to their text descriptions. In addition, we report **NIQE** [35] and **NIMA** [47], two no-reference image quality metrics that assess perceptual fidelity and better align with human visual judgment.

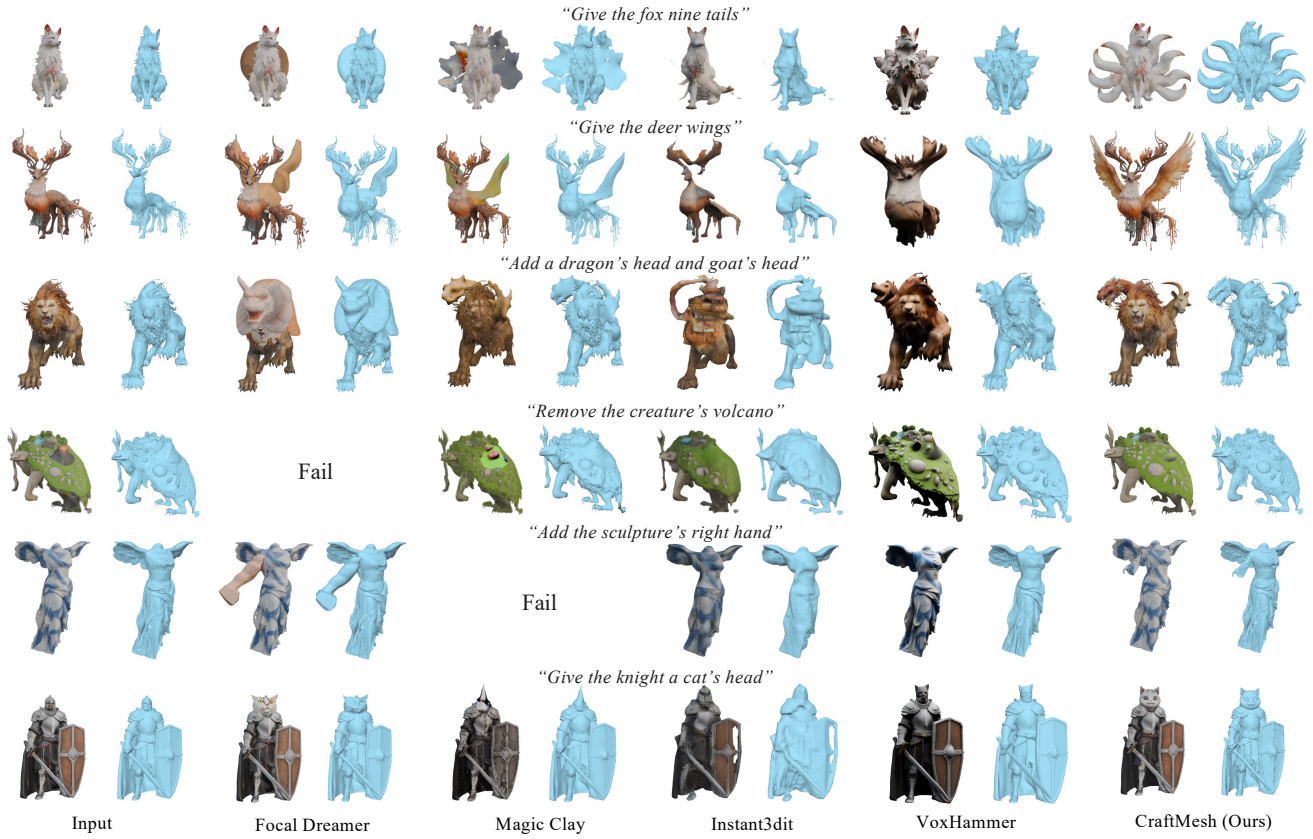


Figure 4. Qualitative comparisons show that our method produces harmonious geometry structure, intricate local details, and high-fidelity colors, while other methods give simple and inconsistent results.

Results. Tab. 1 summarizes the quantitative comparisons. Our method consistently achieves the best performance across all metrics, highlighting its superior ability to perform edits that are both semantically faithful to the target objectives and aesthetically satisfying to human vision, meeting the standards for high fidelity.

4.3. Qualitative Results

Fig. 4 presents a qualitative comparison with baseline methods. As illustrated, the baselines struggle with complex tasks, resulting in coarse geometry and a lack of detail. The generated colors are often overly simple, flat, and visually inconsistent. In contrast, our method produces harmonious global structure, rich local details, and high-fidelity colors.

For the fourth row, where the task is to delete the volcano, MagicClay [2] replaces the volcano with a distorted rock; Instant3dit [3] substitutes the volcano with a patch of grass but alters parts of the original mesh. In contrast, our method seamlessly removes the volcano, filling the space with rocks similar to those in adjacent regions, achieving both visual and geometric harmony.

4.4. Drag-based Mesh Editing

Beyond text-driven editing, our framework can be naturally extended to support more sophisticated mesh manipulation tasks. To demonstrate its flexibility and broader applicability, we apply our approach to enable **drag-based mesh editing** through **drag-based image editing**.

Unlike prompt-based editing, drag-based image editing allows users to specify edits by drawing arrows to encode the desired spatial deformations, offering precise and intuitive control. In our implementation, we use LightningDrag [45] as the underlying drag-based image editor. The workflow of drag-based mesh editing is as follows. First, drag-based image editing is performed to generate the desired image modifications. Second, mesh deletion is applied to corresponding regions of the original mesh. Finally, mesh insertion is then performed using the edited region meshes generated from drag-based image editing.

Fig. 5a shows the original meshes with arrow annotations illustrating the intent to open the angel's wings and lift the cat's paws, while Fig. 5b presents the results of drag-based editing. The successful results highlight the adaptability of

(a) Original (b) Edited

Figure 5. Our method effectively adapts to drag-based mesh editing, enabling precise and intuitive geometry manipulation through user-specified drag controls. (a) shows the original mesh, with arrows drawn to signify drag; (b) shows the editing results.

our framework, demonstrating its strong potential for generalization to other advanced mesh editing scenarios.

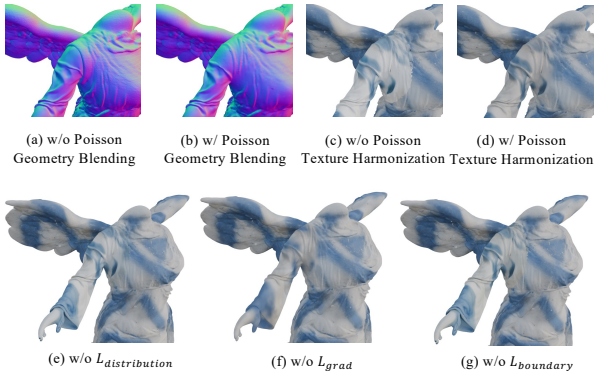
4.5. Ablation Studies

We conduct ablation studies on key components of our method: Poisson Geometry Blending, Poisson Texture Harmonization and the Mesh Generation Backend. Qualitative comparisons are presented in Fig. 6, and quantitative results are shown in Tab. 2. To ensure fairness, we establish the baseline by applying mesh Boolean [8] between the original mesh and the edited region mesh.

Effects of Poisson Geometry Blending. This module effectively resolves abrupt geometric discontinuities (Fig. 6), replacing them with realistic details such as cloth wrinkles, achieving a seamless transition. Compared with the baseline, it yields substantially improved geometric fidelity and quantitative performance.

Effects of Poisson Texture Harmonization. This module improves texture harmony, for example, harmonizing the bright tone of the hand with the body's darker gray while achieving texture continuity along boundaries. Further ablation experiments on each loss term demonstrate their significance: removing the distribution loss (Fig. 6e) produces unharmonious hues, omitting the gradient loss (Fig. 6f) results in blurred colors, and excluding the boundary loss (Fig. 6g) introduces visible texture discontinuities.

Effects of Mesh Generation Backend. To demonstrate that our improvements stem from the proposed fusion framework rather than a specific generative backbone, we evaluate three distinct backends: Trellis [54], Hunyuan3D 2.5 [20] and MeshyAI [33]. As shown in Table 2, different configurations achieve consistently comparable state-of-the-art performance, thereby confirming that the observed improvements arise from our proposed fusion pipeline.



(a) w/o Poisson Geometry Blending (b) w/ Poisson Geometry Blending (c) w/o Poisson Texture Harmonization (d) w/ Poisson Texture Harmonization

(e) w/o $L_{distribution}$ (f) w/o L_{grad} (g) w/o $L_{boundary}$

Figure 6. Ablation studies. (a,b) Poisson Geometry Blending; (c,d) Poisson Texture Harmonization; (e) the distribution loss; (f) the gradient loss; (g) the boundary loss.

Method	CLIP_{sim} ↑	CLIP_{dir} ↑	NIQE ↓	NIMA↑
Baseline	17.723	10.348	5.802	5.073
w/ Geometry Blending	20.502	11.979	5.774	5.290
w/ Texture Harmonization	19.399	10.724	5.290	5.184
Ours (Hunyuan3D 2.5)	19.903	18.622	6.108	5.749
Ours (Trellis)	19.166	**18.911**	6.246	**5.989**
Ours (MeshyAI)	**20.801**	18.479	**4.710**	5.928

Table 2. Quantitative results of the ablation study. Both Geometry Blending and Texture Harmonization individually improve over the baseline, while combining them yields the best performance.

5. Conclusion

We present CraftMesh, a framework for high-fidelity 3D mesh manipulation. Our approach combines 2D image editing with 3D generative models through a Poisson Seamless Fusion strategy. Built upon the hybrid SDF/Mesh representation, our method ensures both geometric and texture consistency, supporting complex and detailed edits that are seamlessly integrated into the original mesh. Experimental results demonstrate that CraftMesh outperforms existing baselines, delivering more coherent geometric structure, finer surface details, and higher-fidelity texture. The framework is also versatile, supporting advanced operations such as drag-based mesh editing. Future work may explore more sophisticated mesh editing techniques.

Limitation. CraftMesh relies on off-the-shelf 2D and 3D generative models and therefore naturally inherits their limitations. For more complex meshes (e.g., those containing open, multiple layers, or noisy surfaces), this dependency may limit the achievable fidelity. Nevertheless, more powerful generative models can be readily substituted as they become available.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (U25A20444, 62025207) and the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (JYB2025XDXM113).

References

[1] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. In *ACM SIGGRAPH*, pages 294–302, 2004. 3

[2] Amir Barda, Vladimir Kim, Noam Aigerman, Amit Haim Bermano, and Thibault Groueix. Magicclay: Sculpting meshes with generative neural fields. In *ACM SIGGRAPH Asia*, pages 1–10, 2024. 2, 3, 4, 5, 6, 7

[3] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16273–16282, 2025. 2, 3, 4, 6, 7

[4] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 28(3):24, 2009. 3

[5] 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*, 2024. 2, 3

[6] Zhaoxi Chen, Jiaxiang Tang, Yuhao Dong, Ziang Cao, Fangzhou Hong, Yushi Lan, Tengfei Wang, Haozhe Xie, Tong Wu, Shunsuke Saito, et al. 3dtopia-xl: Scaling high-quality 3d asset generation via primitive diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26576–26586, 2025. 2

[7] Xinhua Cheng, Tianyu Yang, Jianan Wang, Yu Li, Lei Zhang, Jian Zhang, and Li Yuan. Progressive3d: Progressively local editing for text-to-3d content creation with complex semantic prompts. *arXiv preprint arXiv:2310.11784*, 2023. 3

[8] Gianmarco Cherchi, Fabio Pellacini, Marco Attene, and Marco Livesu. Interactive and robust mesh booleans. *arXiv preprint arXiv:2205.14151*, 2022. 5, 8

[9] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 4

[10] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. In *Advances in Neural Information Processing Systems*, pages 35799–35813, 2023. 2, 6

[11] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 2

[12] 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. *arXiv preprint arXiv:2204.11918*, 2022. 6

[13] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *arXiv preprint arXiv:2108.00946*, 2021. 6

[14] Will Gao, Dilin Wang, Yuchen Fan, Aljaz Bozic, Tuur Stuyck, Zhengqin Li, Zhao Dong, Rakesh Ranjan, and Nikolaos Sarafianos. 3d mesh editing using masked lrms. *arXiv preprint arXiv:2412.08641*, 2024. 3

[15] Xinyu Gao, Ziyi Yang, Bingchen Gong, Xiaoguang Han, Sipeng Yang, and Xiaogang Jin. Towards realistic example-based modeling via 3d gaussian stitching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26597–26607, 2025. 2, 3

[16] Bingchen Gong, Yuehao Wang, Xiaoguang Han, and Qi Dou. Seamlessnerf: Stitching part nerfs with gradient propagation. In *ACM SIGGRAPH*, pages 1–10, 2023. 2, 3

[17] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19740–19750, 2023. 1, 3

[18] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. In *ACM SIGGRAPH*, pages 795–802, 2005. 3

[19] Black Forest Labs. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025. 2, 4, 6

[20] Zeqiang Lai, Yunfei Zhao, Haolin Liu, Zibo Zhao, Qingxiang Lin, Huiwen Shi, Xianghui Yang, Mingxin Yang, Shuhui Yang, Yifei Feng, et al. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details. *arXiv preprint arXiv:2506.16504*, 2025. 1, 3, 4, 8

[21] 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. *arXiv preprint arXiv:2311.06214*, 2023. 2

[22] 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. *arXiv preprint arXiv:2508.19247*, 2025. 6

[23] Peng Li, Suizhi Ma, Jialiang Chen, Yuan Liu, Congyi Zhang, Wei Xue, Wenhan Luo, Alla Sheffer, Wenping Wang, and Yike Guo. Cmd: Controllable multiview diffusion for 3d

editing and progressive generation. In *ACM SIGGRAPH*, pages 1–10, 2025. 2, 3

[24] Weiyu Li, Jiarui Liu, Hongyu Yan, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman3d: High-fidelity mesh generation with 3d native generation and interactive geometry refiner. *arXiv preprint arXiv:2405.14979*, 2024. 1, 4

[25] Xingyu Li, Fei Tao, Wei Ye, Aydin Nassehi, and John W Sutherland. Generative manufacturing systems using diffusion models and chatgpt. *arXiv preprint arXiv:2405.00958*, 2024. 1

[26] Yuhan Li, Yishun Dou, Yue Shi, Yu Lei, Xuanhong Chen, Yi Zhang, Peng Zhou, and Bingbing Ni. Focaldreamer: Text-driven 3d editing via focal-fusion assembly. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3279–3287, 2024. 2, 3, 4, 6

[27] Jessica E Liang. Diffusion models for robotics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 29587–29589, 2025. 1

[28] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6517–6526, 2024. 2

[29] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 2

[30] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10072–10083, 2024. 2

[31] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 2

[32] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9970–9980, 2024. 2

[33] Meshy Inc. Meshy.ai: Ai-powered 3d generation platform, 2025. https://www.meshy.ai/. 4, 5, 6, 8

[34] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1

[35] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a "Completely Blind" Image Quality Analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2013. 6

[36] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8280–8290, 2022. 5

[37] Jonas Oppenlaender. The creativity of text-to-image generation. In *Proceedings of the 25th International Academic Mindtrek Conference*, pages 192–202, 2022. 2

[38] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM SIGGRAPH*, page 313–318, 2003. 3, 5, 6

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

[40] 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. *arXiv preprint arXiv:2103.00020*, 2021. 6

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2

[42] Benet Oriol Sabat, Alessandro Achille, Matthew Trager, and Stefano Soatto. Nerf-insert: 3d local editing with multimodal control signals. *arXiv preprint arXiv:2404.19204*, 2024. 3

[43] Etai Sella, Gal Fiebelman, Peter Hedman, and Hadar Averbuch-Elor. Vox-e: Text-guided voxel editing of 3d objects. *arXiv preprint arXiv:2303.12048*, 2023. 6

[44] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 2

[45] Yujun Shi, Jun Hao Liew, Hanshu Yan, Vincent Y. F. Tan, and Jiashi Feng. Lightningdrag: Lightning fast and accurate drag-based image editing emerging from videos. *arXiv preprint arXiv:2405.13722*, 2024. 7

[46] Yawar Siddiqui, Tom Monnier, Filippos Kokkinos, Mahendra Kariya, Yanir Kleiman, Emilien Garreau, Oran Gafni, Natalia Neverova, Andrea Vedaldi, Roman Shapovalov, et al. Meta 3d assetgen: Text-to-mesh generation with high-quality geometry, texture, and pbr materials. *Advances in Neural Information Processing Systems*, 37:9532–9564, 2024. 1

[47] Hossein Talebi and Peyman Milanfar. Nima: Neural image assessment. *IEEE Transactions on Image Processing*, 27(8): 3998–4011, 2018. 6

[48] Khanh Hoa Thi Vo. Augmented reality, virtual reality, and mixed reality: A pragmatic view from diffusion of innovation. *International Journal of Architectural Computing*, 23 (1):27–45, 2025. 1

[49] Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In *European Conference on Computer Vision*, pages 439–457, 2024. 2

[50] Junjie Wang, Jiemin Fang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. Gaussianeditor: Editing 3d gaussians delicately with text instructions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20902–20911, 2024. 1, 3

[51] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36:8406–8441, 2023. 2

[52] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report. *https://arxiv.org/abs/2508.02324*, 2025. 4

[53] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 803–814, 2023. 2

[54] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21469–21480, 2025. 2, 4, 8

[55] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. 1

[56] Yizhou Yu, Kun Zhou, Dong Xu, Xiaohan Shi, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Mesh editing with poisson-based gradient field manipulation. In *ACM SIGGRAPH*, pages 644–651, 2004. 2, 3

[57] 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*, 43(4):1–20, 2024. 2

[58] Xiaoming Zhu, Xu Huang, Qinghongbing Xie, Zhi Deng, Junsheng Yu, Yirui Guan, Zhongyuan Liu, Lin Zhu, Qijun Zhao, Ligang Liu, and Long Zeng. Imaginarium: Vision-guided high-quality 3d scene layout generation. *arXiv preprint arXiv:2510.15564*, 2025. 5

[59] 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*, 43(4):1–12, 2024. 1, 3