Training-Free Geometric Image Editing on Diffusion Models

Hanshen Zhu[1*] Zhen Zhu[2*] Kaile Zhang[1] Yiming Gong[2] Yuliang Liu[1] Xiang Bai[1†]
[1]Huazhong University of Science and Technology
[2]University of Illinois at Urbana-Champaign
[1]{zhs_china, klzhang, ylliu, xbai}@hust.edu.cn
[2]{zhenzhu4, yimingg8}@illinois.edu

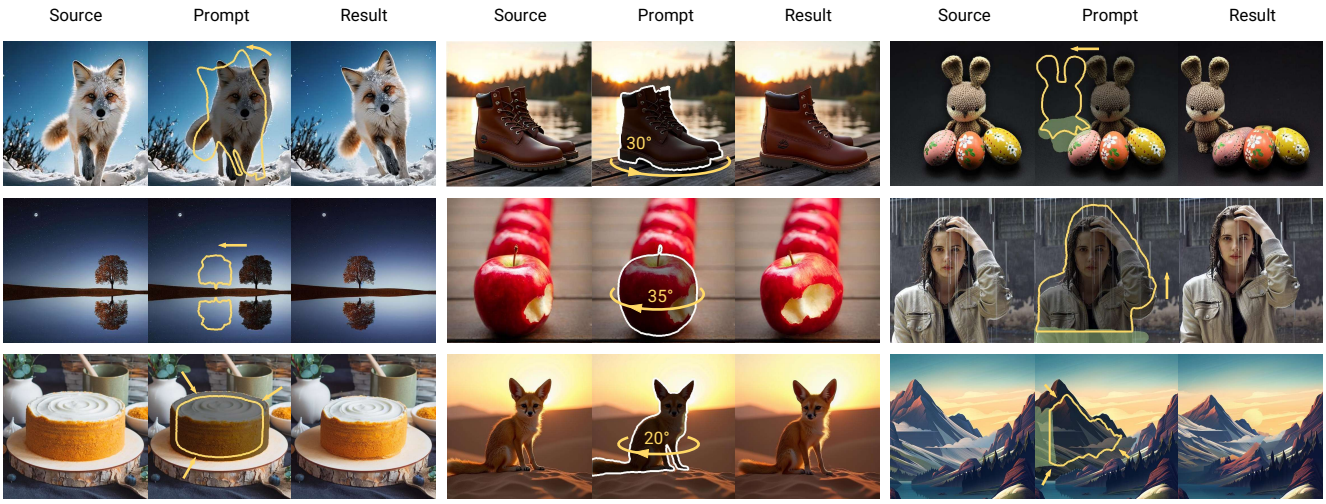


Figure 1. Given an image and an editing instruction, our method precisely performs geometric edits while maintaining high fidelity and avoiding artifacts. Besides, our training-free framework achieves impressive structural completion and background generation.

Abstract

We tackle the task of geometric image editing, where an object within an image is repositioned, reoriented, or reshaped while preserving overall scene coherence. Previous diffusion-based editing methods often attempt to handle all relevant subtasks in a single step, proving difficult when transformations become large or structurally complex. We address this by proposing a decoupled pipeline that separates object transformation, source region inpainting, and target region refinement. Both inpainting and refinement are implemented using a ***training-free*** *diffusion approach,* ***FreeFine****. In experiments on our new* ***GeoBench*** *benchmark, which contains both 2D and 3D editing scenarios, FreeFine outperforms state-of-the-art alternatives in image fidelity, and edit precision, especially under demanding transformations. Code and benchmark are available at:* https://github.com/CIawevy/FreeFine

1. Introduction

Image generation models have made remarkable progress in producing photorealistic and detail-rich results [2, 43, 47, 48]. With these advancements, the community has shown growing interest in controllable image editing to enable users to manipulate existing images with both high fidelity and accuracy. In this paper, we address the task of repositioning, reorienting, or reshaping an object within an image (*e.g.*, moving an object to a new location, rotating it in 3D, or changing its proportions) while preserving overall scene coherence, a task we refer to as **geometric image editing**.

This problem requires solving multiple interdependent subtasks: (1) coarsely transforming the object to its desired location, (2) inpainting the source region to avoid artifacts, and (3) refining the relocated object to blend seamlessly with the background. Recent methods that support drag-based edits [8, 10, 33, 39, 50] typically address these goals with a single, unified objective, yielding compelling results for smaller or moderate transformations. However, they of-

*Equal contribution. †Corresponding author.

ten struggle with large or geometrically complex transformations, possibly because balancing multiple subtasks in one optimization framework creates competing demands. For instance, strictly preserving the background may conflict with generating newly exposed object surfaces. Although it is challenging to pinpoint exactly how these objectives interact, the risk of artifacts in more extensive edits motivates us to take a **decoupled** strategy. Concretely, we separate geometric editing into three sequential steps, individually handling object transformation, source region inpainting, and target region refinement. This decomposition avoids the pitfalls of fusing significant structural changes and fine-grained touch-ups in a single loop, while allowing us to incorporate specialized off-the-shelf components, such as advanced depth estimators [64] or video models [61] for 3D transformations and dedicated inpainting models [28, 57] for large-scale removals.

Within this pipeline, both inpainting and refinement can be cast as selectively altering regions based on spatial masks and generative priors. Instead of leveraging task-specific models [22, 54, 57] that necessitate dedicated training, we adopt a *training-free* diffusion-based strategy capable of handling both tasks with minimal tweaks, which combines three complementary modules: *Temporally Contextual Attention* to balance self-attention with mask-guided attention over the course of diffusion steps, *Local Perturbation* for selective noise injection that encourages substantial changes in user-defined areas, and *Content-specified Generation* for text-driven local refinements. These modules collectively preserve global fidelity while generating plausible details in newly exposed or structurally incomplete regions.

To systematically evaluate geometric editing, we introduce **GeoBench**, a benchmark designed to test 2D and 3D transformation editing scenarios in varied degrees of difficulty. Each scenario includes a source image, one or more geometric editing instructions, and optional structural completion masks. We adopt popular generative metrics such as FID [16], as well as measures for subject/background consistency [19] and edit precision [49].

Our **contributions** can be summarized as follows:

- We present a decoupled geometric image editing pipeline that splits the editing process into object transformation, source region inpainting, and target region refinement, supporting both 2D and 3D transformations.
- We propose FreeFine, a *training-free*, diffusion-based solution powered by Temporal Contextual Attention, Local Perturbation, and Content-specified Generation. This design provides fine-grained control over the editing regions while maintaining global coherence.
- We introduce GeoBench, a benchmark specialized for evaluation on geometric image editing with diverse instructions and metrics. Our method significantly outperforms existing approaches in both large and small transformations.

2. Related Work

Diffusion models. Diffusion models [18, 53] synthesize images by iteratively denoising noisy inputs, with improvements such as non-Markovian sampling [53] and latent diffusion [47] greatly accelerating generation. Combined with large-scale language modeling, they power state-of-the-art text-to-image systems like DALL-E [2, 44, 45] and Imagen [48]. Recent extensions include rectified flow models [32] for more efficient sampling. We refer readers to [65] for broader coverage of research on diffusion models.

Geometric editing with generative models. Early methods for view manipulation and alignment [7, 20] lacked the ability to perform direct image editing. Advances in novel view synthesis and 3D representations [25, 31, 38, 61] enable more expressive geometric changes by reconstructing scenes in 3D space [66]. However, these methods typically require multi-view inputs or per-scene optimization, making them less practical for single-image, real-time editing. More recent work on single-image 3D reconstruction attempts to infer 3D shapes from sparse or single-view data [31, 61], and can be seamlessly integrated into our framework: for larger 3D transformations, the object is temporarily "lifted" to 3D, manipulated, and reprojected back to 2D. Other approaches, such as pose transfer [37, 52, 68, 69] or virtual try-on [5, 12, 62], focus on human-centric transformations and cannot generalize beyond that domain without significant modification.

Given source and target locations or constraints, a decent volume of research efforts focus on manipulating latents to minimize feature discrepancies. They can be categorized by their guidance signals: (i) *Point-based methods* [6, 27, 30, 41, 50, 51] accept user-defined points or anchors to indicate how parts of the object should move, (ii) *Region-based methods* [8, 42, 49] operate on segmented areas to apply object-level edits, and (iii) *Combined or more diverse signals* [33, 39] fuse point and region information. Further variations exploit flow fields [10], scene layouts [21, 35, 46], or higher-level editable elements [40]. Our method can optionally work with single-image 3D lifting [61] to handle substantial viewpoint changes. Meanwhile, our method remains compatible with diverse 2D editing operations without requiring multi-view data or specialized human-centric models.

Training-Free Inpainting and Restoration. While some inpainting methods often rely on a specialized training procedure tailored to repair designated areas [22, 54], recent works [9, 36, 56, 67] have exploited the strong generative priors of large diffusion models for image inpainting and restoration without further training. For instance, RePaint [34] iteratively refines an inpainted region through a

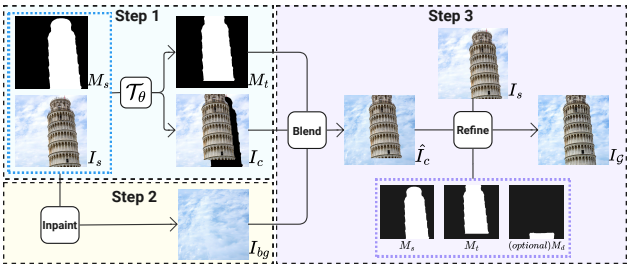


Figure 2. Overview of our geometric image editing pipeline.

corruption-aware sampling loop, while DDRM [24] adapts diffusion priors to tasks like denoising or super-resolution by enforcing consistency with the degraded observations. DreamClean [63] further extends unsupervised test-time sampling to remove unknown corruption. Most of these methods focus on local artifact correction or low-level enhancements. By contrast, our approach supports various geometric transformations and can synthesize new content guided by user prompts and masks.

3. Method

3.1. Geometric Editing Pipeline

Our goal is to perform geometric editing on any object within the image, manipulating its shape, orientation, or position while preserving the overall coherence and realism of the scene. Our pipeline (Fig. 2) comprises **three** steps:

Step 1: Object Transformation. The pipeline takes a source image I_s containing an object to be edited, along with a binary mask M_s indicating the source region for the object. To reduce the burden of drawing precise masks, we employ an interactive segmentation model [26] that generates M_s with just a few user clicks. Next, we convert user's editing instruction (*e.g.*, "rotate the object along the z-axis by 30°") to a transformation function $\mathcal{T}_\theta$ with parameters θ. The transformation function receives I_s and M_s as inputs, and generates transformed M_t that indicates the target region for the object, and a *coarse* image I_c:

$$I_c, M_t = \mathcal{T}_\theta(I_s, M_s).$$

For purely 2D edits, $\mathcal{T}_\theta$ simply represents an affine transformation. For more advanced 3D edits, we first estimate scene depth using a depth estimator [64], then apply geometric transformations in 3D space and re-project the transformed object back into the image. For more demanding 3D changes, we employ single-image 3D lifting methods such as SV3D [13, 61] for a more complete 3D representation.

Step 2: Source Region Inpainting. With the object relocated, the original source region often requires cleanup (see I_c in Fig. 2). This step aims to generate a clean background image I_{bg} by inpainting the source region of I_s, ensuring natural blending with surrounding pixels:

$$I_{bg} = \texttt{Inpaint}(I_s, M_s).$$

Step 3: Target Region Refinement. Given the target mask M_t and the coarse image I_c from Step 1, and I_{bg} with clean background in the source object location obtained from Step 2, we can easily `Blend` them together to create a composite image $\hat{I}_c$:

$$\hat{I}_c = M_t \cdot I_c + (1 - M_t) \cdot I_{bg}.$$

$\hat{I}_c$ can be imperfect: as I_c and I_{bg} are separately built, the blended result may fall short in unnatural boundaries around the target object regions. More critically, occlusion or incompleteness of the original object can severely compromise the realism of the edited result. As an example in Fig. 2, the tower in $\hat{I}_c$ is up in the air without realistic structure in the base part. These limitations necessitate an additional refinement step on $\hat{I}_c$. This step requires many inputs obtained from previous steps:

$$I_{\mathcal{G}} = \texttt{Refine}(\hat{I}_c, I_s, M_s, M_t, [M_d]),$$

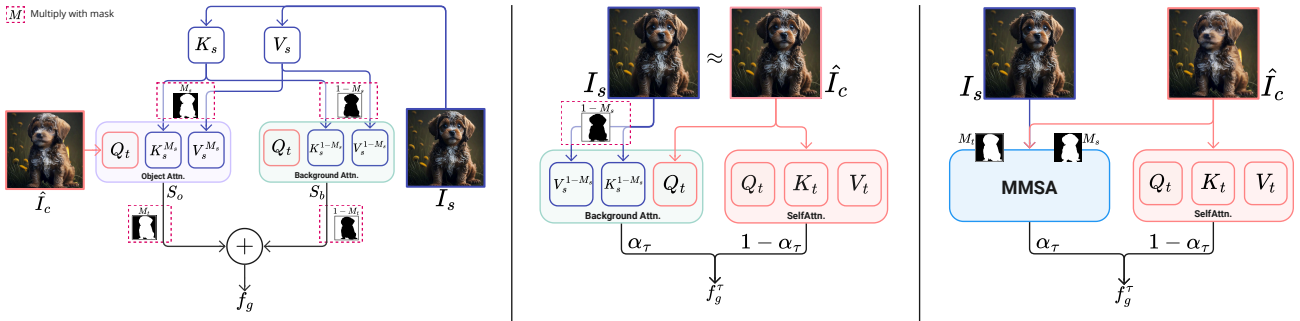
where M_d is an *optionally* user-provided completion mask for controlled content generation based on $\hat{I}_c$.

Although `Inpaint` and `Refine` have different objectives, both boil down to refining selected pixels using existing context from the available contexts. To that end, we propose a training-**free** image re**fine**ment approach, **FreeFine**, to perform these steps in a unified manner. In what follows, we detail the FreeFine framework and how it integrates into Steps 2 and 3.

3.2. Training-Free Image Refinement

A widely adopted approach for diffusion-based editing is to perform DDIM inversion on the input image and manipulate the latent at each denoising step. Here, we invert the source image I_s *once* and reuse its latent for both Step 2 (source region inpainting) and Step 3 (target region refinement). Once the composite image $\hat{I}_c$ is generated, we additionally invert it to initialize the latent representation for Step 3. Past studies [1, 4, 14, 15, 29, 60] demonstrate that manipulating self-attention features can be highly effective, given the rich semantic information learned by large-scale pretrained models. However, it is crucial to manage *where* and *how* such modifications occur. To this end, we introduce three complementary modules:

- **Temporal Contextual Attention (TCA)** in Sec. 3.2.1, which embodies the mechanism to smoothly transition from mask-guided mutual attention to full self-attention and preserve global structure during large edits.
- **Local Perturbation (LP)** in Sec. 3.2.2, a method that selectively injects noise (via DDPM updates) into user-defined regions to permit substantial content changes without disturbing the rest of the image.



(a) Mask-guided Mutual Self Attention (**MMSA**) (b) Temporal Contextual Attention (**Step 2**) (c) Temporal Contextual Attention (**Step 3**)

Figure 3. Comparison of Context Aggregation Methods. This figure illustrates different approaches for context alignment in image editing tasks: (a) MMSA [4] replaces Key-Value (KV) pairs and enforces explicit feature interaction between regions. (b) TCA (Step2) for source region inpainting, and (c) TCA (Step3) for target region refinement, which smoothly transition from MMSA to full self-attention.

- **Content-specified Generation (CG)** in Sec. 3.2.3, which utilizes classifier-free guidance and cross-attention based on user-provided prompts to steer newly generated content, ensuring desired details appear only where intended.

Together, these modules enable FreeFine to perform inpainting and refinement in a training-free manner while preserving the realism and coherence of the edited image.

3.2.1. Temporal Contextual Attention

Diffusion-based editing aims to adjust latents in editable regions while keeping other parts intact. As self-attention layers capture feature dependencies, they enable diffusion models to have "repairing abilities": corrupting the regions to be edited and then performing the denoising process can harmonize corrupted regions and generate plausible results. But it is difficult to control both *where* such "repairment" occurs and its intensity. Mask-guided mutual self-attention (MMSA) [4] (illustrated in Fig. 3 (a)) avoids this by restricting each query in the self-attention layers to attend only to the corresponding masked features:

$$\begin{aligned} S_o &= \text{SelfAttn}(Q_t, K_s, V_s;\, M_s), \\ S_b &= \text{SelfAttn}(Q_t, K_s, V_s;\, 1 - M_s), \\ f_g &= S_o \cdot M_t \;+\; S_b \cdot (1 - M_t), \end{aligned} \tag{1}$$

where S_o and S_b represent the self-attention outputs for objects and backgrounds, respectively, and f_g is the final feature output of the whole module. While effective, this approach is limited in generating structural content if large edits are desirable (shown in Sec. 4.3).

Our key observation (aligned with the literature [4, 11, 14]) is that major content changes tend to occur at the beginning of the diffusion process, whereas later steps naturally refine local details. Hence, we seek a balanced approach that starts with MMSA to preserve global structure and then gradually incorporates self-attention as denoising progresses, enabling local adjustments without compromising the larger transformation.

To implement this idea, we propose a *Temporal Contextual Attention (TCA)* mechanism shown in Fig. 3 (c) that dynamically blends MMSA with self-attention from the target over the diffusion steps. Formally, let $S_t = \text{SelfAttn}(Q_t, K_t, V_t)$ be the self-attention output capturing global context from the current image latent. We then define the final attention output f_g^τ at step τ as

$$f_g^\tau \;=\; (1 - \alpha_\tau)\, S_t \;+\; \alpha_\tau \Big[S_o \cdot M_t \;+\; S_b \cdot (1 - M_t) \Big],$$

where

$$\alpha_\tau = \frac{\tau_1 - \tau}{\tau_1 - \tau_0}$$

is a *temporally changing* blend weight that smoothly transitions from relying on MMSA at early steps (for large content edits) to incorporating self-attention at later steps (for fine details and completion). Concretely, α_τ linearly decreases from 1 when $\tau = \tau_0$ to 0 when $\tau = \tau_1$, where τ_0 is the initial denoising step, and τ_1 is the final denoising step. A straightforward alternative is to define a single threshold: for steps it, use MMSA from the source; once past it, switch entirely to self-attention. This approach is highly dependent on threshold choice and incurs varied performance for different thresholds (see Appendix for details). Our approach is free from this concern and thus more robust.

TCA can be applied to both Step 2 (Fig. 3(b)) and Step 3 (Fig. 3(c)), depending on the masks and the values of τ_0 and τ_1. For Step 2, we maintain background self-attention and combine it with full self-attention through α_τ, ensuring that source regions query only the background of I_s while leveraging self-dependency for self-remedy. For Step 3, when structure completion is required, we extend the target mask M_t by incorporating the user-defined mask M_d, forming the full target mask $M_t^* = M_t \cup M_d$. This allows the completion region to focus on the foreground region of I_s while smoothly combining self-attention for coherent content completion.

3.2.2. Local Perturbation

While DDIM [53] provides deterministic denoising updates, in many editing scenarios there is a need to *unfreeze* certain regions to allow more dramatic changes, such as inpainting large occlusions or fixing problematic artifacts. To accomplish this, we introduce *Local Perturbation* (LP), which injects controlled stochasticity *only* within a specified region $\mathcal{M}$. Formally, we selectively apply a DDPM-like update in $\mathcal{M}$ while retaining DDIM updates elsewhere:

$$x_{t-1} = \begin{cases} \text{DDPM}(x_t), & \text{if } x \in \mathcal{M}, \\ \text{DDIM}(x_t), & \text{otherwise.} \end{cases}$$

By applying stochasticity only in $\mathcal{M}$, LP provides extra flexibility to rearrange or complete local structures within the region without disrupting content outside it. For Step 2, $\mathcal{M}$ typically corresponds to the source mask M_s, while in Step 3, $\mathcal{M}$ is defined by the user-defined structure completion mask M_d or other regions requiring substantial modification, such as areas containing inpainting artifacts or the boundaries between objects and the background. More details can be found in the Appendix.

3.2.3. Content-specified Generation

Though LP encourages greater variety in how masked regions evolve, it still relies on the model's generative prior to fill in missing details, which can be arbitrary if the desired content is not specified. To address this, we introduce a textual conditional input $\mathcal{C}$ (*e.g.*, "add a missing foot", "make an empty scene") and define two regions: $\mathcal{M}_1$ for localized cross-attention and $\mathcal{M}_2$ to guide classifier-free guidance [17].

Within $\mathcal{M}_1$, we replace the source key–value pairs with those projected from the textual embeddings $(K_\mathcal{C}, V_\mathcal{C})$:

$$\begin{aligned} \tilde{f}_t \; = \text{CrossAttn}(Q_t, K_\mathcal{C}, V_\mathcal{C}) \cdot \mathcal{M}_1 + \\ \text{CrossAttn}(Q_t, K_\varnothing, V_\varnothing) \cdot (1 - \mathcal{M}_1), \end{aligned} \tag{2}$$

where Q_t is the target query and $(K_\varnothing, V_\varnothing)$ are text embeddings from null texts. This ensures cross-attention is locally focused on $\mathcal{M}_1$, enabling precise, user-specified generation within the region while preserving external content.

Furthermore, we apply classifier-free guidance *exclusively* $\mathcal{M}_2$:

$$\hat{\epsilon}_\theta(x_t, \mathcal{C}) = \epsilon_\theta(x_t, \varnothing) + w \left[\epsilon_\theta(x_t, \mathcal{C}) - \epsilon_\theta(x_t, \varnothing)\right] \cdot \mathcal{M}_2, \tag{3}$$

where w is the guidance scale. This confines semantic directives in $\mathcal{C}$ to $\mathcal{M}_2$, preventing unintended changes to well-established parts of the image. For Step 2, $\mathcal{M}_1 = \mathcal{M}_2 = M_s$, where M_s is the source region mask. For Step 3, $\mathcal{M}_1 = M_t^*$, while $\mathcal{M}_2 = M_d$. By combining LP's controlled stochasticity with CG's text-driven conditioning, users can refine or synthesize content exactly where needed while preserving overall image coherence.

4. Experiments

4.1. Experimental Settings

Additional implementation details of our method, compared approaches, dataset, and metric calculations are provided in the Appendix.

Implementation Details. We adopt Stable Diffusion v1-5 [47] as our base model, with an image resolution of 512×512, consistent with previous methods. The number of the denoising steps τ_1 is set to 50. TCA is applied at all timesteps, starting from the tenth U-Net layer, following [4]. For Step 2, we set τ_0 to 1, minimizing the influence of residual object features on background reconstruction. For Step 3, τ_0 is set to 13 to balance structural completion and detail preservation. For general refinement without structural completion, τ_0 is set to 25 for fine-grained adjustments. CG is applied in both Step 2 and Step 3 with a default guidance scale $w = 7.5$.

Datasets. To evaluate on geometric editing, we construct a comprehensive benchmark, **GeoBench**, by combining source images from PIE-Bench [23] and Subjects200K [58], which contain a mix of real and synthetic images with apparent objects suitable for this task. For each source image, we provide multiple geometric editing instructions, including object-centric operations such as move, rotate, and resize, each with varying directions and three intensity levels (*i.e.*, easy, medium, hard). The GeoBench dataset comprises 811 source images and 5,988 editing instructions in total, offering a robust foundation for evaluating geometric editing methods. Our benchmark includes diverse editing scenarios and a challenging subset requiring *structural completion*. Additionally, we provide detailed captions and object category labels for all images, as well as annotated regions for structural completion tasks.

Metrics. We employed seven metrics to quantitatively evaluate the generated results from three different perspectives. (1) **Image Quality**. FID [16] is computed to comprehensively evaluate the quality of the generated images. We randomly sample 2k images from PIE-Bench [23] and Subjects200K [58] as data from the target. In addition, we separately use Kernel distances [3] (KD) and DINOv2 [55] feature distance (DINOv2) to improve the FID and obtain more comprehensive results. (2) **Consistency**. Inspired by VBench [19], we adopt Subject Consistency (SUBC) and Background Consistency (BC) to assess the fidelity of the generated image to the input source images. After separating the subject from the background using M_s and M_t, we calculate the cosine similarity between their foregrounds and between their backgrounds in the feature space. (3) **Editing Effectiveness**. In image editing tasks, it is crucial to evaluate whether the generated images adhere to the input editing instructions. We employ the same Warp Error (WE)

Table 1. Comparison on 2D Edits, 3D Edits, and Structure Completion (SC) tasks. Best results are in bold, second best in underlined.

Methods	External Model	Editing Type	FID	DINOv2	KD	SUBC	BC	WE	MD
Self-Guidance [8]	SAM [26]	2D	49.15	647.56	0.438	0.575	0.759	0.268	116.89
RegionDrag [33]	SAM [26]		40.21	504.50	0.241	0.796	0.970	0.120	32.50
DragonDiffusion [39]	SAM [26]		37.09	507.67	0.144	0.840	0.968	0.158	32.36
MotionGuidance [10]	SAM [26], RAFT [59]		106.39	1189.23	3.871	0.521	0.736	0.186	90.03
DragDiffusion [50]	SAM [26]		36.58	455.68	**0.142**	0.758	0.966	0.199	41.31
Diffusion Handles [42]	SAM [26], LaMa [57], DepthAnything [64]		44.81	549.69	0.618	0.725	0.852	0.180	40.27
GeoDiffuser [49]	SAM [26], DepthAnything [64]		**33.89**	**437.75**	0.173	0.762	0.938	0.166	34.88
DesignEdit [21]	SAM [26]		35.22	480.91	0.179	0.874	0.959	0.098	10.15
FreeFine	SAM [26]		34.72	478.18	0.144	**0.907**	**0.971**	**0.055**	**9.25**
DragDiffusion [50]	SAM [26]	3D	157.42	**1867.02**	0.348	0.603	**0.958**	0.199	61.97
Diffusion Handles [42]	SAM [26], LaMa [57], DepthAnything [64]		156.90	1882.66	0.523	0.705	0.882	0.128	26.10
GeoDiffuser [49]	SAM [26], DepthAnything [64]		152.06	1894.26	0.351	0.749	0.941	0.097	34.34
FreeFine	SAM [26], DepthAnything [64]		**150.89**	1879.69	**0.310**	**0.786**	0.956	**0.079**	**20.32**
BrushNet [22]	SAM [26]	SC	186.93	**2516.52**	**0.971**	0.925	0.948	0.060	11.31
SD-inpainting [54]	SAM [26]		193.71	2556.44	1.047	0.913	0.928	0.064	14.43
FreeFine	SAM [26]		**184.84**	2526.38	0.982	**0.928**	**0.952**	**0.056**	**9.56**

and Mean Distance(MD) as GeoDiffuser [49] to measure editing effectiveness, which warps the source object to the target location and then computes L1 error within masked regions of the generated images.

Baselines. Our evaluation includes two main aspects: (1) direct comparisons with state-of-the-art image editing methods, and (2) comparisons with representative inpainting methods integrated into our framework to address Step 2 and Step 3. For (1), we compare with DragonDiffusion [39], Self-Guidance [8], Motion-Guidance [10], and RegionDrag [33], DragDiffusion [50], GeoDiffuser [49], DiffusionHandles [42], and DesignEdit [21]. All methods are implemented based on their official codebases, with minimal adjustments to fit our benchmark (see Appendix). For (2), we compare with BrushNet [22], Stable Diffusion Inpainting [54], LaMa [57], and MAT [28]. Since LaMa [57] and MAT [28] learned to remove content (inpainting) rather than performing target region refinement, they are excluded from the main comparison but included in our user study (see Section 4.2).

4.2. Comparison with Other Methods

Quantitative Results. We conduct a comprehensive quantitative evaluation of FreeFine against SotA methods across 2D edits, 3D edits, and structure completion tasks in Table 1. FreeFine demonstrates consistent superiority across all scenarios and metrics: For 2D edits, FreeFine outperforms all the counterparts, with SUBC, WE and MD significantly better. Among compared methods, DesignEdit and GeoDiffuser stand out, where the former excels at edit precision and the latter is better at image quality. For 3D edits, methods (*e.g.*, DesignEdit [21], RegionDrag[33]) good at 2D edits fail to support holistic object 3D rotation by design. FreeFine undoubtedly achieves the best performance across all evaluation dimensions under the same depth-based transformation paradigm. For structure completion tasks, we further benchmark against representative inpainting models by integrating them into our editing pipeline. Our method comprehensively outperforms training-based methods like BrushNet, with no additional training needed.

Qualitative Results. We present a qualitative evaluation of our method against baseline approaches, focusing on both geometric editing and structural completion tasks. As illustrated in Fig. 4, our method achieves high-fidelity editing without noticeable artifacts and enables more accurate and diverse transformations. All other methods have clear artifacts: DragonDiffusion and RegionDrag leave residual artifacts in the Teacup example and fail in the rotate and resize examples; SelfGuidance, MotionGuidance and DiffusionHandles change the image contents significantly, explaining their lower SUBC and BC in Tab. 1. GeoDiffuser also faces residual artifacts and struggles in move example. DesignEdit, though excels at precision, falls short in the realism of edited results, corresponding to its relatively lower image quality in Tab. 1. Our method is the most successful approach in the 3D rotation example, while all other methods fail in this case.

We further compare our method with representative inpainting models in two key aspects: (a) Source Region Inpainting and (b) Target Region Refinement, as shown in Fig. 6. For (a), our method achieves less hallucination compared to BrushNet and SD-inpainting, while producing more realistic textures and finer details than LaMa and MAT. For (b), our method not only generates complex structures while maintaining contextual consistency with the object (*e.g.*, the bird and the horse) and producing more details (*e.g.*, dog's shadow).

User Study. For a comprehensive quantitative evaluation, we conducted a user study to assess the perceptual quality and editing effectiveness of our method. We recruited 35 participants with diverse backgrounds in computer vision and collected 2,622 valid votes. Each participant was presented with 30 randomly selected editing samples from different tasks (2D-edits, 3D-edits, region refinement and

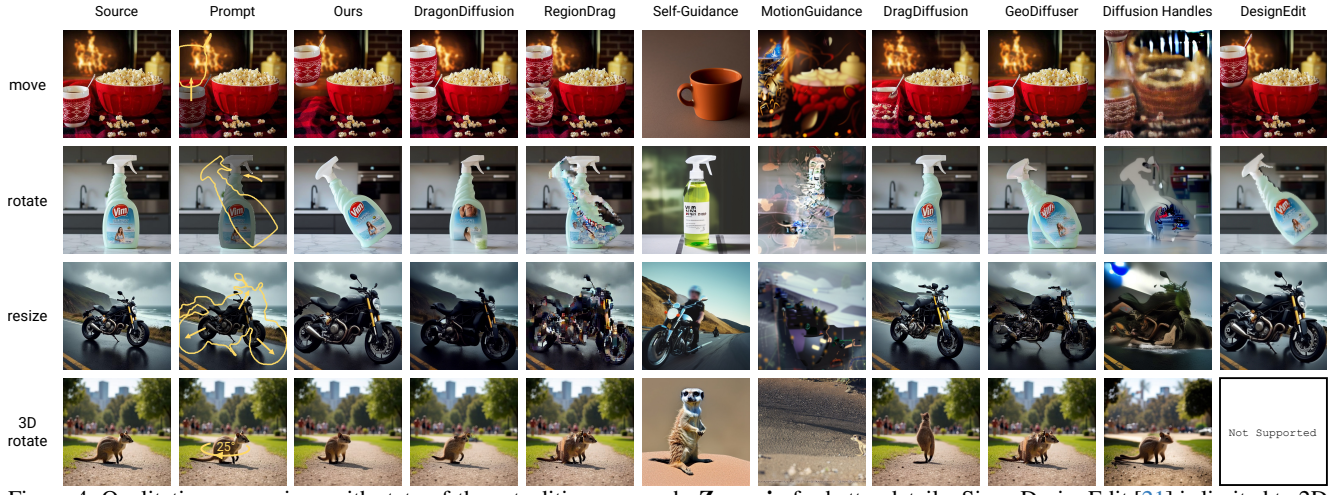


Figure 4. Qualitative comparison with state-of-the-art editing approach. **Zoom-in** for better details. Since DesignEdit [21] is limited to 2D layout-based editing and lacks 3D task support, its 3D editing result is marked with a "Not Supported" placeholder. Additional results are supplemented in the Appendix.

region inpainting). Each sample contained the original image along with a series of corresponding edited results generated by our method and other comparative models. Participants were asked to pick the best edited images based on: 1) **Image Quality**: How visually realistic and artifact-free the edited image appears; 2) **Consistency**: How well the edited image preserves the original subject and background; 3) **Editing Effectiveness**: How accurately the edited image reflects the intended geometric transformations (e.g., move, rotate, or resize). In the Appendix, we show our user study has close alignment with our quantitative metrics in Sec. 4.1.

As shown in Fig. 5 (a), our method performs well in all editing tasks, especially 2D-edits (70.2% user preference) and 3D-edits (88.8% user preference). Notably, while our method leads in most tasks, it is second to LaMa [57] in region inpainting, which is better at removing objects but tends to leave a blurry mess. As for the evaluation criteria in 2D-edits and 3D-edits (Fig. 5 (b)), our method is significantly better: the voting rates are all more than three times ahead. Despite DragonDiffusion [39]'s comparable FID scores to ours in Tab. 1, it lags greatly in perceptual quality assessments (20.8% vs. 71.1%). These findings further validate the robustness and practicality of our method in real-world geometric editing scenarios. More details about the user study and the statistics are included in the Appendix.

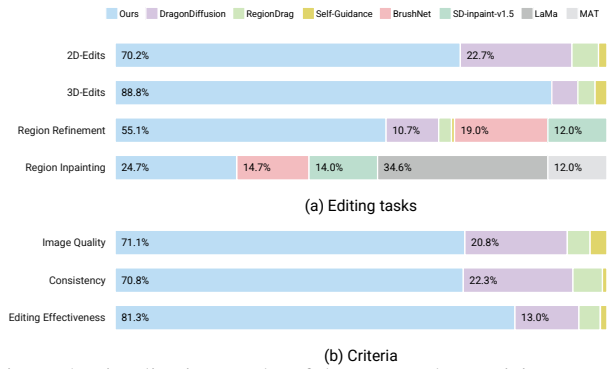
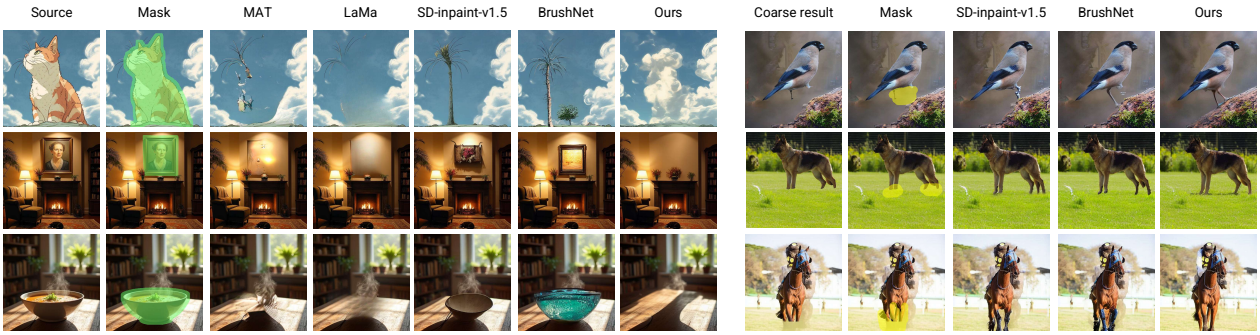


Figure 5. Visualization results of the user study. Participants preferred our edited images both in the different editing tasks and from three different criteria.

4.3. Ablation Studies

We conduct ablation studies on TCA, LP, and CG to dissect the impact of each component on the generated results. While ablating one component, we make sure other techniques are kept the same.

Attention Mechanisms. Beside MMSA [4], we compare TCA with two additional attention mechanisms: (a) Shared Self-Attention (SSA) [15], which shares a set of Key-Value (KV) pairs encoded from different text prompts to encourage the model to align features implicitly across the entire latent space (b) Subject-Driven Self-Attention (SDSA) [60], which selectively applies this sharing selectively only on the key and value vectors of foreground regions for object alignment. In our context, the shared KV pairs are constructed by concatenating the keys from the source and target $[K_s, K_t]$, and similarly for the values to form $[V_s, V_t]$. As shown in Fig. 7, for source region inpainting, SSA, SDSA, and standard Self-Attention (removing TCA) produce similar results where undesired changes appear in the source region, due to the non-constrained global feature sharing. MMSA explicitly restricts attention between the target region and the background region, achieving object removal but introducing texture artifacts (*e.g.*, stains on the wall in the second row). In contrast, due to the repair of the self-attention in larger denoisng timesteps, TCA generates better background. For target region refinement examples, standard Self-Attention alters details in regions where



(a) Source Region Inpainting (b) Target Region Refinement

Figure 6. Qualitative comparison with state-of-the-art inpainting methods. Notably, our method is **training-free**, while all the compared methods are training-based.

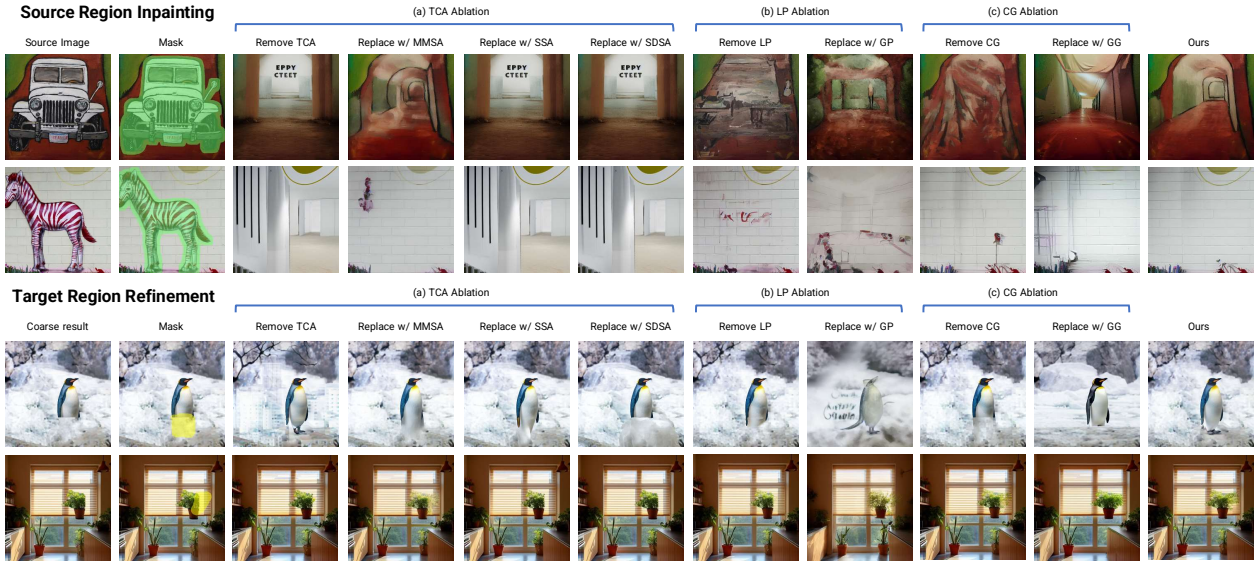


Figure 7. Ablation studies on the impact of removing individual components from **FreeFine** and different internal variations of each component while keeping other techniques applied at the same scale.

changes are not desired. Both SSA and SDSA do not incorporate masks, therefore showing limited completion capabilities. MMSA struggles when M_d contains semantics not present in M_s (*e.g.*, the penguin's feet), whereas TCA demonstrates smooth and robust completion performance, ensuring contextual consistency.

Perturbation and Guidance. We study the impact of LP and CG for both source region inpainting and target region refinement. Removing LP is equivalent to using DDIM deterministic denoising, whereas the other option is to use DDPM stochastic denoising, which we refer to as Global Perturbation (GP). Removing CG is equivalent to setting $\mathcal{C} = \varnothing$ in Eq. (2) and Eq. (3). Setting $\mathcal{M}_1 = 1$ in Eq. (2) and $\mathcal{M}_2 = 1$ in Eq. (3) leads to Global Guidance (GG). As illustrated in Fig. 7, for both source region inpainting and target region refinement, removing LP reduces randomness in results while using GP leads to undesired global changes. Similarly, removing CG eliminates explicit context guidance, hindering the generation of desired content. Replacing CG with global attention also alters textures across the entire image, reducing background and subject consistency.

5. Conclusion and Limitations

We present a principled framework for geometric image editing that systematically addresses the subtasks of object transformation, source region inpainting, and target region refinement. By decoupling these tasks, our approach more effectively balances large structural changes against fine-grained adjustments. The proposed FreeFine, equipped with Temporally Contextual Attention, Local Perturbation, and Content-specified Generation, demonstrates consistent gains in both fidelity and edit precision on our proposed GeoBench benchmark. A detailed discussion of limitations is provided in the Appendix. We believe that continued efforts in either addressing these limitations or developing more powerful and intuitive geometric image editing methods could benefit both the research community and industrial applications.

Acknowledgments

This work was supported by the NSFC (62441615 and 62225603).

References

[1] Yuval Alaluf, Daniel Garibi, Or Patashnik, Hadar Averbuch-Elor, and Daniel Cohen-Or. Cross-image attention for zero-shot appearance transfer. In *ACM SIGGRAPH*, pages 1–12, 2024. 3

[2] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023. 1, 2

[3] Mikolaj Binkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD gans. In *ICLR*. OpenReview.net, 2018. 5

[4] 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 *ICCV*, pages 22503–22513, 2023. 3, 4, 5, 7

[5] Aiyu Cui, Daniel McKee, and Svetlana Lazebnik. Dressing in order: Recurrent person image generation for pose transfer, virtual try-on and outfit editing. In *ICCV*, pages 14618–14627. IEEE, 2021. 2

[6] Yutao Cui, Xiaotong Zhao, Guozhen Zhang, Shengming Cao, Kai Ma, and Limin Wang. Stabledrag: Stable dragging for point-based image editing. In *ECCV*, pages 340–356, 2024. 2

[7] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *CoRR*, abs/1606.03798, 2016. 2

[8] Dave Epstein, Allan Jabri, Ben Poole, Alexei A. Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. In *NeurIPS*, 2023. 1, 2, 6

[9] Ben Fei, Zhaoyang Lyu, Liang Pan, Junzhe Zhang, Weidong Yang, Tianyue Luo, Bo Zhang, and Bo Dai. Generative diffusion prior for unified image restoration and enhancement. In *CVPR*, pages 9935–9946, 2023. 2

[10] Daniel Geng and Andrew Owens. Motion guidance: Diffusion-based image editing with differentiable motion estimators. In *ICLR*, 2024. 1, 2, 6

[11] Qin Guo and Tianwei Lin. Focus on your instruction: Fine-grained and multi-instruction image editing by attention modulation. In *CVPR*, pages 6986–6996, 2024. 4

[12] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S. Davis. VITON: an image-based virtual try-on network. In *CVPR*, pages 7543–7552. Computer Vision Foundation / IEEE Computer Society, 2018. 2

[13] Xu He, Zhiyong Wu, Xiaoyu Li, Di Kang, Chaopeng Zhang, Jiangnan Ye, Liyang Chen, Xiangjun Gao, Han Zhang, and Haolin Zhuang. Magicman: Generative novel view synthesis of humans with 3d-aware diffusion and iterative refinement. In *AAAI*, pages 3437–3445, 2025. 3

[14] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *ICLR*, 2023. 3, 4

[15] Amir Hertz, Andrey Voynov, Shlomi Fruchter, and Daniel Cohen-Or. Style aligned image generation via shared attention. In *CVPR*, pages 4775–4785, 2024. 3, 7

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, pages 6626–6637, 2017. 2, 5

[17] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *CoRR*, abs/2207.12598, 2022. 5

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2

[19] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024. 2, 5

[20] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In *NeurIPS*. Curran Associates, Inc., 2015. 2

[21] Yueru Jia, Yuhui Yuan, Aosong Cheng, Chuke Wang, Ji Li, Huizhu Jia, and Shanghang Zhang. Designedit: Multi-layered latent decomposition and fusion for unified & accurate image editing. *CoRR*, abs/2403.14487, 2024. 2, 6, 7

[22] Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In *ECCV*, pages 150–168, 2024. 2, 6

[23] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Pnp inversion: Boosting diffusion-based editing with 3 lines of code. *International Conference on Learning Representations (ICLR)*, 2024. 5

[24] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *NeurIPS*, 2022. 3

[25] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139:1–139:14, 2023. 2

[26] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. In *ICCV*, pages 3992–4003, 2023. 3, 6

[27] Ruining Li, Chuanxia Zheng, Christian Rupprecht, and Andrea Vedaldi. Dragapart: Learning a part-level motion prior for articulated objects. In *ECCV*, pages 165–183, 2024. 2

[28] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. MAT: mask-aware transformer for large hole image inpainting. In *CVPR*, pages 10748–10758, 2022. 2, 6

[29] Bingyan Liu, Chengyu Wang, Tingfeng Cao, Kui Jia, and Jun Huang. Towards understanding cross and self-attention in stable diffusion for text-guided image editing. In *CVPR*, pages 7817–7826. IEEE, 2024. 3

[30] Haofeng Liu, Chenshu Xu, Yifei Yang, Lihua Zeng, and Shengfeng He. Drag your noise: Interactive point-based editing via diffusion semantic propagation. In *CVPR*, pages 6743–6752, 2024. 2

[31] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*, pages 9264–9275. IEEE, 2023. 2

[32] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023. 2

[33] Jingyi Lu, Xinghui Li, and Kai Han. Regiondrag: Fast region-based image editing with diffusion models. In *ECCV*, pages 231–246, 2024. 1, 2, 6

[34] Andreas Lugmayr, Martin Danelljan, Andrés Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, pages 11451–11461, 2022. 2

[35] Grace Luo, Trevor Darrell, Oliver Wang, Dan B. Goldman, and Aleksander Holynski. Readout guidance: Learning control from diffusion features. In *CVPR*, pages 8217–8227, 2024. 2

[36] Xiaoqian Lv, Shengping Zhang, Chenyang Wang, Yichen Zheng, Bineng Zhong, Chongyi Li, and Liqiang Nie. Fourier priors-guided diffusion for zero-shot joint low-light enhancement and deblurring. In *CVPR*, pages 25378–25388, 2024. 2

[37] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *NeurIPS*, pages 406–416, 2017. 2

[38] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, pages 405–421. Springer, 2020. 2

[39] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Dragondiffusion: Enabling drag-style manipulation on diffusion models. In *ICLR*, 2024. 1, 2, 6, 7

[40] Jiteng Mu, Michaël Gharbi, Richard Zhang, Eli Shechtman, Nuno Vasconcelos, Xiaolong Wang, and Taesung Park. Editable image elements for controllable synthesis. In *ECCV*, pages 39–56, 2024. 2

[41] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. Drag your GAN: interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH*, pages 78:1–78:11, 2023. 2

[42] Karran Pandey, Paul Guerrero, Matheus Gadelha, Yannick Hold-Geoffroy, Karan Singh, and Niloy J. Mitra. Diffusion handles enabling 3d edits for diffusion models by lifting activations to 3d. In *CVPR*, pages 7695–7704. IEEE, 2024. 2, 6

[43] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024. 1

[44] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, pages 8821–8831, 2021. 2

[45] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022. 2

[46] Jiawei Ren, Mengmeng Xu, Jui-Chieh Wu, Ziwei Liu, Tao Xiang, and Antoine Toisoul. Move anything with layered scene diffusion. In *CVPR*, pages 6380–6389, 2024. 2

[47] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10674–10685, 2022. 1, 2, 5

[48] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022. 1, 2

[49] Rahul Sajnani, Jeroen van Baar, Jie Min, Kapil Katyal, and Srinath Sridhar. Geodiffuser: Geometry-based image editing with diffusion models. In *WACV*, pages 472–482, 2025. 2, 6

[50] Yujun Shi, Chuhui Xue, Jun Hao Liew, Jiachun Pan, Hanshu Yan, Wenqing Zhang, Vincent Y. F. Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *CVPR*, 2024. 1, 2, 6

[51] Joonghyuk Shin, Daehyeon Choi, and Jaesik Park. Instantdrag: Improving interactivity in drag-based image editing. In *ACM SIGGRAPH*, pages 39:1–39:10, 2024. 2

[52] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable gans for pose-based human image generation. In *CVPR*, pages 3408–3416. Computer Vision Foundation / IEEE Computer Society, 2018. 2

[53] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 2, 5

[54] Lorenzo Stacchio. Train stable diffusion for inpainting, 2023. 2, 6

[55] George Stein, Jesse C. Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Leigh Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L. Caterini, J. Eric T. Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. In *NeurIPS*, 2023. 5

[56] Wenhao Sun, Xue-Mei Dong, Benlei Cui, and Jingqun Tang. Attentive eraser: Unleashing diffusion model's object removal potential via self-attention redirection guidance. In *AAAI*, pages 20734–20742, 2025. 2

[57] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *WACV*, pages 3172–3182, 2022. 2, 6, 7

[58] Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. Ominicontrol: Minimal and universal control for diffusion transformer. *CoRR*, abs/2411.15098, 2024. 5

[59] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. In *ICCV*, pages 402–419, 2020. 6

[60] Yoad Tewel, Omri Kaduri, Rinon Gal, Yoni Kasten, Lior Wolf, Gal Chechik, and Yuval Atzmon. Training-free consistent text-to-image generation. *ACM Transactions on Graphics (TOG)*, 43(4):1–18, 2024. 3, 7

[61] 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 *ECCV*, pages 439–457, 2024. 2, 3

[62] Bochao Wang, Huabin Zheng, Xiaodan Liang, Yimin Chen, Liang Lin, and Meng Yang. Toward characteristic-preserving image-based virtual try-on network. In *ECCV*, pages 607–623. Springer, 2018. 2

[63] Jie Xiao, Ruili Feng, Han Zhang, Zhiheng Liu, Zhantao Yang, Yurui Zhu, Xueyang Fu, Kai Zhu, Yu Liu, and Zheng-Jun Zha. Dreamclean: Restoring clean image using deep diffusion prior. In *ICLR*, 2024. 3

[64] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, pages 10371–10381, 2024. 2, 3, 6

[65] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Comput. Surv.*, 56(4):105:1–105:39, 2024. 2

[66] Jiraphon Yenphraphai, Xichen Pan, Sainan Liu, Daniele Panozzo, and Saining Xie. Image sculpting: Precise object editing with 3d geometry control. In *CVPR*, pages 4241–4251, 2024. 2

[67] Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. In *CVPR*, pages 1219–1229, 2023. 2

[68] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *CVPR*, pages 2347–2356, 2019. 2

[69] Zhen Zhu, Tengteng Huang, Mengde Xu, Baoguang Shi, Wenqing Cheng, and Xiang Bai. Progressive and aligned pose attention transfer for person image generation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(8):4306–4320, 2022. 2