EditInfinity: Image Editing with Binary-Quantized Generative Models

Jiahuan Wang[*1] **Yuxin Chen**[*2] **Jun Yu**[1] **Guangming Lu**[†1] **Wenjie Pei**[†1]

[1]Harbin Institute of Technology, Shenzhen [2]The Hong Kong University of Science and Technology

wangjiahuanszhit@163.com ychenqa@connect.ust.hk yujun@hit.edu.cn
luguangm@hit.edu.cn wenjiecoder@outlook.com

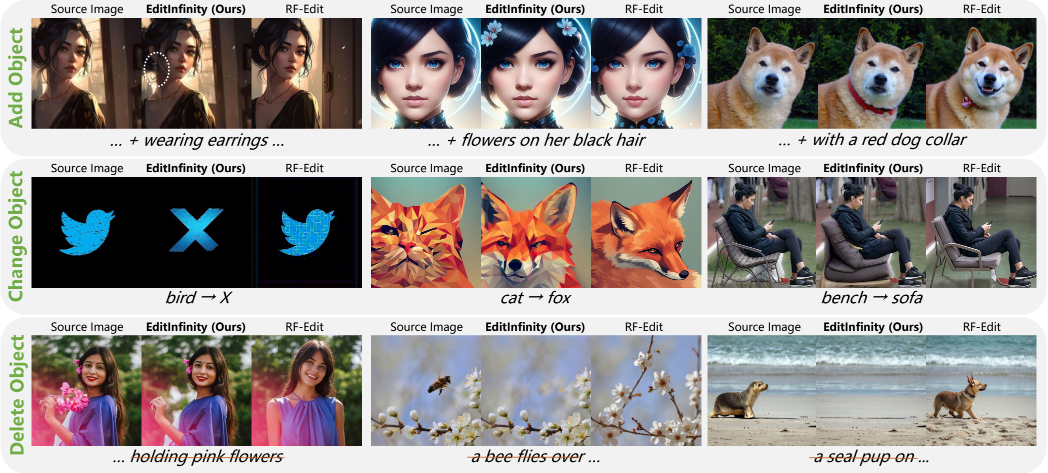


Figure 1: Our method, *EditInfinity*, delivers strong performance in **background preservation** in unedited regions and **text alignment** in edited regions across diverse editing tasks, including add, change and delete object, showing clear advantages over the latest state-of-the-art diffusion-based method RF-Edit [53], as illustrated by representative examples.

Abstract

Adapting pretrained diffusion-based generative models for text-driven image editing with negligible tuning overhead has demonstrated remarkable potential. A classical adaptation paradigm, as followed by these methods, first infers the generative trajectory inversely for a given source image by image inversion, then performs image editing along the inferred trajectory guided by the target text prompts. However, the performance of image editing is heavily limited by the approximation errors introduced during image inversion by diffusion models, which arise from the absence of exact supervision in the intermediate generative steps. To circumvent this issue, we investigate the parameter-efficient adaptation of binary-quantized generative models for image editing, and leverage their inherent characteristic that the exact intermediate quantized representations of a source image are attainable, enabling more effective supervision for precise image inversion. Specifically, we propose *EditInfinity*, which adapts *Infinity*, a binary-quantized generative model, for image editing. We propose an efficient yet effective image inversion mechanism that integrates text prompting rectification and image style preservation, enabling precise image inversion. Furthermore, we devise a

* Equal contribution. † Corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

holistic smoothing strategy which allows our *EditInfinity* to perform image editing with high fidelity to source images and precise semantic alignment to the text prompts. Extensive experiments on the PIE-Bench benchmark across ‘add’, ‘change’, and ‘delete’ editing operations, demonstrate the superior performance of our model compared to state-of-the-art diffusion-based baselines. Code available at: `https://github.com/yx-chen-ust/EditInfinity`.

1 Introduction

Text-driven image editing aims to modify the content of an image in accordance with the given text prompts while maintaining the integrity of the unedited regions. In contrast to training-from-scratch methods [11, 22, 63] that incur expensive training costs, the adaptation of pre-trained models, particularly diffusion-based generative models, with lightweight fine-tuning overhead has emerged as a predominant paradigm for image editing, demonstrating remarkable potential [29, 21, 43].

A classical adaptation paradigm in diffusion models for image editing [29, 20, 2] consists of two essential steps: 1) image inversion, which aims to infer the generative trajectory along the sampling process in reverse for a given source image, striving to reconstruct the image accurately, and 2) image editing, conducted along the inferred trajectory guided by the target text prompts. Consequently, the precision of image inversion is critical to the performance of image editing. Nevertheless, it is intractable to obtain the exact sampling trajectory of a source image for a pretrained diffusion model. Thus, image inversion is either performed employing the deterministic sampling technique [5, 42, 10, 15, 20, 56] to approximate the intermediate noisy representations along the reversed sampling path, or it is formulated as a optimization problem to finetune the pretrained diffusion model to fit the approximate intermediate results along the sampling path [29, 9]. Consequently, a potential limitation of this adaptation paradigm of diffusion models for image editing is that the performance of image editing is heavily constrained by the approximate errors introduced during image inversion.

To address aforementioned limitation, in this work, we investigate the parameter-efficient adaptation of binary-quantized generative models for image editing. Unlike diffusion models, binary-quantized generative models quantize images into a discrete latent space and model the data distribution in this quantized space for generation. Thus, an inherent characteristic of binary-quantized models is that the exact quantized representations for an arbitrary image can be directly inferred, potentially enabling more precise image inversion. Motivated by this observation, we propose *EditInfinity*, which adapts *Infinity*—a binary-quantized generative model with powerful text-to-image generation capability—for image editing, following the classical ‘image inversion-image editing’ adaptation paradigm.

Considering a pretrained *Infinity* as a mapping function between the distribution of textual prompts and image data distribution, performing inverse inference on the pretrained model to obtain the exact textual embedding for a source image is intractable, whereas the user-provided source text prompts generally cannot precisely match with the source image. Therefore, we formulate the image inversion process of *EditInfinity* as an optimization problem, aiming to learn an accurate textual embedding for a given source image, guided by provided source text prompts. A notable advantage of this design is that the intermediate multi-scale quantized representations by *Infinity* for the source image can be utilized as exact supervision to optimize the image inversion process, yielding precise image inversion and thereby, high-quality image editing. To conclude, we make the following contribution.

- We propose *EditInfinity*, which apply the classical ‘image inversion-image editing’ adaptation paradigm to *Infinity*, a prominent binary-quantized model, to investigate the parameter-efficient adaptation of binary-quantized generative models for image editing.
- We design an efficient yet effective image inversion mechanism comprising text prompting rectification and image style preservation, leveraging the quantized representations as exact supervision to enable precise image inversion.
- We devise a holistic smoothing strategy which allows our *EditInfinity* to perform image editing with high fidelity to source image and precise semantic alignment to the text prompts.
- We conduct extensive experiments on the PIE-Bench benchmark and comprehensively demonstrate the superior performance of our *EditInfinity* compared to state-of-the-art diffusion-based approaches across diverse editing operations, excelling in both background preservation and semantic alignment with target text prompts.

2 Related Work

2.1 Image Editing with Diffusion Models

Image editing researches [27, 16, 21, 31, 4] have been predominantly driven by diffusion models [36, 38, 33, 22], and are broadly categorized into training-based and training-free paradigms [40]. Training-based methods [3, 58, 11, 39, 18, 24, 22] achieve impressive editing capabilities, but their requirement for an expensive training dataset limits practical applicability. In contrast, training-free [48, 23, 43] methods have emerged as a more flexible alternative, establishing an inversion-editing paradigm [40]. The inversion stage focuses on accurate latent code inversion. Recent works [29, 9, 28, 20, 53] have developed improved inversion samplers to ease the inherent reconstruction inaccuracies. In the editing stage, numerous methods [16, 47, 4, 1, 32, 25, 26] leverage attention in diffusion models to edit while preserving overall image structure. Despite these advances, a key limitation remains: existing methods fail to preserve both text alignment fidelity and source image consistency. This trade-off between editability and faithfulness motivates us to investigate more robust editing frameworks.

2.2 Autoregressive Image Generation Models

Autoregressive models have demonstrated remarkable scalability in image generation by leveraging next-token prediction, a paradigm inherited from LLMs (**L**arge **L**anguage **M**odel**s**) [52]. Early methods like PixelCNN [49] and PixelRNN [50] model pixels directly, but their quadratic dependency growth makes high-resolution generation impractical. Thus, subsequent works avoid modeling the data distribution directly in pixel space and instead model it in a compact latent space. As a pioneering work, VQVAE [51] constructs a discrete latent space by vector quantization and learns the underlying latent distribution by autoregressive models. Recently, VAR (**V**isual **A**uto**R**egressive Modeling) [46] reformulates autoregressive image generation as a next-scale prediction task, capturing global structural priors to achieve state-of-the-art generation quality while improving sampling speed.

The success of autoregressive models naturally extends to text-to-image generation. Pioneering works like DALL-E [35] and CogView [8] unify text and image tokens within a single transformer decoder. Subsequently, Parti [57] and LlamaGen [44] decouple text and image processing by employing dedicated text encoders to guide the autoregressive decoder. Then, HART [45] integrates VAR's hybrid tokenizers to improve generation quality. Latest, Infinity [14] advances autoregressive image generation by introducing Bitwise Visual AutoRegressive Modeling. It establishes a new foundational model for autoregressive text-to-image models and achieves competitive results with diffusion-based approaches. As our method builds on Infinity, we outline its architecture in Section 3.

3 Preliminary: Infinity

Bitwise Multi-scale Residual Quantization. An image $I \in \mathbb{R}^{H\times W\times 3}$ is first encoded into the original feature F, which is then tokenized into bitwise multi-scale residual maps $\{R_k\}_{k=1}^{K}$ through iterative residual approximation. At scale k, residual features are computed between the original feature F and the cumulative feature F_{k-1} from previous scales.

$$z_k = \text{down}\left(F - F_{k-1}, (h_k, w_k)\right) \in \mathbb{R}^{h_k\times w_k\times d}, \tag{1}$$

where down$(\cdot)$ performs bilinear downsampling to target resolution (h_k, w_k). To quantize residuals, Infinity adopts BSQ (**B**inary **S**pherical **Q**uantization) [61]:

$$R_k = \mathcal{Q}(z_k) = \frac{1}{\sqrt{d}}\text{sign}\left(\frac{z_k}{\|z_k\|}\right) \in \{\frac{-1}{\sqrt{d}}, \frac{1}{\sqrt{d}}\}^{h_k\times w_k\times d}. \tag{2}$$

Then, the cumulative feature F_k at scale k is computed recursively:

$$F_k = \sum_{i=1}^{k} \text{up}(R_i, (h_K, w_K)) \in \mathbb{R}^{h_K\times w_K\times d}, \tag{3}$$

where up$(\cdot)$ denotes bilinear upsampling.

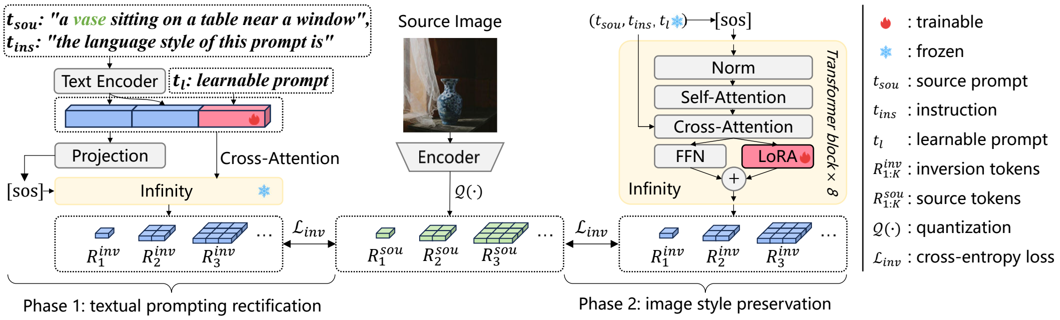
Bitwise Autoregressive Modeling. The transformer predicts residuals autoregressively across K scales, conditioned on the prompt t. Formally, the autoregressive likelihood is:

$$p(R_{1:K}|\Psi(t)) = \prod_{k=1}^{K} p\big(R_k|\underbrace{R_1,\dots,R_{k-1}}_{\text{all previous scales}},\Psi(t)\big), \tag{4}$$

where $\Psi(\cdot)$ denotes Flan-T5 [6]. To tackle the large codebook challenge, Infinity proposes the Infinite-Vocabulary Classifier, which decomposes the prediction into d independent binary classifiers.

4 Method

Successful image editing requires precise content modifications that semantically align with target prompts while remaining faithful to unedited regions. To this end, a classical adaptation paradigm repurposes a pretrained text-to-image generative model to image editing through two steps: 1) image inversion, which inversely infers the corresponding generative trajectory for the source image by reversing the sampling process, and 2) image editing, performed along the inferred generative trajectory guided by the target text prompts. Our proposed *EditInfinity* applies this paradigm to Infinity [14], a binary-quantized generative model, harnessing the inherent characteristics of quantized generative models to potentially achieve precise image inversion and high-quality image editing.



t_{sou}: "a vase sitting on a table near a window",
t_{ins}: "the language style of this prompt is"
Text Encoder
t_l: learnable prompt
Projection
Cross-Attention
[sos]
Infinity
R_1^{inv} R_2^{inv} R_3^{inv} …
$\mathcal{L}_{inv}$
Source Image
Encoder
$\mathcal{Q}(\cdot)$
R_1^{sou} R_2^{sou} R_3^{sou} …
$\mathcal{L}_{inv}$
(t_{sou}, t_{ins}, t_l) → [sos]
Norm
Self-Attention
Cross-Attention
FFN
LoRA
+
Infinity
Transformer block × 8
R_1^{inv} R_2^{inv} R_3^{inv} …
Phase 1: textual prompting rectification
Phase 2: image style preservation
: trainable
: frozen
t_{sou} : source prompt
t_{ins} : instruction
t_l : learnable prompt
$R_{1:K}^{inv}$: inversion tokens
$R_{1:K}^{sou}$: source tokens
$\mathcal{Q}(\cdot)$: quantization
$\mathcal{L}_{inv}$: cross-entropy loss

Figure 2: **Image Inversion with Exact Supervision.** Given a source image I_{sou} and its prompt t_{sou}, we first quantize I_{sou} into exact tokens $R^{sou}_{1...K}$. Then, we concatenate t_{sou} with an instruction t_{ins} and a learnable prompt t_l, which is optimized via $\mathcal{L}_{inv}$ under the supervision of $R^{sou}_{1...K}$. Afterwards, the prompt is frozen, and LoRA is applied to the FFN layers of Infinity to further reconstruct I_{sou}.

4.1 Image Inversion with Exact Supervision

A text-to-image generative model performs image generation by learning a mapping from the distribution of text prompts to image data distribution. However, since the mapping function is unknown, it is intractable to inversely obtain the exact textual embedding for a given image. Meanwhile, the user-provided source text prompt generally cannot precisely match the source image. To circumvent this problem, we formulate the image inversion process as an optimization problem with exact supervision to infer the text embedding precisely matched with the source image:

$$\mathcal{L}_{inv} = -\frac{1}{K}\sum_{k=1}^{K}\left(R_k^{sou}\cdot\log p(R_k^{inv}|R_{<k}^{sou},\Psi(t))\right), \tag{5}$$

where $\mathcal{L}_{inv}$ is formulated as a cross-entropy loss applied to each inversion token R_k^{inv}. Compared to the diffusion-based models for image editing, a key advantage of binary-quantized generative models is that the exact groundtruth of the intermediate outputs ($R^{sou}_{1...K}$ in 'Infinity') for a given image along the generative trajectory is attainable by token-wise quantization, enabling exact supervision for optimization of image inversion in Equation 5.

Textual prompting rectification. To guide the optimization of Equation 5 toward a text embedding that matches the source image, we treat the source prompt t_{sou} as a reference and apply text prompting tuning to rectify it into a semantically aligned textual condition. Concretely, we first augment t_{sou}

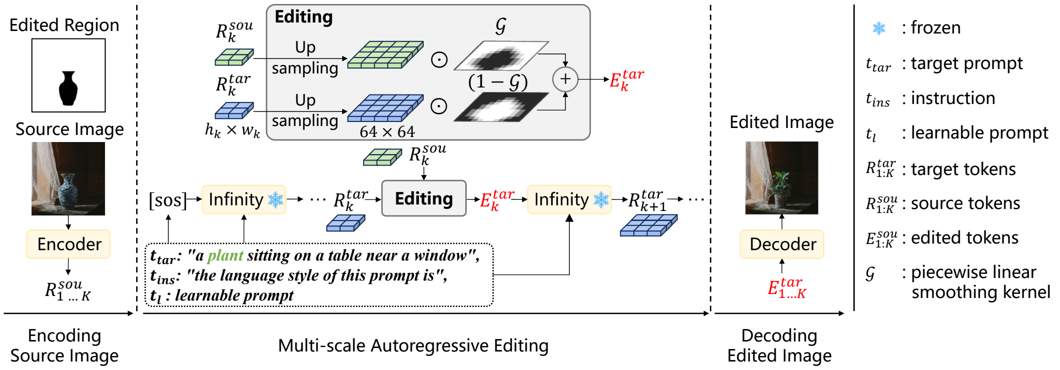


Figure 3: **Image Editing with Holistic Smoothing.** First, source image is encoded into $R^{sou}_{1\ldots K}$. At each step k of autoregressive generation, generated R^{tar}_k is conditioned on the concatenation of the target prompt t_{tar}, instruction t_{ins}, and optimized learnable prompt t_l and then, is blended with R^{sou}_k guided by piecewise linear smoothing kernel $\mathcal{G}$, forming edited tokens E^{tar}_k to prepare for guiding the next-scale generation. Finally, $E^{tar}_{1\ldots K}$ is decoded into the edited image.

with 20 learnable prompt tokens t_l and an instruction prompt t_{ins} (e.g., "the language style of this prompt is") to bridge the semantic gap between the source prompts and the solution. Second, we pass t_{sou} and t_{ins} through the text encoder $\Psi(\cdot)$ of Infinity to obtain text embeddings $\Psi(t_{sou}, t_{ins})$. We then concatenate those embeddings with t_l to form the textual conditioning input $[\Psi(t_{sou}, t_{ins}), t_l]$ for Infinity. Finally, we freeze all Infinity parameters and optimize only t_l using cross-entropy loss, where the supervision signals are exact tokens $R^{sou}_{1\ldots K}$ derived from the source image.

Image style preservation. While the learnable prompt adapt the semantic content, they may fall short in preserving structural style characteristics. The low-rank bias [17, 19, 41] (rank $\ll \dim(W)$) favors smooth and global modifications to the output distribution, thereby encouraging reconstructions that preserve overall structure and appearance while avoiding overfitting to high-frequency artifacts. To this end, we employ LoRA [17] to refine the pretrained weights W with minimal overhead ΔW (only inserts trainable low-rank matrices into FFN layers [52]) after rectifying textual prompt. Then, the learned ΔW is retained during editing, allowing the model to faithfully preserve global style traits of the source image even when applying novel target prompts.

4.2 Image Editing with Holistic Smoothing

We aim to manipulate only the desired regions while preserving the structural integrity of unedited areas. To this end, we introduce a precise token replacement strategy that enables localized, semantically aligned editing at the token level. Given the optimized learnable prompt t_l and LoRA ΔW, we perform conditional generation under the target prompt t_{tar}, instruction t_{ins} and optimized learnable prompt t_l, which ensures the edited image adheres to the target semantics while maintaining structural fidelity with the source image.

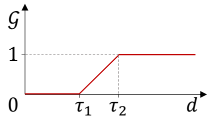


Figure 4: $\mathcal{G}$ as a function of d, enabling smooth transitions from edited to unedited regions.

Piecewise Linear Smoothing Kernel. The core idea of our editing paradigm is to construct the edited tokens $E^{tar}_{1:K}$ by blending source tokens $R^{sou}_{1:K}$ and target tokens $R^{tar}_{1:K}$ in a spatially controlled manner. A direct blend will result in a splicing phenomenon, so we first localize the edit with a user-provided mask M—a standard setting in image editing [30, 62] where text-only prompts often lack spatial specificity [16, 37, 59]. Then, we define a piecewise linear smoothing kernel $\mathcal{G}$ to guide the blending. Specifically, $\mathcal{G}$ is defined over the Manhattan distance d to calculate location weights per location, as in Equation 6:

$$\mathcal{G}^{i,j} = \begin{cases} 0, & d^{i,j} \leq \tau_1 \\ \frac{d^{i,j}-\tau_1}{\tau_2-\tau_1}, & \tau_1 < d^{i,j} < \tau_2 \ , \\ 1, & d^{i,j} \geq \tau_2 \end{cases} \quad d^{i,j} = \min_{(x,y)\in M} \left(|i-x| + |j-y|\right). \tag{6}$$

Here, $d^{i,j}$ denotes the Manhattan distance from token (i, j) to the nearest token within M. The kernel $\mathcal{G}^{i,j}$ is designed to gradually transition from 0 to 1 within a controllable band defined by thresholds τ_1 and τ_2, which is visualized in Figure 4. Specifically, tokens within a distance of τ_1 from the edit region are assigned zero weight to encourage full preservation from target content, while those beyond τ_2 are fully replaced by the source. Tokens in the intermediate band are assigned weights via linear interpolation, facilitating smooth blending between source and target content. This formulation effectively suppresses boundary artifacts by promoting seamless transitions between source and edited regions.

Multi-scale Autoregressive Editing. Building on image inversion and the piecewise linear smoothing kernel $\mathcal{G}$, we realize image editing as a multi-scale autoregressive token-replacement process. At each scale, generated target tokens are blended with source tokens under spatial weights provided by $\mathcal{G}$, and the resulting edited tokens serve as context for the next scale. This coarse-to-fine schedule localizes semantic changes to the masked region while preserving global structure elsewhere. Algorithm 1 details the procedure.

Our algorithm begins by quantizing I_{sou} to extract precise source tokens $R^{sou}_{1:K}$. At each scale k, Infinity$(\cdot)$ generates the target token R^{tar}_k conditioned on previous tokens $\hat{R}^{tar}_{<k}$ and prompts embedding $[\Psi(t_{tar}, t_{ins}), t_l]$. We then upsample R^{tar}_k and R^{sou}_k to (h_K, w_K) and blend them under the guidance of $\mathcal{G}$ to obtain the edited token E^{tar}_k. This aligns edited regions with target semantics while preserving source fidelity elsewhere. If $k < K$, we downsample E^{tar}_k to (h_{k+1}, w_{k+1}) to form $\hat{R}^{\text{tar}}_k$, which serves as the autoregressive state at the next scale, allowing blended semantics and structure to propagate across scales. After traversing all scales, edited tokens $E^{tar}_{1:K}$ are decoded to edited image I_{tar}.

Algorithm 1 Multi-scale Autoregressive Editing

```
Inputs: source image $I_{sou}$; target prompt $t_{tar}$; instruction $t_{ins}$; optimized learnable prompt $t_l$;
Hyperparameters: scales $K$, resolutions$(h_k, w_k)^K_{k=1}$
$R^{sou}_{1...K} = \mathcal{Q}(\mathcal{E}(I_{sou}))$        ▷ $\mathcal{E}$: encoder; $\mathcal{Q}$: quantizer
$[\Psi(t_{tar}, t_{ins}), t_l]$ projected into $\hat{R}^{tar}_0$(i.e., [sos])
for $k = 1$ to $K$ do
    $R^{tar}_k = \text{Infinity}(\hat{R}^{tar}_{<k}, [\Psi(t_{tar}, t_{ins}), t_l])$
    $E^{tar}_k = \text{Upsample}(R^{tar}_k, (h_K, w_K)) \odot (1 - \mathcal{G}) +$
             $\text{Upsample}(R^{sou}_k, (h_K, w_K)) \odot \mathcal{G}$
    if $k < K$ then
        $\hat{R}^{tar}_k = \text{Downsample}(E^{tar}_k, (h_{k+1}, w_{k+1}))$
    end if
end for
$I_{tar} = \mathcal{D}(E^{tar}_{1...K})$        ▷ $\mathcal{D}$: decoder
Return edited Image $I_{tar}$
```

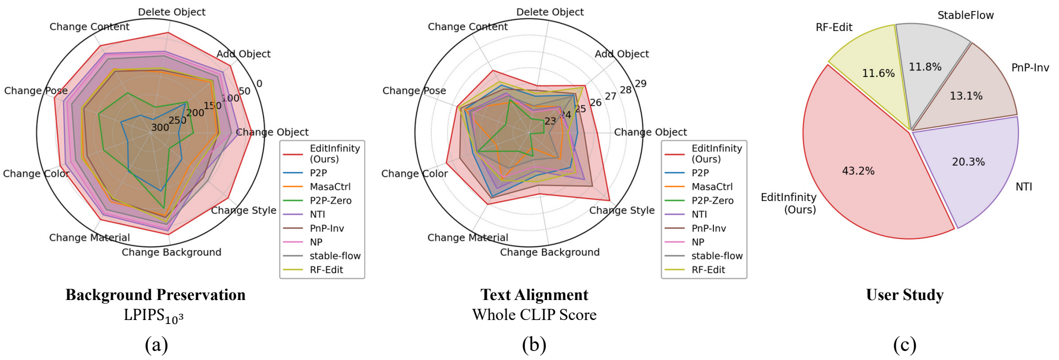


Figure 5: **Comprehensive performance evaluation on PIE-Bench.** (a) and (b) report background preservation and text alignment metrics across nine tasks. (c) summarizes user study preferences.

5 Experiments

5.1 Experimental Setup

Comparison Methods. We compare our method against a range of methods. (1) Open-source methods: including Diffusion UNet models—P2P [16], MasaCtrl [4], P2P-Zero [31], NTI [29], PnP-Inv [20], and NP [28]—and Diffusion Transformer models—StableFlow [2] and RF-Edit [53]. (2) Closed-source method: Gemini 2.0 [13], a current frontier of large-scale commercial model.

Benchmark. We conduct comprehensive experiments on PIE-Bench (**P**rompt-based **I**mage **E**diting **Bench**mark) [20], the prevailing standard in image editing evaluation. This benchmark contains 700 test cases covering nine editing types. Each case provides a source image with a corresponding prompt, target editing prompt, and the editing mask.

Metrics. Our evaluation employs seven carefully selected metrics across two critical dimensions. For background preservation, we use four complementary metrics: PSNR and MSE for pixel-level accuracy, LPIPS [60] for perceptual similarity, and SSIM [54] for structural similarity. For text-image alignment, we report CLIP scores [34] of the whole image and the edited region with the target prompt. Additionally, we adopt IR (**I**mage **R**eward [55]), a learned metric trained on human preference data, specifically sensitive to editing failures, often assigning negative scores to failed outputs.

Implementation Details. We implement our method based on Infinity-2B [0]. For editing, we set $\tau_1 = 1$ and $\tau_2 = 4$ in Equation 6. Inversion is trained on two NVIDIA L20 GPUs, and editing runs on a single NVIDIA L20 GPU. Refer to Supplementary Material A.2 for more details.

Table 1: **Quantitative results on PIE-Bench.** In the 'Base Model' column, 'U', 'T', and 'A' represent Diffusion UNet, Diffusion Transformer, and Autoregressive models, respectively. Diffusion UNet models employ Stable Diffusion v1.4, with the exception of PnP-Inv, which utilizes v1.5. Diffusion Transformer models leverage FLUX.1-dev, while Autoregressive models use Infinity.

Method	Venue	Base Model	Background Preservation				Text Alignment		
			PSNR↑	LPIPS$_{10^3}$↓	MSE$_{10^4}$↓	SSIM$_{10^2}$↑	Whole↑	Edited↑	IR$_{10}$↑
P2P[16]	ICLR'23	U	17.87	208.80	219.88	71.14	25.01	22.44	0.017
MasaCtrl[4]	ICCV'23		22.17	106.62	86.97	79.67	23.96	21.16	-1.66
P2P-Zero[31]	SIGGRAPH'23		20.44	172.22	144.12	74.67	22.80	20.54	-6.59
NTI[29]	CVPR'23		27.03	60.67	35.86	84.11	24.75	21.86	2.77
PnP-Inv[20]	ICLR'24		22.46	106.06	80.45	79.68	25.41	22.62	4.17
NP[28]	WACV'25		26.21	69.01	39.73	83.40	24.61	21.87	2.42
StableFlow[2]	CVPR'25	T	21.64	92.28	115.21	84.94	24.65	21.70	1.88
RF-Edit[53]	ICML'25		23.22	131.18	75.00	81.44	25.22	22.40	5.18
Gemini[13]	-	-	23.22	105.17	188.63	81.10	25.28	22.28	5.30
EditInfinity	NeurIPS'25	A	**27.95**	**33.08**	**24.27**	**92.12**	**26.41**	**23.47**	**5.88**

Table 2: **Evaluation of Base Models on the GenEval Benchmark.** When evaluating Infinity, we adopt the same evaluation protocol as used for Stable Diffusion v1.4, v1.5, and FLUX.1-dev, i.e., without prompt rewriting.

Base Model	Overall	Single Object	Two Object	Counting	Colors	Position	Attribute Binding
Stable Diffusion v1.4	0.42	0.97	0.39	0.33	0.73	0.03	0.05
Stable Diffusion v1.5	0.43	0.97	0.38	0.35	0.76	0.04	0.06
FLUX.1-dev	**0.66**	**0.98**	**0.81**	**0.74**	0.79	0.22	0.45
Infinity-2B	**0.66**	**0.98**	0.78	0.63	**0.83**	**0.25**	**0.53**

5.2 Comparison to State-of-the-Arts

Quantitative Results. As demonstrated in Table 1, our method sets a new state-of-the-art in text-driven image editing by significantly improving the trade-off between two key objectives: (1) rigorous background preservation and (2) precise text-aligned editing. While existing methods struggle with this inherent trade-off, our framework achieves a superior balance, outperforming all others by notable margins in both aspects. Notably, our method attains the best IR$_{10}$ score (5.88), reflecting substantially higher editing success rates than competing methods. We further provide task-wise comparisons of LPIPS and full CLIP scores across all edit types. As shown in Figure 5 (a) and (b), these results consistently validate the effectiveness of our method across diverse editing scenarios.

[0] `https://huggingface.co/FoundationVision/Infinity/blob/main/infinity_2b_reg.pth`

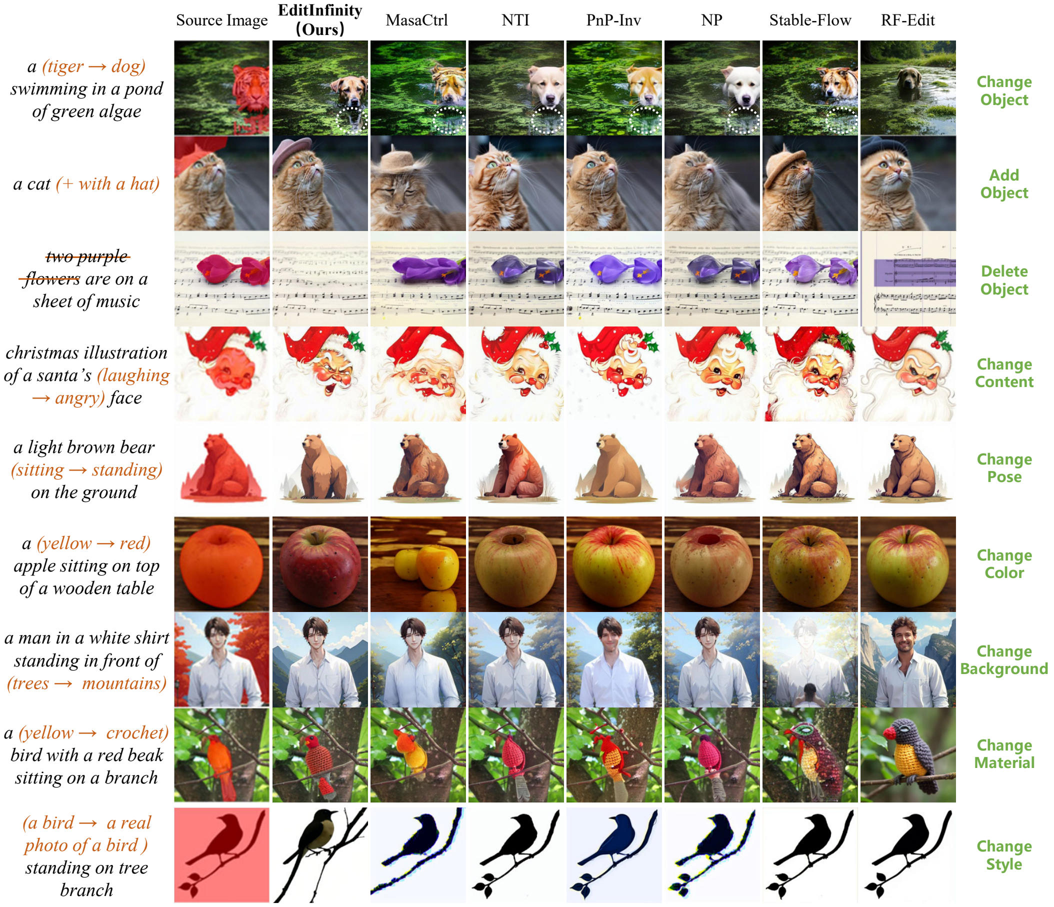
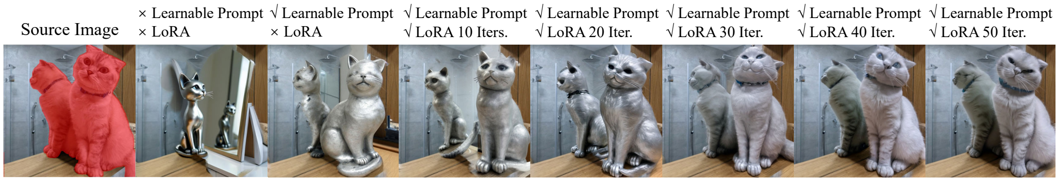

Source Image
EditInfinity (Ours)
MasaCtrl
NTI
PnP-Inv
NP
Stable-Flow
RF-Edit
a (tiger → dog) swimming in a pond of green algae
Change Object
a cat (+ with a hat)
Add Object
~~two purple flowers~~ are on a sheet of music
Delete Object
christmas illustration of a santa's (laughing → angry) face
Change Content
a light brown bear (sitting → standing) on the ground
Change Pose
a (yellow → red) apple sitting on top of a wooden table
Change Color
a man in a white shirt standing in front of (trees → mountains)
Change Background
a (yellow → crochet) bird with a red beak sitting on a branch
Change Material
(a bird → a real photo of a bird) standing on tree branch
Change Style

Figure 6: **Qualitative results on PIE-Bench across all nine tasks.** The red mask denotes the edited region M, expected to follow the target prompt, while other regions retain the background.

To ensure that the advantage of our method is not only attributed to the stronger generative capacity of base model, we further analyze the base models used by all compared methods. Although our framework is lightweight and tailored to Infinity, its reliance on the base model's generative capacity is consistent with other editing paradigms. As reported in Table 2, we evaluate each method's base model on the GenEval benchmark [12]. Infinity performs comparably to the popular FLUX [22] and even underperforms in certain tasks (e.g., two-object and counting). Nevertheless, our Infinity-based approach surpasses FLUX-based methods such as StableFlow and RF-Edit by a large margin, demonstrating the effectiveness of our method despite the base model not having a clear advantage.

Qualitative Results. Visual quality is critical for evaluating image editing. Figure 6 presents qualitative comparisons across all PIE-Bench tasks. For space, we omit P2P and P2P-Zero, which show weaker background preservation and text alignment, respectively (see Table 1). Our method achieves a better trade-off between preserving unedited regions and accurately aligning edited regions with the target prompt. More visualizations are provided in Supplementary Material A.5.

User Study. Our method compares against two UNet-based and two transformer-based diffusion models, all showing competitive performance in Table 1. The study uses 140 images from the 'random class' in PIE-Bench [20], covering all editing types. Each of the 60 volunteers is randomly assigned 20 editing cases. For each case, they are shown a source image, a target text prompt, and five edited results (randomly ordered and anonymized). Volunteers selected the best result via a custom web interface, as shown in Supplementary Material Figure 10. Results in Supplementary Material Figure 5 (c) show 43.2% preferred our method—the highest among all approaches, confirming that it maintains strong subjective visual quality.



a (+silver) cat (+sculpture) sitting next to a mirror

Figure 7: Illustrations of ablating the Learnable Prompt and LoRA.

Runtime Comparison. We conduct a runtime comparison of our method and other methods on a single NVIDIA L20 GPU, measuring both inversion and editing time, as shown in Table 3. A key advantage of our method is efficient support for multiple edits on the same image, a common real-world scenario. Once the inversion for a given image is completed, subsequent edits can be performed within 3.64 seconds, which is over 7× faster than other methods on average, while the initial inversion time is only 4× longer than other methods on average. This design effectively front-loads the computational cost, making it ideal for iterative workflows.

Table 3: **Runtime comparison.** Time for n edits on an image equals Inversion $+ n \times$ Per-editing.

Method	Inversion (s)	Per-editing (s)
P2P[16]	14.40	10.28
MasaCtrl[4]	**5.19**	17.45
P2P-Zero[31]	13.31	62.29
NTI[29]	95.54	10.32
PnP-Inv[20]	8.32	9.54
NP[28]	9.00	10.37
StableFlow[2]	13.85	27.20
RF-Edit[53]	55.48	54.07
EditInfinity	107.06	**3.64**

5.3 Ablation Study

Ablation studies are performed on the 'random class' of PIE-Bench, covering all types of editing and allowing an efficient and unbiased evaluation.

Ablation on Learnable Prompt and LoRA. We design a precise image inversion by leveraging quantized tokens as exact supervision. It integrates the learnable prompt for textual rectification and a LoRA for style preservation. As shown in Figure 7, removing both components causes significant structural inconsistencies. The learnable prompt improves alignment with the target prompt but often shifts global style. Adding LoRA further restores stylistic consistency with the source image. However, prolonged training leads to overfitting, causing the model to ignore editing intents. To balance editability and fidelity, we stop training LoRA after 20 iterations, as shown in Figure 8.

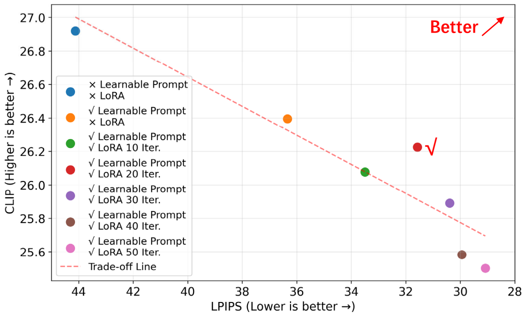


Figure 8: Quantitative results of ablating the Learnable Prompt and LoRA.

Ablation on Piecewise Linear Smoothing Kernel. We introduce $\mathcal{G}$ to ensure smooth transitions between edited and unedited regions and to suppress boundary artifacts. As shown in Figure 9 (c), removing $\mathcal{G}$ results in sharp discontinuities along object boundaries (e.g., the cat's ears), confirming its effectiveness in producing seamless edits. To further examine the choice of smoothing function, we compare the linear kernel defined in Equation 6 with a Gaussian kernel $(1 - e^{-d^2/2\alpha^2})$. With proper hyperparameter tuning, the linear kernel achieves superior results, as reported in Table 4. Complete results under both settings are provided in Supplementary Material (Tables 9 and 10).

Ablation on Mask. While our method defaults to user-provided masks, it can also leverage Infinity's cross-attention maps [16] for automatic mask generation without modifying the framework. Specifically, we automatically align differing words x between the source and target prompts. After completing inversion, we input the source or target prompt containing x into $\text{Infinity}(\cdot)$ and extract the cross-attention map corresponding to x. A threshold is then applied: values above the threshold are set to 0 (mask foreground), and others to 1 (background). Table 5 shows that our method

Table 4: Quantitative results of ablating the Piecewise Linear Smoothing Kernel.

$\mathcal{G}$	Background Preservation				Text Alignment		
	PSNR↑	LPIPS$_{10^3}$↓	MSE$_{10^4}$↓	SSIM$_{10^2}$↑	Whole↑	Edited↑	IR$_{10}$↑
✗	**31.12**	**24.47**	**13.03**	**93.53**	25.44	23.12	2.85
Gaussian kernel	28.15	32.91	24.40	92.17	26.10	23.81	4.61
Linear kernel	28.50	31.58	22.94	92.36	**26.22**	**23.99**	**5.39**

Table 5: **Quantitative results of ablating the mask.** EditInfinity-u denotes user-provided masks, while EditInfinity-c denotes masks automatically generated via cross-attention. Best and second-best results are shown in **bold** and *italics*, respectively.

Method	Base Model	Background Preservation				Text Alignment		
		PSNR↑	LPIPS$_{10^3}$↓	MSE$_{10^4}$↓	SSIM$_{10^2}$↑	Whole↑	Edited↑	IR$_{10}$↑
NTI[29]	U	*28.08*	57.94	36.10	85.17	24.71	22.51	3.63
RF-Edit[53]	T	27.26	92.27	*34.46*	86.67	24.65	22.03	0.61
EditInfinity-c	A	27.47	*44.97*	46.91	*90.30*	*25.71*	*23.22*	**5.40**
EditInfinity-u	A	**28.50**	**31.58**	**22.94**	**92.36**	**26.22**	**23.99**	*5.39*

is not highly sensitive to the source of the mask—strong performance is achieved in both cases. Comprehensive comparisons are reported in Supplementary Material Table 11.

Ablation on Multi-scale Autoregressive Editing. By autoregressively blending source and target tokens (AR), source tokens in un-edited regions effectively guide the generation of editing regions. As illustrated in Figure 9 (b), blending source tokens at the end of autoregressive generation (NAR, Non-Autoregressive) results in incoherent and visually inconsistent edits. Thus, incorporating guidance at every scale is essential for producing harmonious and realistic results. The quantitative comparison is represented in Supplementary Material Table 12.

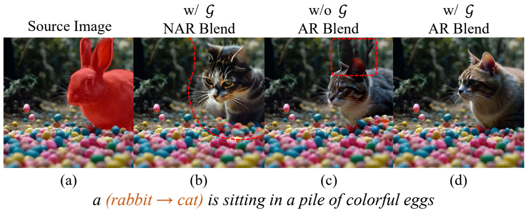


Figure 9: Illustrations of ablating the Piecewise Linear Smoothing Kernel and Multi-scale Autoregressive Editing.

6 Conclusion

We present *EditInfinity*, a parameter-efficient adaptation of binary-quantized generative models for text-driven image editing, following the classical 'inversion–editing' paradigm. During inversion, we formulate the process as an optimization problem supervised by the exact intermediate quantized representations. During editing, we propose a holistic smoothing strategy to blend source and target tokens, preserving unedited regions while aligning with target prompts. Experiments on PIE-Bench show that *EditInfinity* outperforms diffusion-based baselines.

7 Limitation

While our method demonstrates strong performance across diverse editing tasks, it shows limitations in extreme cases such as style change, where no background needs to be preserved, and the image contains detailed structural patterns. In such cases, the blending between source and target tokens is constrained, which may lead to suboptimal preservation of structural fidelity from the original image. Nonetheless, thanks to our image inversion strategy, which effectively learns the generative trajectory of source image, our method can still accomplish intended edits, despite slight structural degradation. In contrast, other methods often fail in such challenging scenarios. For example, as shown in Figure 6, row 9, while other methods are unable to convert the painted bird into a realistic one, our method successfully achieves the style change, with only minor deviation in the bird's head pose.

8 Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant NO. 62572145, 62176077, 62372133, 62125201 and U24B20174), in part by the Shenzhen Key Technical Project (Grant NO. JCYJ20241202123728037, JSGG20220831092805009, JSGG202208311056 03006 and KJZD20230923115117033), in part by the Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (Grant NO. 2022B1212010005) and in part by "Guangdong Special Support Plan" (Grant No. 2023TQ07A784).

References

[1] Yuval Alaluf, Daniel Garibi, Or Patashnik, Hadar Averbuch-Elor, and Daniel Cohen-Or. Cross-image attention for zero-shot appearance transfer. In *Proceedings of ACM Special Interest Group on Computer Graphics and Interactive Techniques Conference*, pages 1–12, 2024.

[2] Omri Avrahami, Or Patashnik, Ohad Fried, Egor Nemchinov, Kfir Aberman, Dani Lischinski, and Daniel Cohen-Or. Stable flow: Vital layers for training-free image editing. *arXiv preprint arXiv:2411.14430*, 2024.

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023.

[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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22560–22570, 2023.

[5] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[6] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[7] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2019.

[8] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in neural information processing systems*, 34:19822–19835, 2021.

[9] Wenkai Dong, Song Xue, Xiaoyue Duan, and Shumin Han. Prompt tuning inversion for text-driven image editing using diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7430–7440, 2023.

[10] Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. Renoise: Real image inversion through iterative noising. In *European Conference on Computer Vision*, pages 395–413. Springer, 2024.

[11] Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Houqiang Li, Han Hu, et al. Instructdiffusion: A generalist modeling interface for vision tasks. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 12709–12720, 2024.

[12] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023.

[13] Google. Gemini 2.0 flash preview image generation. `https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-0-flash`, 2024. Accessed: 2025-10-06.

[14] Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. *arXiv preprint arXiv:2412.04431*, 2024.

[15] Ligong Han, Song Wen, Qi Chen, Zhixing Zhang, Kunpeng Song, Mengwei Ren, Ruijiang Gao, Anastasis Stathopoulos, Xiaoxiao He, Yuxiao Chen, et al. Improving tuning-free real image editing with proximal guidance. *arXiv preprint arXiv:2306.05414*, 2023.

[16] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 1(2):3, 2022.

[18] Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, et al. Smartedit: Exploring complex instruction-based image editing with multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8362–8371, 2024.

[19] Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks. *arXiv preprint arXiv:2103.10427*, 2021.

[20] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Direct inversion: Boosting diffusion-based editing with 3 lines of code. *arXiv preprint arXiv:2310.01506*, 2023.

[21] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6007–6017, 2023.

[22] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[23] Shanglin Li, Bohan Zeng, Yutang Feng, Sicheng Gao, Xiuhui Liu, Jiaming Liu, Lin Li, Xu Tang, Yao Hu, Jianzhuang Liu, et al. Zone: Zero-shot instruction-guided local editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6254–6263, 2024.

[24] Yaowei Li, Yuxuan Bian, Xuan Ju, Zhaoyang Zhang, Ying Shan, Yuexian Zou, and Qiang Xu. Brushedit: All-in-one image inpainting and editing. *arXiv preprint arXiv:2412.10316*, 2024.

[25] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7817–7826, 2024.

[26] Shilin Lu, Yanzhu Liu, and Adams Wai-Kin Kong. Tf-icon: Diffusion-based training-free cross-domain image composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2294–2305, 2023.

[27] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

[28] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2063–2072. IEEE, 2025.

[29] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6038–6047, 2023.

[30] Yotam Nitzan, Zongze Wu, Richard Zhang, Eli Shechtman, Daniel Cohen-Or, Taesung Park, and Michaël Gharbi. Lazy diffusion transformer for interactive image editing. In *European Conference on Computer Vision*, pages 55–72. Springer, 2024.

[31] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *Proceedings of ACM Special Interest Group on Computer Graphics and Interactive Techniques Conference*, pages 1–11, 2023.

[32] Or Patashnik, Daniel Garibi, Idan Azuri, Hadar Averbuch-Elor, and Daniel Cohen-Or. Localizing object-level shape variations with text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 23051–23061, 2023.

[33] 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. *arXiv preprint arXiv:2307.01952*, 2023.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[35] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021.

[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[37] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. 2022.

[38] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

[39] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8871–8879, 2024.

[40] Xincheng Shuai, Henghui Ding, Xingjun Ma, Rongcheng Tu, Yu-Gang Jiang, and Dacheng Tao. A survey of multimodal-guided image editing with text-to-image diffusion models. *arXiv preprint arXiv:2406.14555*, 2024.

[41] Kihyuk Sohn, Nataniel Ruiz, Kimin Lee, Daniel Castro Chin, Irina Blok, Huiwen Chang, Jarred Barber, Lu Jiang, Glenn Entis, Yuanzhen Li, et al. Styledrop: Text-to-image generation in any style. *arXiv preprint arXiv:2306.00983*, 2023.

[42] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[43] Xue Song, Jiequan Cui, Hanwang Zhang, Jingjing Chen, Richang Hong, and Yu-Gang Jiang. Doubly abductive counterfactual inference for text-based image editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9162–9171, 2024.

[44] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

[45] Haotian Tang, Yecheng Wu, Shang Yang, Enze Xie, Junsong Chen, Junyu Chen, Zhuoyang Zhang, Han Cai, Yao Lu, and Song Han. Hart: Efficient visual generation with hybrid autoregressive transformer. *arXiv preprint arXiv:2410.10812*, 2024.

[46] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

[47] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023.

[48] Dani Valevski, Matan Kalman, Eyal Molad, Eyal Segalis, Yossi Matias, and Yaniv Leviathan. Unitune: Text-driven image editing by fine tuning a diffusion model on a single image. *ACM Transactions on Graphics*, 42(4):1–10, 2023.

[49] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

[50] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.

[51] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[53] Jiangshan Wang, Junfu Pu, Zhongang Qi, Jiayi Guo, Yue Ma, Nisha Huang, Yuxin Chen, Xiu Li, and Ying Shan. Taming rectified flow for inversion and editing. *arXiv preprint arXiv:2411.04746*, 2024.

[54] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[55] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: learning and evaluating human preferences for text-to-image generation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 15903–15935, 2023.

[56] Xiaofeng Yang, Cheng Chen, Xulei Yang, Fayao Liu, and Guosheng Lin. Text-to-image rectified flow as plug-and-play priors. *arXiv preprint arXiv:2406.03293*, 2024.

[57] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.

[58] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023.

[59] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023.

[60] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[61] Yue Zhao, Yuanjun Xiong, and Philipp Krähenbühl. Image and video tokenization with binary spherical quantization. *arXiv preprint arXiv:2406.07548*, 2024.

[62] Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. In *European Conference on Computer Vision*, pages 195–211. Springer, 2024.

[63] Xianwei Zhuang, Yuxin Xie, Yufan Deng, Dongchao Yang, Liming Liang, Jinghan Ru, Yuguo Yin, and Yuexian Zou. Vargpt-v1. 1: Improve visual autoregressive large unified model via iterative instruction tuning and reinforcement learning. *arXiv preprint arXiv:2504.02949*, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and the introduction clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. The abstract and introduction provide a concise overview of the research area, the novel approaches introduced, and the key contributions, ensuring that readers have a clear understanding of what the paper aims to achieve and the significance of its findings.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations of the work in detail and create a separate "Limitations" section in paper. See Section Discussion for details.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All formulas and proofs in the paper are numbered and cross-referenced. The paper provides complete proofs of the formulas, and for those appearing in the supplementary material, a brief proof sketch is provided in the main text. The proofs in the paper are rigorously reasoned, adhere to accepted mathematical principles, and do not omit any critical steps. The theorems and lemmas relied upon in the proofs are appropriately referenced and cross-referenced. See Section 3 and 4 for details.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have clearly demonstrated the complete algorithm in the paper using formulas and pseudocode, making it easy for readers to reproduce the results in the paper. At the same time, we are organizing and preparing to open source our code for readers to use. See Section 4 for details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: This paper does not currently provide open-access code, but it is planned to be made public in the future.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provided a detailed explanation of our experimental setup in the main paper, including the learnable prompt and lora training settings, as well as the hyperparameter settings during inference. See Section 5 for more details. We will attach more experimental settings in the supplementary materials.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The paper does not report error bars suitably and correctly defined, nor does it provide other appropriate information about the statistical significance of the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments in subsection 5.1 and supplementary materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: In the research conducted in our paper, we followed the NeurIPS Code of Ethics`https://neurips.cc/public/EthicsGuidelines` in all aspects.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss both potential positive societal impacts and negative societal impacts of the work performed in the supplementary material.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We will prevent the misuse of the model by requiring users to follow the usage guidelines. See supplementary materials for details.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and the license and terms of use explicitly mentioned and properly respected.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The code and documentation are currently not open source. We are organizing and preparing to open source the complete code and documentation together for readers to use.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: Our complete experimental instructions, task interface diagrams, and compensation details will be attached in the supplementary materials.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: We discussed the potential risks, informed consent process, and ethical review approval of human studies in the supplementary materials.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [No]

    Justification: The manuscript was proofread using LLM, which did not affect the research methodology.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Supplementary Material

The supplementary material is organized as follows:

- Subsection A.1 presents two applications of our method: facial attribute editing and complex-scene image editing.
- Subsection A.2 shows the supplementary implementation details of EditInfinity.
- Subsection A.3 presents more details of user study.
- Subsection A.4 provides more comprehensive ablation studies.
- Subsection A.5 exhibits additional qualitative results for supplementary.
- Subsection A.6 declares broader impacts of our proposed EditInfinity.
- Subsection A.7 declares safeguards of our proposed EditInfinity.
- Subsection A.8 states ethical considerations for EditInfinity.

A.1 Applications.

Facial Attribute Change. To verify that our method generalizes to *unmaskable* edits, where localized masks are impractical, we conduct experiments on facial attribute modification. Specifically, we randomly select 20 images from FFHQ to perform unmasked edits, including age, expression, skin tone. Since the setting is unmasked, there is no background to preserve, and thus standard metrics that rely on background consistency are not applicable. In addition to retaining the Whole metric (CLIP score between the entire edited image and the target prompt), we introduce ArcFace [7] for evaluating identity preservation, and CLIP-I for measuring similarity between the source and edited images. Table 6 shows that our method outperforms strong baselines, including the leading Diffusion UNet model NTI [29] and the Diffusion Transformer model RF-Edit [53]. Thanks to our proposed image inversion algorithm, which effectively learns the generative trajectory of the source image, our method can accomplish the intended edits.

Table 6: Quantitative results on facial images from FFHQ.

Method	Base Model	ArcFace↑	CLIP-I↑	Whole↑
NTI[29]	U	0.56	0.83	23.67
RF-Edit[53]	T	0.61	0.79	23.54
EditInfinity	A	**0.63**	**0.86**	**24.82**

Complex Scene Images Editing. Given that PIE-Bench already contains nearly 50% natural images, it serves as a comprehensive benchmark for evaluating our method on open-ended editing tasks. However, to further assess performance on *complex scenes* involving multiple interacting objects, we conduct an additional evaluation on complex scene images editing. We select 20 MagicBrush images (due to time constraints) filtered by GPT-4o, comprising five samples each with 2, 3, 4, and 5 primary objects. Table 7 demonstrates the superiority of our method in handling complex scenes, compared to the two strong baselines, i.e., NTI [29] and RF-Edit [53].

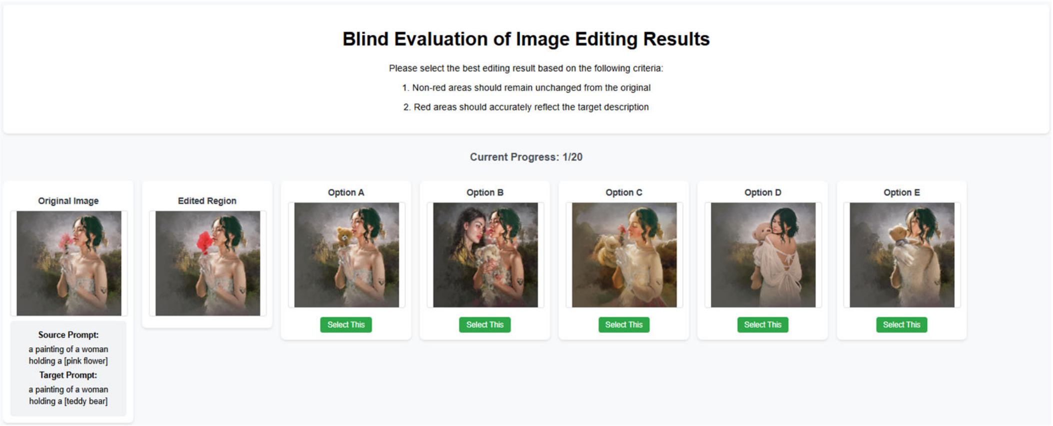
Table 7: Quantitative results on the complex scene images from MagicBrush.

Method	Base Model	Background Preservation				Text Alignment	
		PSNR↑	LPIPS$_{10^3}$↓	MSE$_{10^4}$↓	SSIM$_{10^2}$↑	Whole↑	Edited↑
NTI[29]	U	8.81	452.03	1380.48	39.63	19.90	16.83
RF-Edit[53]	T	26.00	121.84	33.98	84.73	24.13	18.29
EditInfinity	A	**31.23**	**24.30**	**9.90**	**91.70**	**24.19**	**20.07**

A.2 Supplementary Implementation Details.

During image inversion, we set the learning rate to 4.6875e-5 and use AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.97$) for both the learnable prompt and LoRA training. The two components are optimized sequentially, starting with the learnable prompt, followed by LoRA. To accelerate the convergence of training LoRA, a KL-divergence loss is introduced in addition to the standard cross-entropy loss. Typically, the learnable prompt is trained for 10 iterations, while LoRA is trained for 20 iterations. These settings may be adapted according to the specific editing scenario to optimal performance.

A.3 User Study Details.



Blind Evaluation of Image Editing Results

Please select the best editing result based on the following criteria:

1. Non-red areas should remain unchanged from the original
2. Red areas should accurately reflect the target description

Current Progress: 1/20

Source Prompt:
a painting of a woman holding a [pink flower]

Target Prompt:
a painting of a woman holding a [teddy bear]

Figure 10: Custom web interface of user study.

Each volunteers is asked to select the best editing result via a custom web interface specifically developed for this evaluation, as shown in Figure 10. The interface presents a source image along with its corresponding prompt, the edited region, a target prompt, and five edited results. The methods behind these results are anonymized and displayed in a randomized order for each evaluation.

A.4 More Ablation Study.

Ablation on Transformer LoRA. As shown in Table 8, applying LoRA solely to FFN layers yields a more favorable trade-off between background preservation and text alignment compared to other configurations. Therefore, we adopt this configuration in our final design, enabling effective editing with minimal additional parameter overhead.

Table 8: Ablation on Transformer LoRA. Attn denotes both self-attention and cross-attention.

FFN	Attn	Background Preservation				Text Alignment		
		PSNR↑	LPIPS$_{10^3}$↓	MSE$_{10^4}$↓	SSIM$_{10^2}$↑	Whole↑	Edited↑	IR$_{\times 10}$↑
✓	✗	28.50	31.58	22.94	92.36	**26.22**	**23.99**	**5.39**
✗	✓	28.50	31.11	23.07	92.32	25.72	23.34	3.93
✓	✓	**28.81**	**30.31**	**21.39**	**92.64**	25.61	23.29	4.23

Ablation on Piecewise Linear Smoothing Kernel. Table 2 in the main body only presents better balance results for Gaussian and linear kernels. The full results under varying hyperparameter configurations of Gaussian and linear kernels are provided in Tables 9 and 10, respectively. In the case of Gaussian kernel, increasing α enlarges the smooth transition zone, compromising background retention while improving text alignment. In the case of the linear kernel, when fixing τ_1 and gradually increasing τ_2, the transition zone width of the linear kernel ($\tau_2 - \tau_1$) increases accordingly. This leads to improved text alignment metrics but at the cost of degraded background preservation performance. Conversely, when τ_2 is fixed and τ_1 increases, both text alignment and background preservation tend to deteriorate. These observations indicate that τ_1 and τ_2 play a critical role in balancing edit fidelity and content preservation. Overall, the linear kernel setting of $\tau_1 = 1$ and $\tau_2 = 4$ offers a better trade-off, achieving strong text alignment (e.g., IR$_{10}$ = 5.39) while keeping background distortion (e.g., LPIPS = 31.58) within acceptable limits.

Table 9: Quantitative results of ablating the Gaussian kernel.

α	Background Preservation				Text Alignment		
	PSNR↑	LPIPS$_{10^3}$↓	MSE$_{10^4}$↓	SSIM$_{10^2}$↑	Whole↑	Edited↑	IR$_{10}$↑
1	**29.40**	**28.59**	**18.45**	**93.01**	26.09	23.79	4.74
2	28.63	31.16	21.71	92.45	26.08	23.71	4.66
3	28.15	32.91	24.40	92.17	26.10	**23.81**	4.61
4	28.08	33.05	25.00	92.25	**26.23**	23.73	**4.91**

Table 10: Quantitative results of ablating the linear kernel.

τ_1	τ_2	Background Preservation				Text Alignment		
		PSNR↑	LPIPS$_{10^3}$↓	MSE$_{10^4}$↓	SSIM$_{10^2}$↑	Whole↑	Edited↑	IR$_{10}$↑
0	1	**29.19**	**29.15**	**19.47**	**92.87**	26.13	23.74	4.83
0	2	29.16	29.23	19.48	92.86	26.11	23.86	5.04
0	3	28.92	29.87	20.35	92.79	**26.24**	23.85	5.00
0	4	28.69	30.60	21.38	92.66	26.19	**24.01**	4.79
0	5	28.45	31.68	22.86	92.51	26.20	23.80	4.71
1	2	28.80	30.21	22.34	92.65	26.07	23.71	4.72
1	3	28.46	31.75	22.54	92.35	26.13	23.83	4.85
1	4	28.50	31.58	22.94	92.36	26.22	23.99	**5.39**
1	5	28.41	31.74	22.98	92.36	26.14	23.83	4.73
2	3	28.50	31.44	22.46	92.50	26.18	23.53	4.96
2	4	28.14	32.69	24.47	92.31	26.17	23.72	4.91
2	5	27.53	36.17	29.65	91.79	26.21	23.74	5.12

Table 11: **Quantitative results of ablating the mask.** EditInfinity-u denotes user-provided masks; EditInfinity-c denotes cross-attention masks. Best and second-best results are shown in **bold** and *italics*.

Method	Base Model	Background Preservation				Text Alignment		
		PSNR↑	LPIPS$_{10^3}$↓	MSE$_{10^4}$↓	SSIM$_{10^2}$↑	Whole↑	Edited↑	IR$_{10}$↑
P2P[16]	U	18.81	197.11	197.69	73.68	25.10	22.98	0.29
MasaCtrl[4]		23.36	95.45	77.63	81.88	23.30	20.92	-3.82
P2P-Zero[31]		20.92	161.28	137.64	77.02	22.89	21.09	-5.71
NTI[29]		*28.08*	57.94	36.10	85.17	24.71	22.51	3.63
PnP-Inv[20]		23.60	103.12	72.77	81.11	25.05	22.94	3.34
NP[28]		27.24	62.40	37.79	84.92	24.89	22.67	2.92
StableFlow[2]	T	23.68	72.77	78.61	88.11	23.17	21.21	0.76
RF-Edit[53]		27.26	92.27	*34.46*	86.67	24.65	22.03	0.61
EditInfinity-c	A	27.47	*44.97*	46.91	*90.30*	*25.71*	*23.22*	**5.40**
EditInfinity-u		**28.50**	**31.58**	**22.94**	**92.36**	**26.22**	**23.99**	*5.39*

Ablation on Mask. Our method assumes the user provides masks. Indeed, this is a well-established task setting in image editing [37, 59], especially when text alone is insufficient for the precise localization of the user-desired editing region. This challenge of accurately conveying user intent has long been recognized in controllable image generation. To enhance controllability, ControlNet [59] leverages visual priors such as edge maps, while DreamBooth [37] utilizes user-provided images to capture detailed features not easily conveyed by text.

While our method assumes user-provided masks by default, it can also leverage Infinity's cross-attention maps [16] for automatic mask generation without modifying the framework. Table 11 reports comprehensive comparisons and shows that our method is not highly sensitive to the source of the mask—strong performance is achieved in both cases.

Ablation on Multi-scale Autoregressive Editing. By blending source tokens at each scale in an autoregressive (AR) manner, our method provides continuous guidance for editing region generation at subsequent scales. In contrast, the non-autoregressive (NAR) approach blends source tokens only at the end of each scale, without influencing the token generation process at the next scale. This leads to incoherent transitions and visually inconsistent edits, as illustrated in Figure 11. Table 12 further supports this observation: AR consistently outperforms NAR in both background preservation and text alignment. These results highlight the necessity of autoregressive guidance for achieving harmonious and realistic edits.

Table 12: Quantitative results of multi-scale autoregressive editing.

Blend	Background Preservation				Text Alignment		
	PSNR↑	LPIPS$_{10^3}$↓	MSE$_{10^4}$↓	SSIM$_{10^2}$↑	Whole↑	Edited↑	IR$_{10}$↑
NAR	25.50	42.59	38.39	91.00	25.98	23.64	3.54
AR	**28.50**	**31.58**	**22.94**	**92.36**	**26.22**	**23.99**	**5.39**

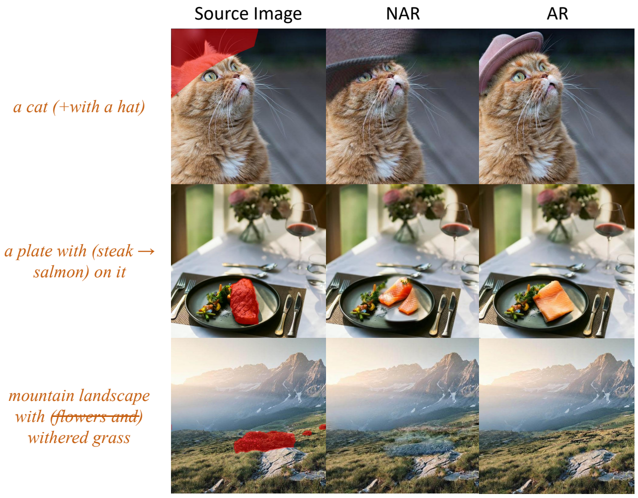

Source Image
NAR
AR
a cat (+with a hat)
a plate with (steak → salmon) on it
mountain landscape with ~~(flowers and)~~ withered grass

Figure 11: Illustrations of ablating the Multi-scale Autoregressive Editing.

A.5 Additional Qualitative Results.

We present additional qualitative results to further demonstrate the effectiveness of our method, as shown in Figure 12 and 13. These results include diverse editing types across add object, change object, delete object, change content, change pose, change color, change background, change material, and change style. We also provide comparisons with state-of-the-art methods, highlighting our model's ability to preserve background details and align with target prompts in image editing.

A.6 Broader Impacts

Our proposed method enables high-quality image editing. Positive societal impacts include its potential applications in education (e.g., visual content adaptation for learning) and creative industries (e.g., graphic design and media production). However, potential negative societal impacts include misuse for deceptive content creation (e.g., deepfakes or misinformation). We acknowledge the dual-use nature of image generation technologies and emphasize responsible deployment.

A.7 Safeguards

To mitigate risks associated with misuse, we adopt the following safeguards:

- We will release the model under a research-use-only license.
- Model checkpoints and code will include a usage agreement that prohibits harmful or deceptive use cases (e.g., unauthorized alteration of real people's images).
- All datasets used for editing are publicly available and contain no private or personally identifiable information.

A.8 Ethical Considerations

There is no potential risks incurred by study participants in this paper. As such, Institutional review board (IRB) approval was not required.

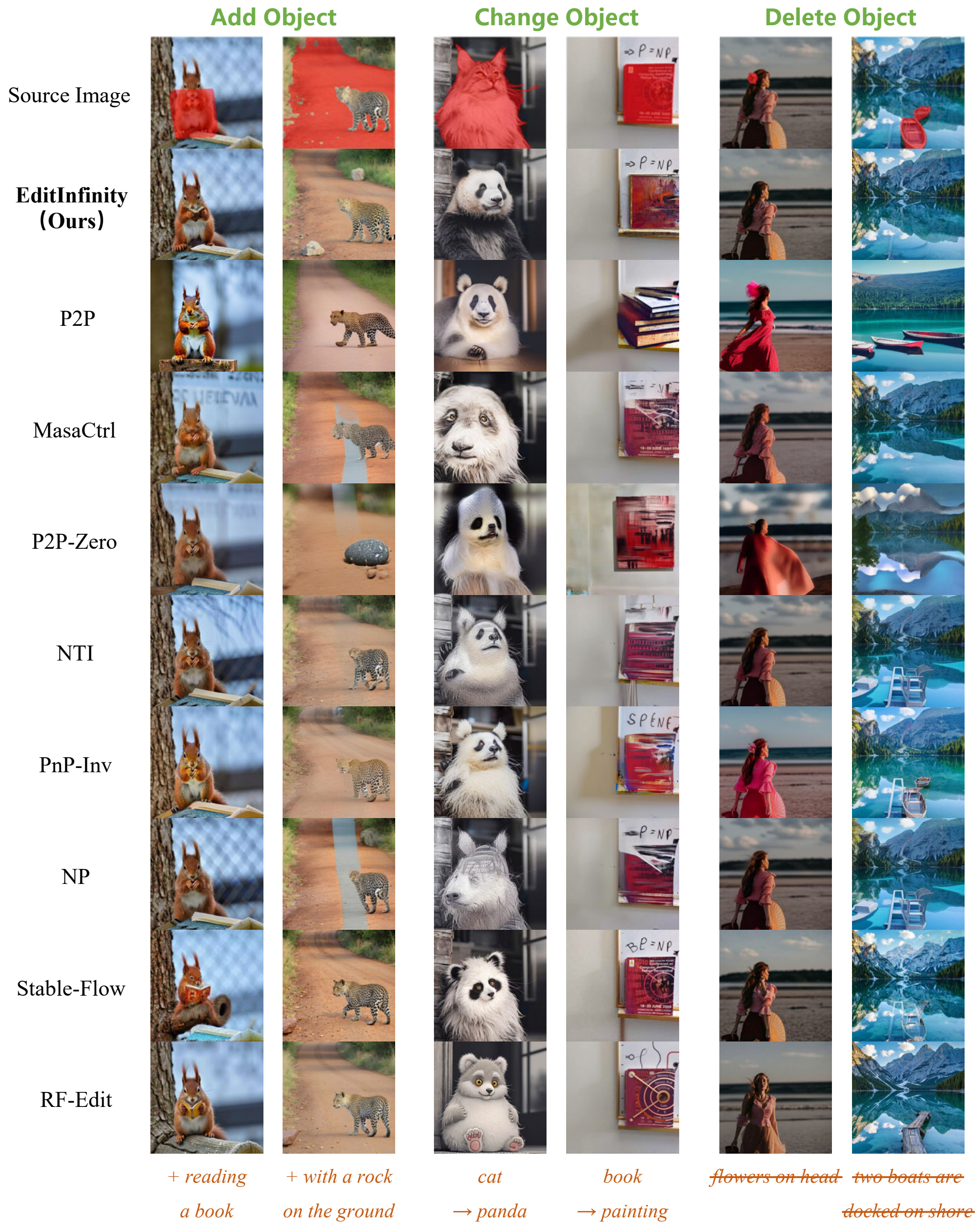


Figure 12: **Qualitative results on PIE-Bench across add, change, and delete object.** The red mask denotes the edited region M, expected to follow the prompt, while other regions retain the background.

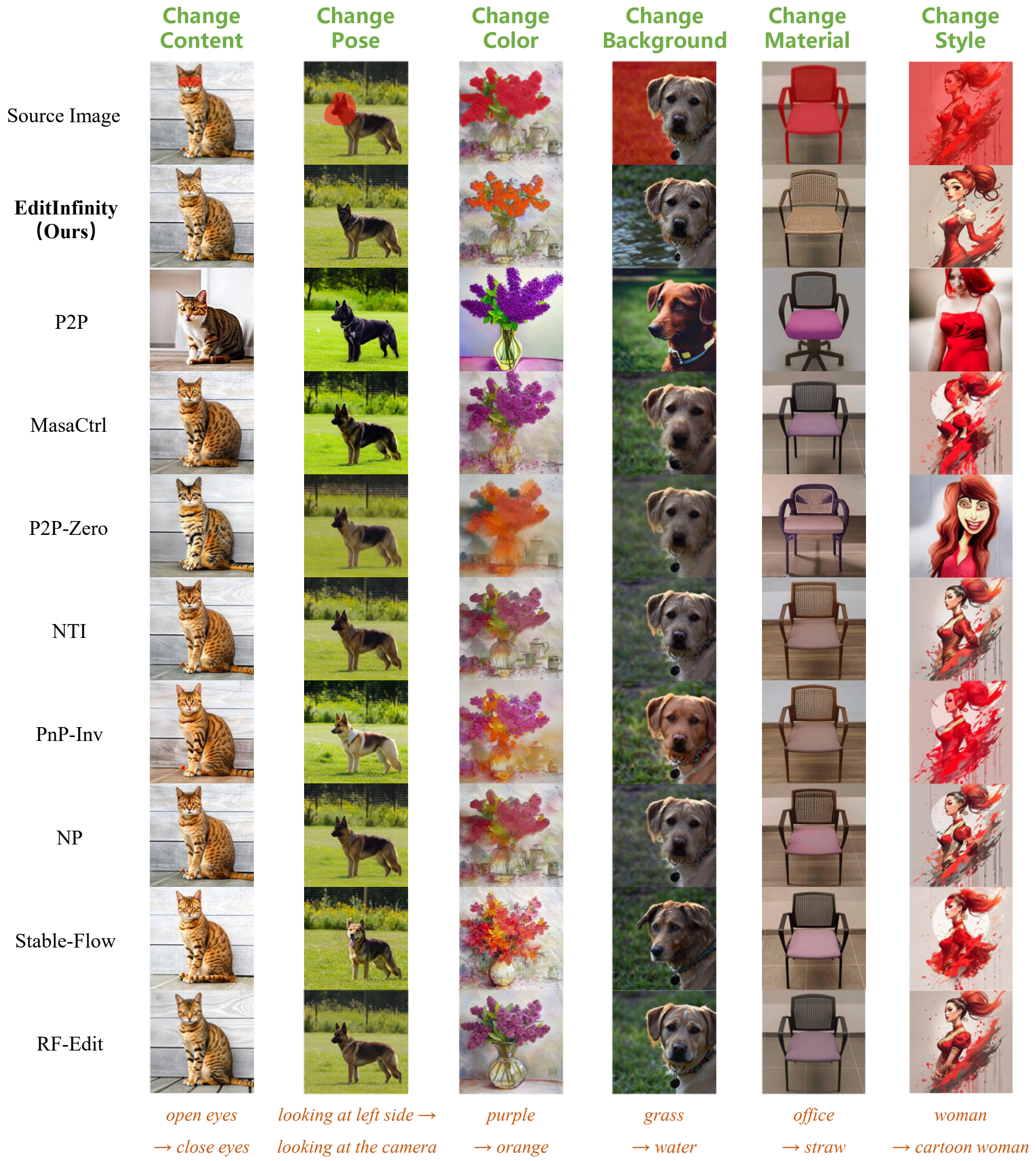


Figure 13: **Qualitative results on PIE-Bench across change content, change pose, change color, change background, change material, and change style.** The red mask denotes the edited region M, expected to follow the prompt, while other regions retain the background.