ProEdit: Inversion-based Editing From Prompts Done Right

Zhi Ouyang[1,*] **Dian Zheng**[2,*] **Xiao-Ming Wu**[3] **Jian-Jian Jiang**[1]
Kun-Yu Lin[4] Jingke Meng[1,5,✉] Wei-Shi Zheng[1,6,✉]

[1]Sun Yat-sen University [2]CUHK MMLab
[3]College of Computing and Data Science, Nanyang Technological University
[4]The University of Hong Kong
[5]Key Laboratory of Tibetan Information Processing, Ministry of Education, Xining
[6]Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
https://isee-laboratory.github.io/ProEdit/

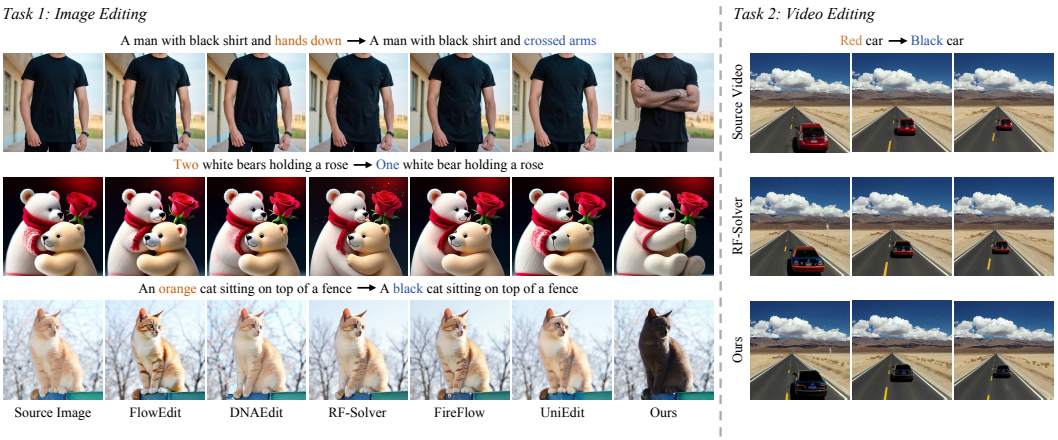

Figure 1: **ProEdit for image and video editing**. We propose a highly accurate, plug-and-play editing method for flow inversion that addresses the problem of excessive source image information injection, which achieves proper modification of attributes such as pose, number, and color. Our method has demonstrated impressive performance in both image editing and video editing tasks.

Abstract

Inversion-based visual editing provides an effective and training-free way to edit an image or a video based on user instructions. Existing methods typically inject source image information during the sampling process to maintain editing consistency. However, this sampling strategy overly relies on source information, which negatively affects the edits in the target image (e.g., failing to change the subject's attributes like pose, number, or color as instructed). In this work, we propose **ProEdit** to address this issue both in the attention and latent aspects. In the attention aspect, we introduce KV-mix, which mixes KV features of the source and the target in the edited region, mitigating the influence of the source image on the editing region while maintaining background consistency. In the latent aspect, we propose Latents-Shift, which shifts the distribution of the edited region in the inverted noise, eliminating the negative influence of the inverted noise on the sampling process. Extensive experiments on several image and video editing

* equal contributions.
✉ corresponding authors. Code is available

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

benchmarks demonstrate that our method achieves SOTA performance. In addition, our design is plug-and-play, which can be seamlessly integrated into existing inversion and editing methods, such as RF-Solver, FireFlow and UniEdit.

1 Introduction

Inversion-based visual editing [1, 2, 3, 4, 5, 6, 7] has emerged as a highly effective and valuable research direction. Leveraging powerful generative models, it offers a training-free paradigm for modifying images and videos according to user instructions, especially the flow-based editing methods, which provides better generative abilities with fewer sampling steps.

Most established inversion-based methods [8, 9, 10] start from the source image to perform inversion and obtain the inverted noise. They then use this inverted noise as the initial point and execute a new denoising sampling process guided by the target prompt to generate the edited image or video. To preserve fidelity to the original content, these methods typically employ a source attention injection strategy to re-introduce source attention information during the sampling steps. However, as shown in Figure 1, this approach relies too heavily on source information across both latent representations and attention maps. This over-reliance negatively impacts the target image generation, particularly restricting the editing of subject attributes such as color, pose, and object quantity.

In this work, we first investigate the above problems and conclude that completely relying on the inverted latents for sampling and applying global attention feature injection introduces excessive source image information, leading to editing failure. Specifically, in the attention aspect, we find that the global attention feature injection strategy introduces excessive attribute-related information from the source attention, causing the model to overly focus on source information while neglecting text guidance. For the latent aspect, starting from the source image distribution creates an overly strong prior that easily leads the sampling process to reconstruct the source distribution. We systematically validate it in Section 3.2.

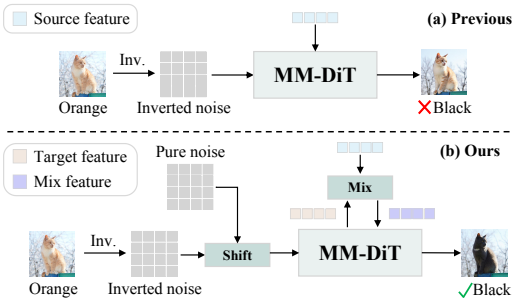

Figure 2: Framework comparison between (a) previous methods and (b) our method. To address the issue of excessive source image information injection, we introduce the Shift module for inverted noise distribution and the Mix module for the attention injection, alleviating the attribute editing failures caused by these issues.

Based on these observations, we propose a novel inversion-based editing method, **ProEdit**, designed to eliminate the negative impact of source image constraints from both attention and latent perspectives. From the attention perspective, we introduce **KV-mix**. This mechanism first identifies the edited regions based on the differences between the source and target prompts. It then mixes the Key (K) and Value (V) features specifically within these edited areas. Conversely, in non-edited regions, the source KV features are fully injected to preserve background consistency. From the latent perspective, we propose **Latents-Shift**. Drawing inspiration from the structure-preserving distribution transformations of AdaIN [11] in style transfer, we only apply Gaussian noise to shift the distribution of the inverted noise within the edited regions. This operation effectively diminishes the influence of original subject attributes while strictly maintaining overall structural and background consistency. As illustrated in Figure 2, our method successfully overcomes the restrictive effects of source information inherent in standard inversion noise and attention injection mechanisms. ProEdit accurately modifies the target subject's attributes while preserving the integrity of the background and unedited content.

Through extensive experiments, we demonstrate that 1) **Effectiveness**: ProEdit effectively eliminates the negative impact of the source image or video on the edited content, achieving an excellent balance between fidelity and editability, demonstrating best performance; 2) **Plug-and-play**: Our method is plug-and-play, enabling its seamless integration into a wide range of existing inversion-based editing methods; 3) **Attribute Correction**: In attribute editing, where existing methods perform poorly, our approach showcases superior performance. Our contributions can be summarized as follows:

- We investigate the issue of excessive source image information injection in inversion-based editing and identify that this stems from both latent initialization and attention injection mechanisms.
- We propose ProEdit, a novel training-free approach designed to solve the above problem, which can eliminate the negative impact from the source while maintaining background and structural consistency, successfully addressing the failure cases of previous methods.
- Through extensive experiments, we prove that ProEdit is effective, plug-and-play and can be used in various types of image and video editing. Our code will be open-source to boost the developing of the generative community.

2 Related Work

2.1 Text-to-Visual generation

Earlier text-to-image generation primarily leveraged GANs [12, 13, 14, 15, 16] for high-fidelity synthesis. Autoregressive models [17, 18, 19] enabled image generation from open-domain textual descriptions. However, the autoregressive nature leads to slow generation processes. While diffusion models have recently achieved remarkable success and catalyzed numerous foundational frameworks [20, 21, 22, 23, 24], the field is currently witnessing a paradigm shift. Recent research indicates that text-guided image and video generation is transitioning from diffusion models using U-Net [25] architecture toward flow-based models employing MMDiT [26, 23] structures. Flow-based models, such as FLUX [27] and HunyuanVideo [28], utilize probability flow ordinary differential equations (ODEs) to simulate straight-line trajectories between distributions, thereby facilitating superior generation quality with significantly fewer sampling steps. Furthermore, these T2I and T2V advancements streamline image and video editing, enabling the synthesis of target content guided by source visuals and modified textual prompts.

2.2 Text-driven Editing

For visual editing tasks, early works focused on training-based methods [29, 30, 31, 32, 33, 34, 35, 36]. These methods leverage generative models to achieve controllable image editing. Recent focus has shifted towards more flexible and efficient training-free methods. Notably, inversion-based approaches, pioneered by DDIM inversion [37, 21], have inspired various high-precision solvers [38, 39, 40] to minimize inversion errors. Concurrently, sampling-based methods [41, 42, 7, 43, 44, 45] introduce controlled randomness, while attention-based methods [46, 47, 48, 6, 49, 50, 51, 52] modify attention mechanisms for controllable image and video editing.

In flow-based models, recent inversion methods employ advanced solvers [8, 9, 10, 53] and latent anchoring techniques [54, 10] to mitigate error accumulation during the inversion process. Under the MM-DiT [26] framework, attention-based editing approaches aim to achieve background and structural consistency through the use of attention head routing [55] or global attention feature injection [8, 9]. Despite their success in preserving background consistencys, they often struggle to perfectly align with complex editing instructions. Adding noise during sampling [56] improves editing flexibility but disrupts structural consistency. In this work, we attribute these limitations primarily to global attention injection [8, 9] and inverted noise [56], and introduce ProEdit to tackle both challenges concurrently. Our KV-mix introduces a spatial mixing mechanism of attention features to address the suppression caused by directly replacing source image attention on the editing process. Meanwhile, Latents-Shift applies a global shift to the inverted noise distribution, overcoming source distribution suppression while preserving structural consistency.

3 Method

In this section, we first introduce the preliminary to understand our method in Section 3.1, then conduct an investigation in Section 3.2, analyzing the reasons why the inversion-sampling paradigm faces challenges in removing the influence of the source image on the target image's edited contents. Next, we introduce our proposed ProEdit method in Section 3.3 and Section 3.4. Our method eliminates the negative influence of the source image on the target image's edited contents from both the attention guidance and initial latents in the sampling process, while maintaining the consistency of the background structure. Finally, we summarize our method's editing process in Section 3.5.

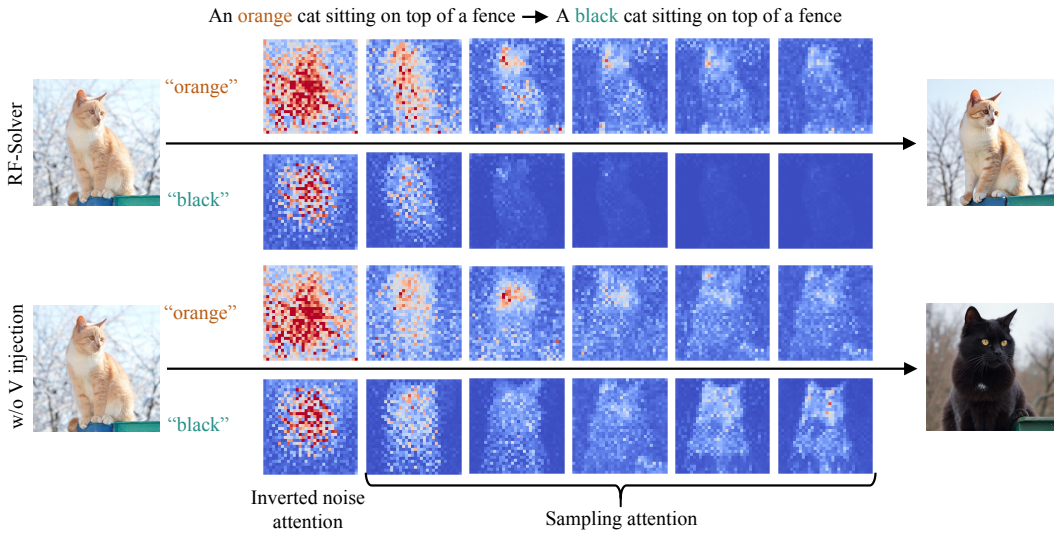

Figure 3: **Excessive source image information injection phenomenon in RF-Solver**. We validate it by visualizing the attention from source and target text tokens to the visual tokens during initial and sampling stage. In RF-Solver, the attention from the source text token to the visual tokens remains higher than that from the target text token. However, after removing attention injection, the attention from "black" and "orange" to visual tokens returns to similar levels, but some subject attributes (e.g., pose) change accordingly.

3.1 Preliminaries

Generative models [20] aim to generate data $\boldsymbol{X}_1$ that follows the real data distribution π_1 from noise $\boldsymbol{X}_0$ that follows a Gaussian distribution π_0. Recently, flow matching [57, 58, 59] has emerged as a method that learns a velocity field $\boldsymbol{v}_\theta$ to transform noise into data along a straight trajectory. The training objective is to solve the following optimization problem:

$$\begin{aligned} \min_{\theta} \mathbb{E}_{\boldsymbol{Z}_0,\boldsymbol{Z}_1,t}\left[\|(\boldsymbol{Z}_1-\boldsymbol{Z}_0)-\boldsymbol{v}_\theta(\boldsymbol{Z}_t,t)\|^2\right], \\ \boldsymbol{Z}_t = t\boldsymbol{Z}_1+(1-t)\boldsymbol{Z}_0,\ t\in[0,1], \end{aligned} \tag{1}$$

where $\boldsymbol{Z}_0 \sim \pi_0$ is initialized from the source distribution, and $\boldsymbol{Z}_1 \sim \pi_1$ is generated at the end of the trajectory. The term $\boldsymbol{Z}_1 - \boldsymbol{Z}_0$ represents the target velocity. The model learns a velocity field to deterministically transform random samples of Gaussian noise into target data via an Ordinary Differential Equation (ODE) defined over the continuous time interval:

$$d\boldsymbol{Z}_t = \boldsymbol{v}_\theta(\boldsymbol{Z}_t,t)dt, \quad t\in[0,1] \tag{2}$$

This ODE can be discretized and numerically solved by solvers as follows:

$$\boldsymbol{Z}_{t_{i+1}} = \boldsymbol{Z}_{t_i} + (t_{i+1}-t_i)v_\theta(\boldsymbol{Z}_{t_i},t_i), \tag{3}$$

where $i \in \{0,\dots,N\}$, with $t_0 = 0$ and $t_N = 1$.

Flow matching has deterministic trajectories. Its reverse process is obtained by reversing the learned flow trajectory. Starting from $\boldsymbol{Z}_1 \sim \pi_1$, the reverse ODE is given by reverse the velocity field:

$$d\boldsymbol{Z}_t = -\boldsymbol{v}_\theta(\boldsymbol{Z}_t,t)dt, \quad t\in[1,0] \tag{4}$$

Correspondingly, this ODE is discretized and solved using a numerical solver:

$$\boldsymbol{Z}_{t_{i-1}} = \boldsymbol{Z}_{t_i} - (t_{i-1}-t_i)v_\theta(\boldsymbol{Z}_{t_i},t_i), \tag{5}$$

where $i \in \{N,\dots,0\}$, with $t_N = 1$ and $t_0 = 0$. This inverse process generates $\boldsymbol{Z}_0 \sim \pi_0$ by utilizing the symmetry of the velocity field v to ensure consistency with the forward process. Naturally, this inversion method is applied in visual reconstruction and visual editing.

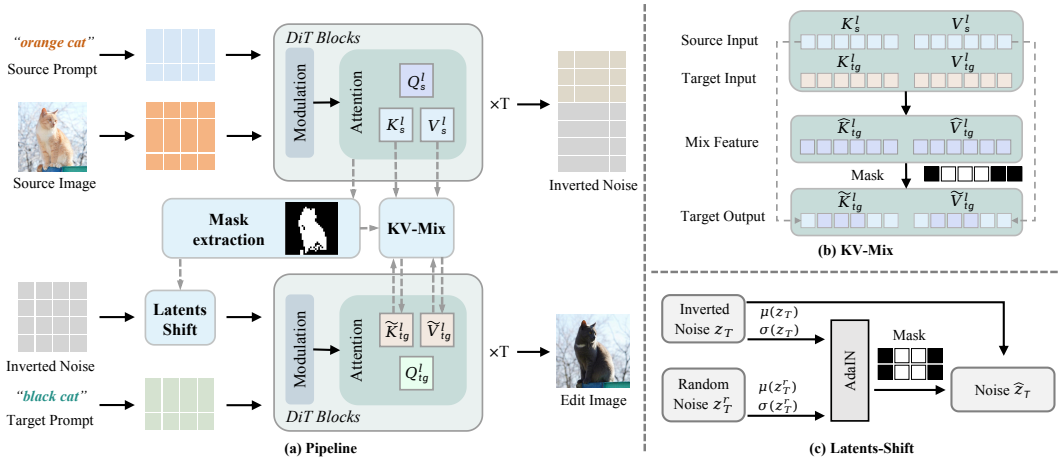

Figure 4: **Pipeline of our ProEdit.** The mask extraction module identifies the edited region based on source and target prompts during the first inversion step. After obtaining the inverted noise, we apply Latents-Shift to controllable perturb the initial distribution in the edited region, reducing source image information. In selected sampling steps, we fuse source and target attention features in the edited region while directly injecting source features in non-edited regions to achieve accurate attribute editing and background preservation simultaneously.

3.2 Rethinking the Inversion-Sampling Paradigm

In this subsection, we investigate the limitations of existing inversion-based editing methods. We observe that these approaches predominantly rely on sampling from inverted noise and utilizing source attention injection to preserve background and structural consistency. However, this design often forces the model to incorporate excessive information from the source image, which restricts necessary modifications and ultimately leads to editing failures. A detailed analysis of this phenomenon is provided below.

Attention Injection Problem. To maintain the overall structural consistency between the target and source images, current methods [8, 9] globally inject the value attention features V into specific time steps during the sampling process, as described by the following equation:

$$z_{tg}^{t}(l+1) = \text{Attn}(\boldsymbol{Q}_{tg}^{t}(l), \boldsymbol{K}_{tg}^{t}(l), \boldsymbol{V}_{s}^{t}(l)), \tag{6}$$

where s denotes the attention features corresponding to the source prompt, tg denotes the attention features corresponding to the target prompt, t represents the denoising time step, and l denotes the corresponding model layer. However, this global attention injection mechanism inadvertently hinders the editing process. As illustrated in Figure 3, while the injection mechanism successfully enhances overall structural consistency, it forces the model to prioritize the original attributes over the target prompt. For example, the model focuses significantly less on the target prompt "black" and remains strongly conditioned by the original "orange" features. This indicates that the source subject's attributes are being rigidly transferred into the target image, substantially increasing the difficulty of performing accurate attribute editing.

Latent Distribution Injection Problem. As shown in Figure 3, the attention from "orange" to visual tokens is significantly higher than from "black", indicating that although the image is inverted back to noise, it still retains substantial source image attributes. Consequently, when there is a large semantic gap between the target and source prompts, this persistent source information overwhelms the target guidance, leading to editing failures.

Summary. The negative impact of the source image on the editing process can be attributed to two factors: global attention feature injection and the latent distribution injection. Therefore, this paper proposes ProEdit to address all these issues from corresponding aspects.

3.3 KV-mix

Motivation. As analyzed earlier, previous methods use a global injection mechanism of visual and textual attention features to maintain consistency, but the excessive injection of source attention

features negatively impacts the editing quality. Towards this end, we propose an improved injection mechanism for attention features. Our method aims to mitigate the excessive injection problem by mixing source and target visual attention, which helps align with the target prompt while maintaining consistency in non-edited content.

Method. Based on the above observations, we apply attention control on the visual components across all blocks, while consistently using the attention features of the target prompt for text attention to achieve effective editing guidance. To distinguish between the editing and non-editing regions, we obtain a binary mask $\boldsymbol{M}$ by processing the attention map to separate the editing region, for the detailed implementation please see the Appendix A. For the non-editing region, we fully inject visual attention features to maintain background consistency. For the editing region, we use a mix of source and target visual attention features to preserve the consistency of non-editing content and improve the editing quality. We restrict the mixing operation to the $\boldsymbol{K}$ and $\boldsymbol{V}$ components, deliberately excluding $\boldsymbol{Q}$ to maintain editing flexibility. In the attention mechanism, Q dictates the model's "intent" and spatial focus, mixing $\boldsymbol{Q}$ would rigidly anchor the generation to the source image's structure and content, hindering the model's ability to adapt to new semantic prompts. In the Appendix D, we conduct ablation studies on different combinations of attention features, demonstrating that mixing $\boldsymbol{Q}$ has a detrimental effect on editing flexibility. Formally, our KV-mix design is as follows:

$$\begin{aligned}
(\hat{\boldsymbol{K}}_{tg}^t(l), \hat{\boldsymbol{V}}_{tg}^t(l)) &= \delta(\boldsymbol{K}_{tg}^t(l), \boldsymbol{V}_{tg}^t(l)) + (1-\delta)(\boldsymbol{K}_{s}^t(l), \boldsymbol{V}_{s}^t(l)), \\
(\tilde{\boldsymbol{K}}_{tg}^t(l), \tilde{\boldsymbol{V}}_{tg}^t(l)) &= \boldsymbol{M} \odot (\hat{\boldsymbol{K}}_{tg}^t(l), \hat{\boldsymbol{V}}_{tg}^t(l)) + (1-\boldsymbol{M}) \odot (\boldsymbol{K}_{s}^t(l), \boldsymbol{V}_{s}^t(l)), \\
z^t(l+1) &= \text{Attn}\left(\boldsymbol{Q}_{tg}^t(l), \tilde{\boldsymbol{K}}_{tg}^t(l), \tilde{\boldsymbol{V}}_{tg}^t(l)\right),
\end{aligned} \tag{7}$$

where $\boldsymbol{M}$ denotes the edited region that is extracted from attention map and applied only to the visual branch. To enable controllable editing strength and preservation of non-edited content, we define the **mixing strength** δ as a ratio of mix for applying attention control in the edited region, which determines the level of non-edited content preservation during editing. In the experiments on different hyperparameter settings in Appendix G, we found that setting δ to a high value, which mixes only a small proportion of the source image attention features, results in excellent editing performance. We attribute this to the fact that only a small proportion of source attention features is needed as anchors, which effectively maintains the consistency of non-editing attributes and structures, while not hindering the editing of target attributes.

This attention mechanism enables precise text control for consistent editing. Since we perform the KV-mix operation only within visual tokens, it is applied in both Double and Single Attention blocks to better align with text prompts.

3.4 Latents-Shift

Motivation. Here we aim to mitigate the problem of distribution injection while preserving the structure consistency. Previous methods [56] have attempted to further initialize the inverted noise with Gaussian noise to enhance its editability. However, this approach struggles to maintain the structural consistency of the edited regions, thereby failing to achieve precise attribute editing. Inspired by AdaIN [11] in style transfer, which transfers color and texture distributions while preserving structural consistency, we adapt this approach to image editing.

Method. As our goal is to eliminate the influence of source image information, we directly use random noise as the style image to shift the distribution of the inverted noise. We improve its formula to implement the shift of the latent distribution for the editing region as follows:

$$\begin{aligned}
\tilde{z}_T &= \sigma(z_T^r)\left(\frac{z_T - \mu(z_T)}{\sigma(z_T)}\right) + \mu(z_T^r), \\
\hat{z}_T &= \boldsymbol{M} \odot (\beta \tilde{z}_T + (1-\beta) z_T) + (1-\boldsymbol{M}) \odot z_T,
\end{aligned} \tag{8}$$

where β denotes the fusion ratio between the inverted noise and pure noise, controlling the level of shift in the inverted noise distribution. $\boldsymbol{M}$ denotes the edited region, which is inherited from KV-mix to achieve the shifted inverted noise distribution for the editing region. For the fusion ratio β, we believe that an overly large fusion ratio would cause the source-image distribution information to be completely lost, while an overly small fusion ratio would be insufficient to suppress the negative effects brought by the source distribution. Therefore, we suggest setting the fusion ratio β to a

Table 1: **Text-driven image editing comparison** on PIE-Bench. We report the results of each baseline, and evaluate our proposed method using flow-based inversion solvers RF-Solver, FireFlow, and UniEdit to demonstrate the effectiveness. $*$ indicates the use of the mask provided by PIE-Bench [7]. The best and second-best results are shown in **bold** and underline respectively.

Method	Model	Structure Distance ($\times 10^3$)↓	BG Preservation PSNR↑	BG Preservation SSIM ($\times 10^2$)↑	CLIP Sim.↑ Whole	CLIP Sim.↑ Edited	Rank Avg. ↓
P2P [60]	Diffusion	69.43	17.87	71.14	25.01	22.44	20.4
PnP [6]	Diffusion	28.22	22.28	79.05	25.41	22.55	16.5
PnP-Inversion [7]	Diffusion	24.29	22.46	79.68	25.41	22.62	14.5
MasaCtrl [48]	Diffusion	28.38	22.17	79.67	23.96	21.16	19.8
InfEdit [47]	Diffusion	13.78	28.51	85.66	25.03	22.22	11.2
FlowAlign [61]	SD3	28.55	25.23	87.69	25.28	22.01	13.8
SplitFlow [62]	SD3	14.55	25.22	87.54	26.23	23.01	6.4
FTEdit [53]	SD3.5	21.67	23.95	89.48	25.65	22.15	11.4
RF-Inversion [63]	FLUX	40.60	20.82	71.92	25.20	22.11	20.2
DCEdit [64]	FLUX	22.36	25.41	85.60	25.47	22.71	9.8
FlowEdit [65]	FLUX	27.82	21.96	83.08	25.25	22.58	15.9
DNAEdit [66]	FLUX	18.87	24.99	85.71	25.79	22.87	9.0
KV-Edit* [56]	FLUX	68.34	33.22	93.71	25.85	23.42	7.6
RF-Solver [8]	FLUX	31.10	22.90	81.90	26.00	22.88	13.6
w/ ProEdit(Ours)	FLUX	27.82	24.77	84.78	26.28	23.25	9.8(↓ 3.8)
FireFlow [9]	FLUX	28.30	23.28	82.82	25.98	22.94	12.6
w/ ProEdit(Ours)	FLUX	27.51	24.78	85.19	26.28	23.24	9.1(↓ 3.5)
UniEdit [10](α=0.6)	FLUX	10.14	29.54	90.42	25.80	22.33	7.6
w/ ProEdit(Ours)	FLUX	**8.90**	30.59	91.17	25.83	22.37	6.6(↓ 1.0)
UniEdit [10](α=0.8)	FLUX	26.85	24.10	84.86	26.97	23.51	8.6
w/ ProEdit(Ours)	FLUX	24.27	24.82	85.87	27.08	23.64	6.2 (↓ 2.4)
w/ ProEdit(Ours)*	FLUX	23.16	**33.92**	**94.12**	**27.12**	**23.73**	**2.4 (↓ 6.2)**

moderate range. Our experiments for different hyperparameter settings in Appendix G demonstrate that choosing a moderate value for β achieves excellent editing performance.

3.5 Overall

The complete process of our pipeline can be summarized as Figure 4: During the inversion stage, the source image and source prompt are input into the model to conduct the inversion process, and $\boldsymbol{K}_s^l$ and $\boldsymbol{V}_s^l$ are cached in the fly. Then, the attention map is processed to obtain the mask of the editing region, and the inverted noise is output as the initial input for the sampling stage.

In the sampling stage, inverted noise passes through the Latents-Shift module to obtain fused noise, which is then input into the model along with the target prompt for sampling. During the sampling process, the source visual attention features $\boldsymbol{K}_s^l$ and $\boldsymbol{V}_s^l$, obtained from inversion, are injected through the KV-mix module, and the model generates the target image after multiple steps of sampling.

4 Experiments

4.1 Setup

Implementation. We primarily produce experiments using FLUX.1-[dev] [27] for image editing and HunyuanVideo-720p [28] for video editing. For image editing, we have made ProEdit plug-and-play for flow-based inversion methods: RF-Solver, FireFlow, and UniEdit. Notably, UniEdit uses α to denote the delay injection rate, and experiments were conducted with $\alpha = 0.6$ and $\alpha = 0.8$. In our experiments, unless otherwise specified, the delay rate α in UniEdit is set to 0.8. For video editing, we have made ProEdit plug-and-play for RF-Solver. For image and video editing, we set the mix strength $\delta = 0.9$ to balance source content preservation and editing performance. The fusion ratio β is set to 0.25 to achieve the best editing results. We compare ProEdit with previous state-of-the-art training-free visual editing methods [60, 6, 7, 48, 47, 63, 64, 53, 62, 65, 61, 56, 66, 8, 9, 10, 67, 68, 69, 70] using their official implementations and default settings in a shared environment, except DCEdit [64], where we use provided results. More implementation details can be found in the Appendix C.

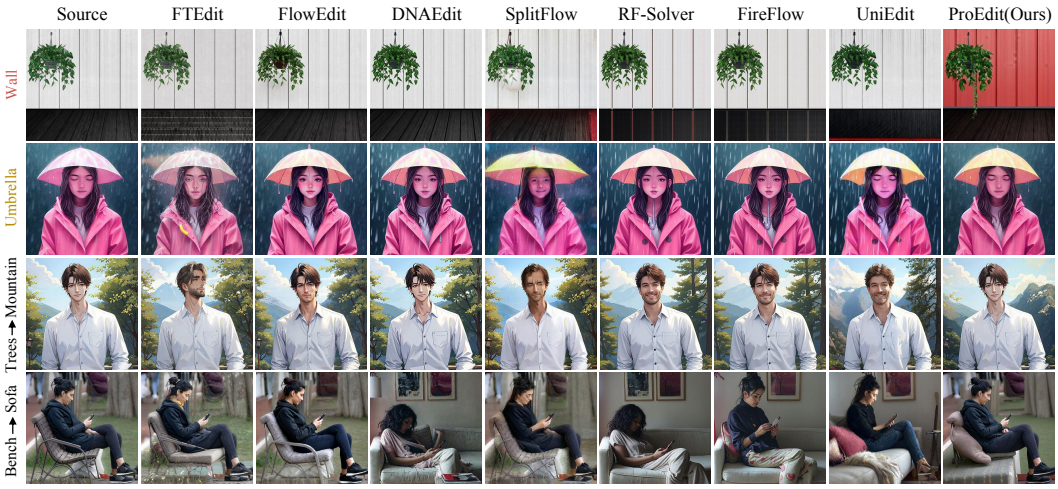

Figure 5: **Qualitative comparison on image editing**. ProEdit achieves semantically consistent editing while maintaining background consistency.

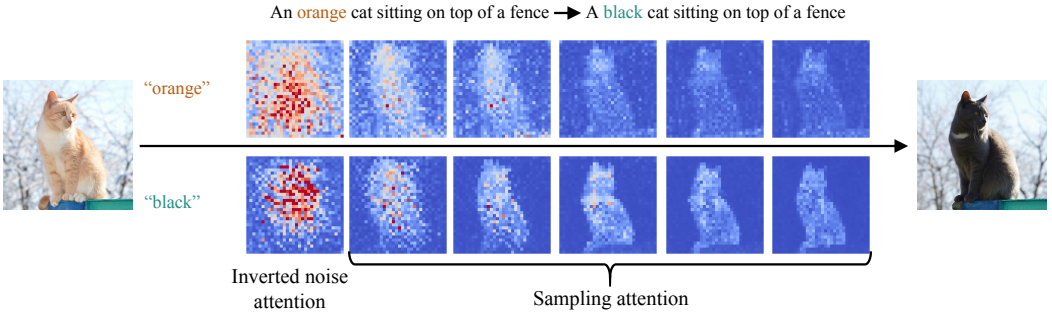

Figure 6: **Visualization of attention maps after performing ProEdit.** The initial distribution is successfully shifted toward the target prompt. During sampling, the model accurately executes the editing task while keeping the background and unedited attributes strictly consistent.

Datasets and Metrics. For text-driven image editing, we evaluate on the PIE-Bench [7], which contains 700 images with 10 different editing types. To evaluate edit-irrelevant context preservation, we use structure distance [71], PSNR [72] and SSIM [73] for annotated unedited regions. The performance of the edits is assessed using CLIP similarity [74] for both the whole image and the edited regions. For text-driven video editing, we collected 55 text-video editing pairs with a resolution of 480×480, 540×960 or 960×540, consist of 40 to 120 frames, including the videos sourced from DAVIS dataset [75] and online platforms. The prompts are derived from GPT-4.1. We follow the metrics proposed in VBench [76, 77], including Subject Consistency, Motion Smoothness, Aesthetic Quality, and Image Quality.

Table 2: **Quantitative comparison on Color Editing**. The best and second-best results are shown in **bold** and underline.

Method	BG Preservation SSIM ($\times 10^2$)↑	CLIP Sim.↑ Whole	CLIP Sim.↑ Edited
RF-Solver	80.21	25.61	20.86
w/ ProEdit(Ours)	86.63	27.30	**22.88**
FireFlow	80.14	26.03	21.02
w/ ProEdit(Ours)	86.53	27.32	22.55
UniEdit	85.39	26.81	21.74
w/ ProEdit(Ours)	**87.54**	**27.36**	22.78

4.2 Text-driven Image Editing

Quantitative Evaluation. Table 1 presents the quantitative results for text-driven image editing. The results in the table show that with our ProEdit, flow-based inversion methods achieved superior results in image editing. Notably, ProEdit combined with the UniEdit inversion method achieves both superior background fidelity and precise semantic edits, resulting in the top average ranking across all metrics and validating the effectiveness of our approach.

Qualitative Evaluation. We compare the performance of our method with several baselines across different types of editing requirements in Figure 5. The baseline methods often fail to maintain the

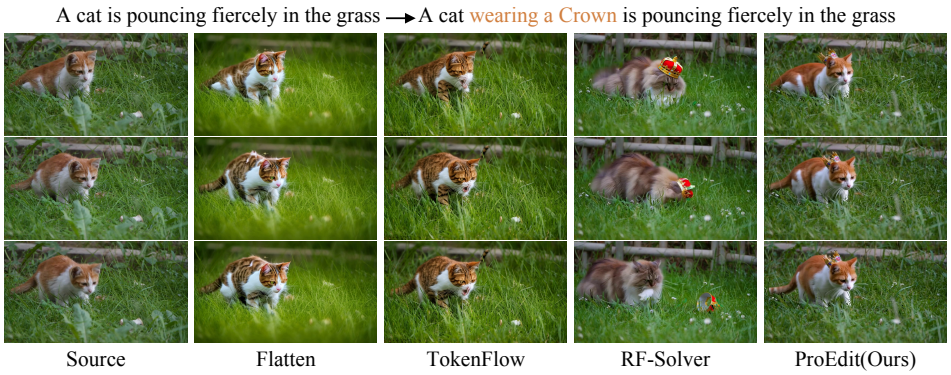

Figure 7: **Qualitative comparison on video editing**. The video comprises 48 frames.

consistency of non-editing attributes such as background and posture, or fail to achieve satisfactory editing results. In contrast, our method achieves high-quality editing results while maintaining the consistency of non-editing content.

Color Editing. To validate that our method addresses the "latent distribution injection" issue overlooked by previous methods, We conducted experiments on the color editing task in PIE-Bench, which is significantly affected by the latent distribution. Table 2 shows the quantitative results for color editing. With our ProEdit, all flow-based inversion methods achieved impressive results. This supports the motivation behind our proposed Latents-Shift module. The AdaIN-based Latents-Shift helps the editing process break free from the constraints imposed by the source image distribution. We further validate it by visualizing the attention map after using Latents-Shift in Figure 6.

Table 3: **Text-driven video editing comparison**. We report the results of each baseline, and evaluate our proposed method using flow-based inversion method RF-Solver. The best and second-best results are shown in **bold** and underline.

Method	SC ↑	MS ↑	AQ ↑	IQ ↑
FateZero [67]	0.9612	0.9740	0.6004	0.6556
Flatten [68]	0.9690	0.9830	0.6318	0.6678
TokenFlow [69]	0.9697	0.9897	0.6436	0.6817
FlowDirector [70]	0.9678	0.9895	0.6244	0.6759
RF-Solver [8]	0.9708	0.9906	0.6497	0.6866
w/ ProEdit(Ours)	**0.9712**	**0.9920**	**0.6518**	**0.6936**

4.3 Text-driven Video Editing

Quantitative Evaluation. Table 3 presents the quantitative results for text-driven video editing. For each metric, we report the average score of all videos in the test dataset. The results in the table show that with our ProEdit, flow-based inversion methods achieved superior results in video editing. This proves the versatility of our method for flow-based models, demonstrating its applicability to video editing tasks and its ability to improve editing performance.

Qualitative Evaluation. We compare the performance of our method with several methods in Figure 7. The baseline methods often fail to maintain the consistency of non-editing attributes such as background and posture, or fail to achieve satisfactory editing results. In contrast, ProEdit achieves high-quality editing while maintaining spatial and temporal consistency.

4.4 Ablation Study

We evaluate the effectiveness of the proposed **KV-mix** and **Latents-Shift** in Table 4. KV-m, LS mean KV-mix, Latents-Shift in our method. Note that without the modules we proposed, each method is evaluated using its original source code setup. When the KV-mix feature injection mechanism is applied to replace the original feature injection mechanism, we observe a significant improvement in CLIP similarity due to the reduced influence of source features in the attention. After incorporating the Latents-Shift module, the CLIP similarity is further enhanced as the influence of the source image on the

Table 4: **Quantitative comparison for the ablation study**. The best and second-best results for each solver are shown in **bold** and underline.

Solver	KV-m	LS	CLIP Sim. Whole↑	CLIP Sim. Edited↑
RF-Solver			26.00	22.88
	✓		26.21	23.21
	✓	✓	**26.28**	**23.25**
FireFlow			25.98	22.94
	✓		26.22	23.18
	✓	✓	**26.28**	**23.24**
UniEdit			26.97	23.51
	✓		27.02	23.54
	✓	✓	**27.08**	**23.64**

inversion noise latent distribution is eliminated. In summary, the various modules of ProEdit work synergistically through a shared masking mechanism to collectively improve the editing results.

5 Conclusion

In this work, we identified excessive injection of source image information caused by the "inverted latent with global injection" strategy used in existing flow-based inversion editing methods, which leads to sacrificing editing quality to maintain background consistency with the source image during editing. We introduce ProEdit, a novel, training-free method that addresses this issue by proposing the KV-mix and Latents-Shift modules from both the attention and latent perspectives, aiming to eliminate the negative impact of excessive source image information injection on editing quality. Extensive experiments show that ProEdit can be seamlessly integrated into existing flow-based inversion methods, achieving high background consistency and excellent editing quality simultaneously.

6 Acknowledgments and Disclosures

Funding. This work was supported partially by National Key Research and Development Program of China (2023YFA1008503), NSFC(92470202), Guangdong NSF Project (No. 2023B1515040025), Guangdong Key Research and Development Program(No.2024B0101040004, No. 2025B0909020002), the Open Project Program (Grant No. QHSF-CS-2601) of Key Laboratory of Tibetan Information Processing, Ministry of Education.

Competing Interests. The authors declare no competing financial interests related to this work.

References

[1] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2022.

[2] Yusuf Dalva, Kavana Venkatesh, and Pinar Yanardag. Fluxspace: Disentangled semantic editing in rectified flow transformers. In *CVPR*, 2025.

[3] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. In *WACV*, 2025.

[4] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, 2023.

[5] Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. Renoise: Real image inversion through iterative noising. In *ECCV*, 2024.

[6] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, 2023.

[7] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Pnp inversion: Boosting diffusion-based editing with 3 lines of code. In *ICLR*, 2024.

[8] 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. In *ICML*, 2025.

[9] Yingying Deng, Xiangyu He, Changwang Mei, Peisong Wang, and Fan Tang. Fireflow: Fast inversion of rectified flow for image semantic editing. In *ICML*, 2025.

[10] Guanlong Jiao, Biqing Huang, Kuan-Chieh Wang, and Renjie Liao. Uniedit-flow: Unleashing inversion and editing in the era of flow models. In *ICLR*, 2026.

[11] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

[12] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019.

[13] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip Torr. Controllable text-to-image generation. In *NeurIPS*, 2019.

[14] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, 2016.

[15] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-modal contrastive learning for text-to-image generation. In *CVPR*, 2021.

[16] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017.

[17] 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*, 2021.

[18] 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. In *NeurIPS*, 2021.

[19] 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. *TMLR*, 2022.

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

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

[22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[23] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *CVPR*, 2023.

[24] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015.

[26] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

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

[28] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[29] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *PAMI*, 2021.

[30] Phillip Isola Jun-Yan Zhu, Taesung Park and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

[31] 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*, 2024.

[32] Dian Zheng, Cheng Zhang, Xiao-Ming Wu, Cao Li, Chengfei Lv, Jian-Fang Hu, and Wei-Shi Zheng. Panorama generation from nfov image done right. In *CVPR*, 2025.

[33] Yaowei Li, Yuxuan Bian, Xuan Ju, Zhaoyang Zhang, Ying Shan, Yuexian Zou, and Qiang Xu. Brushedit: All-in-one image inpainting and editing. *PAMI*, 2026.

[34] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.

[35] Hongyu Li, Manyuan Zhang, Dian Zheng, Ziyu Guo, Yimeng Jia, Kaituo Feng, Hao Yu, Yexin Liu, Yan Feng, Peng Pei, et al. Editthinker: Unlocking iterative reasoning for any image editor. *arXiv preprint arXiv:2512.05965*, 2025.

[36] Trong-Tung Nguyen, Quang Nguyen, Khoi Nguyen, Anh Tran, and Cuong Pham. Swiftedit: Lightning fast text-guided image editing via one-step diffusion. In *CVPR*, 2025.

[37] Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: exact diffusion inversion via coupled transformations. In *CVPR*, 2023.

[38] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 2025.

[39] Fangyikang Wang, Hubery Yin, Yuejiang Dong, Huminhao Zhu, Zhang Chao, Hanbin Zhao, Hui Qian, and Chen Li. BELM: bidirectional explicit linear multi-step sampler for exact inversion in diffusion models. In *NeurIPS*, 2024.

[40] Guoqiang Zhang, John P. Lewis, and W. Bastiaan Kleijn. Exact diffusion inversion via bidirectional integration approximation. In *ECCV*, 2024.

[41] Wenkai Dong, Song Xue, Xiaoyue Duan, and Shumin Han. Prompt tuning inversion for text-driven image editing using diffusion models. In *ICCV*, 2023.

[42] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, 2023.

[43] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *SIGGRAPH*, 2023.

[44] Qiucheng Wu, Yujian Liu, Handong Zhao, Ajinkya Kale, Trung Bui, Tong Yu, Zhe Lin, Yang Zhang, and Shiyu Chang. Uncovering the disentanglement capability in textto-image diffusion models. In *CVPR*, 2023.

[45] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. In *CVPR*, 2024.

[46] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. In *ICLR*, 2023.

[47] Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with natural language. In *CVPR*, 2024.

[48] 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 *CVPR*, 2023.

[49] Yoad Tewel, Rinon Gal, Dvir Samuel, Yuval Atzmon, Lior Wolf, and Gal Chechik. Add-it: Training-free object insertion in images with pretrained diffusion models. In *ICLR*, 2025.

[50] Senmao Li, Joost van de Weijer, Taihang Hu, Fahad Shahbaz Khan, Qibin Hou, Yaxing Wang, and Jian Yang. Stylediffusion: Prompt-embedding inversion for text-based editing. *CVM*, 2025.

[51] 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 *CVPR*, 2023.

[52] Minghong Cai, Xiaodong Cun, Xiaoyu Li, Wenze Liu, Zhaoyang Zhang, Yong Zhang, Ying Shan, and Xiangyu Yue. Ditctrl: Exploring attention control in multi-modal diffusion transformer for tuning-free multi-prompt longer video generation. In *CVPR*, 2025.

[53] Pengcheng Xu, Boyuan Jiang, Xiaobin Hu, Donghao Luo, Qingdong He, Jiangning Zhang, Chengjie Wang, Yunsheng Wu, Charles Ling, and Boyu Wang. Unveil inversion and invariance in flow transformer for versatile image editing. In *CVPR*, 2025.

[54] Omri Avrahami, Or Patashnik, Ohad Fried, Egor Nemchinov, Kfir Aberman, Dani Lischinski, and Daniel Cohen-Or. Stable flow: Vital layers for training-free image editing. In *CVPR*, 2025.

[55] Yu Xu, Fan Tang, Juan Cao, Yuxin Zhang, Xiaoyu Kong, Jintao Li, Oliver Deussen, and Tong-Yee Lee. Headrouter: A training-free image editing framework for mm-dits by adaptively routing attention heads. *ACM Trans. Graph.*, 2026.

[56] Tianrui Zhu, Shiyi Zhang, Jiawei Shao, and Yansong Tang. Kv-edit: Training-free image editing for precise background preservation. In *ICCV*, 2025.

[57] Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *JMLR*, 2025.

[58] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

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

[60] 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.

[61] Jeongsol Kim, Yeobin Hong, Jonghyun Park, and Jong Chul Ye. Flowalign: Trajectory-regularized, inversion-free flow-based image editing. In *ICLR*, 2026.

[62] Sung-Hoon Yoon, Minghan Li, Gaspard Beaudouin, Congcong Wen, Muhammad Rafay Azhar, and Mengyu Wang. Splitflow: Flow decomposition for inversion-free text-to-image editing. In *NeurIPS*, 2025.

[63] Litu Rout, Yujia Chen, Nataniel Ruiz, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Semantic image inversion and editing using rectified stochastic differential equations. In *ICLR*, 2025.

[64] Yihan Hu, Jianing Peng, Yiheng Lin, Ting Liu, Xiaochao Qu, Luoqi Liu, Yao Zhao, and Yunchao Wei. Dcedit: Dual-level controlled image editing via precisely localized semantics. *arXiv preprint arXiv:2503.16795*, 2025.

[65] Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *ICCV*, 2025.

[66] Chenxi Xie, Minghan Li, Shuai Li, Yuhui Wu, Qiaosi Yi, and Lei Zhang. Dnaedit: Direct noise alignment for text-guided rectified flow editing. In *NeurIPS*, 2025.

[67] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. In *CVPR*, 2023.

[68] Yuren Cong, Mengmeng Xu, Christian Simon, Shoufa Chen, Jiawei Ren, Yanping Xie, Juan-Manuel Perez-Rua, Bodo Rosenhahn, Tao Xiang, and Sen He. Flatten: optical flow-guided attention for consistent text-to-video editing. In *ICLR*, 2024.

[69] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. In *ICLR*, 2024.

[70] Guangzhao Li, Yanming Yang, Chenxi Song, and Chi Zhang. Flowdirector: Training-free flow steering for precise text-to-video editing. In *CVPR*, 2026.

[71] Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Splicing vit features for semantic appearance transfer. In *CVPR*, 2022.

[72] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 2008.

[73] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004.

[74] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[75] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[76] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024.

[77] Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, et al. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

[78] Qwen Team. Qwen3 technical report, 2025.

Addendix

A Mask Extraction Details

According to the target object to be edited, we extract a latent-space binary mask from the attention map of the corresponding token, where the threshold is set as the mean attention score plus a small marginal bias. When extracting the mask from the attention map, we adopted a strategy different from DitCtrl [52]. In our observation, the attention map of the last Double block effectively associates the relevant regions of text and image. Figure A presents an example showing that both the editing region mask extracted from the last Double block's attention map and the mask extracted from the average of all attention maps successfully segment the editing and non-editing regions. Additionally, this mask extraction method reduces memory consumption. Therefore, we extract our editing region mask $\boldsymbol{M}$ from the attention map of the last Double block. Notably, we always extract the mask from the first step of inversion, as the images at these time steps are least affected by noise and have the best text-to-image correlation.

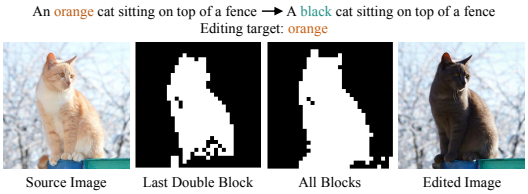

Figure 8: **A visual comparison of the editing region mask extracted from the last Double block and all blocks.** Using "orange" as the editing target, the editing region masks extracted from both the last Double block and all blocks effectively segment the editing region.

Due to the downsampling operation performed in the feature space relative to the pixel space, the extracted mask has a coarser granularity and therefore cannot fully cover the boundary regions of the editing target in the pixel space. To address this, we apply a diffusion operation to the mask, expanding it outward by one step to obtain a broader mask that can fully cover the editing area and enable text-aligned editing. Given the strong global adaptability of generative models, such relatively coarse masking in the latent space is sufficient for semantic alignment. Moreover, the boundary coverage between the editing and non-editing regions helps smooth the edges and avoid image artifacts. As shown in Figure 9, our mask diffusion operation effectively resolves the issues of image artifacts and incomplete editing that may occur during the editing process.



Figure 9: **Effect of the Mask Diffusion Strategy**. Without mask diffusion, the results may contain image artifacts and incomplete edits. Our mask diffusion strategy expands the mask boundary to better cover the editing region, effectively reducing artifacts and improving editing quality.

B Mask Accuracy Analysis

To evaluate the reliability of the masks extracted via our attention-mechanism-based approach, we conduct a quantitative accuracy analysis by comparing our generated masks with the ground truth (GT) masks. We employ the Intersection over Union (IoU) as the primary metric to measure the spatial alignment between our attention-derived masks and the GT annotations.

We conducted experiments using 70 images and their corresponding prompts from the PIE-Bench [7]. As shown in Table 5, our method achieves a high average IoU of 0.7427. Such a high degree of overlap demonstrates that the attention maps within the FLUX-based MM-DiT blocks inherently capture the precise semantic boundaries of target objects. This high-fidelity mask generation ensures that the KV-mix and Latents-shift modules are applied to accurate spatial regions,

Table 5: **Quantitative analysis of mask accuracy.**

vs. GT mask	Average IOU	IOU > 0.6	IOU > 0.7	IOU > 0.8
Attention mask	0.7427	0.89	0.60	0.38

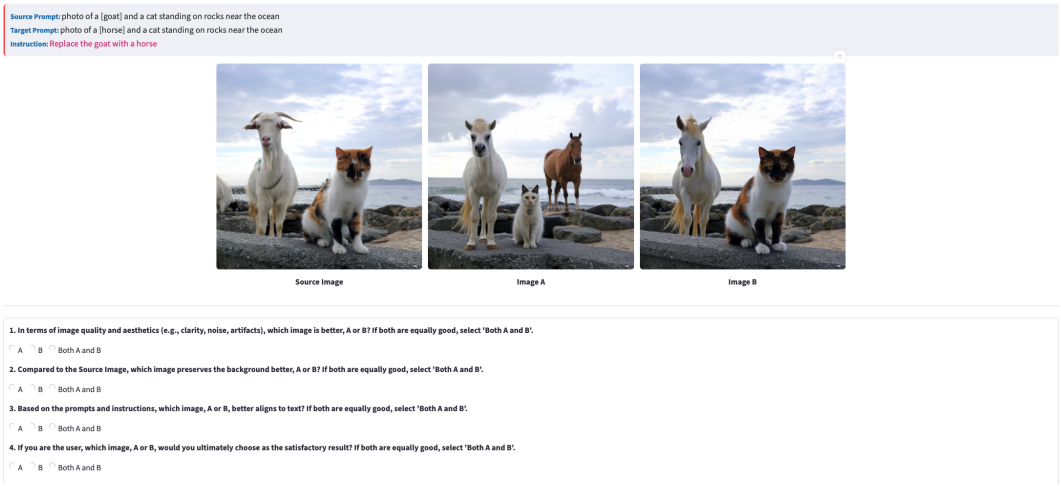

Figure 10: **User study**. We provide a sample where participants were presented with the original image, editing prompts, results from two different methods for comparison and four questions from four aspects.

effectively preventing semantic leakage into the background while maintaining structural integrity throughout the editing process.

C Implementation Details

For image and video editing, we conduct experiments on an L40 (48GB). For image editing, we use 15 steps. For video editing, we use 25 steps. We follow the default experimental configurations of RF-Solver, FireFlow, and UniEdit when applying our method as a plug-and-play module to these baselines. In the quantitative image editing experiments, when we use the masks provided by PIE-Bench, the input masks are first downsampled using bilinear interpolation, and then transformed from single-channel to 64-channel representations following the VAE in Flux [27].

D Quantitative Results of Attention Feature Combination Effect

We evaluated the effectiveness of different attention feature combinations applied in the fusion injection mechanism using the RF-Solver inversion method on PIE-Bench to verify the superiority of our proposed **KV-mix** module. Note that the $\boldsymbol{V}$ attention feature is the most important attention feature for editing quality, so all the attention feature combinations we evaluated include $\boldsymbol{V}$. Table 6 shows the quantitative results of different attention feature combinations in the fusion injection mechanism. Among the four combinations evaluated, the $\boldsymbol{KV}$ combination demonstrated satisfactory results in both background consistency and editing quality. Therefore, we adopted the $\boldsymbol{KV}$ fusion injection mechanism and designed the **KV-mix** module. We observe that when $\boldsymbol{Q}$ is included in the combination, the alignment between the editing results and the text prompts tends to decrease. This can be attributed to the fact that the query feature plays a more direct role in determining how the current representation performs attention-based retrieval. When the $\boldsymbol{Q}$ from the source image is additionally injected, the attention distribution becomes more biased toward the semantic arrangement of the source image, which imposes stronger structural and appearance constraints on the generation process. As a conse-

Table 6: **Quantitative experiments on different attention feature combinations**. The best and second-best results are shown in **bold** and underline respectively.

Method	BG Preservation		CLIP Sim.↑	
	PSNR↑	SSIM ($\times 10^2$)↑	Whole	Edited
$\boldsymbol{Q}\&\boldsymbol{V}$	24.04	82.24	26.16	23.04
$\boldsymbol{Q}\&\boldsymbol{K}\&\boldsymbol{V}$	<u>24.51</u>	<u>83.04</u>	26.20	22.97
$\boldsymbol{V}$	23.69	81.68	<u>26.26</u>	<u>23.15</u>
$\boldsymbol{K}\&\boldsymbol{V}$	**24.77**	**84.78**	**26.28**	**23.25**

Table 8: Quantitative experiments on global editing. The best results are shown in **bold** respectively.

Method	Structure Distance $\times 10^3$ (↓)	Whole CLIP (↑)	Edited CLIP (↑)
RF-Solver	29.59	26.43	26.43
w/ ProEdit(Ours)	28.86	26.76	26.76
FireFlow	27.45	26.56	26.56
w/ ProEdit(Ours)	26.97	26.83	26.83
UniEdit-Flow	27.37	28.58	28.58
w/ ProEdit(Ours)	**26.31**	**28.64**	**28.64**

quence, the regions that are supposed to be modified cannot be edited sufficiently, leading to reduced editing flexibility.

E User Study

We conduct an extensive user study to compare our method with three baselines: RF-Solver [8], Fireflow [9], and UniEdit-Flow [10]. We used 70 images from PIE-Bench [7] across different editing categories, while preserving the distribution of editing categories. More than 20 participants are asked to compare each pair of methods based on four criteria: image quality, background preservation, text alignment, and overall satisfaction. As shown in Table 7, our method significantly outperforms the previous methods. Additionally, users' feedback reveals that alignment with editing instructions plays a crucial role in their final choices

As shown in Figure 10, we conduct our user study in a questionnaire format to collect user preferences for different methods. We observed that when the difference between two images on a particular evaluation metric is marginal, it is often difficult for participants to make a clear preference. Therefore, across the four aspects of image quality, background consistency, text alignment, and overall evaluation, we allowed participants to select "equally good".

Table 7: **User Study**. We compared our method with three popular baselines. We report the win rates of our method compared to baseline. Random* denotes the win rate of random choices.

ours vs.	Quality↑	Background↑	Text↑	Overall↑
Random*	50.0%	50.0%	50.0%	50.0%
RF-Solver [8]	68.4%	92.0%	91.9%	91.8%
Fireflow [9]	78.7%	86.3%	88.5%	86.6%
UniEdit [10]	68.8%	84.9%	81.6%	80.9%

We implement a single-blind mechanism where the compared methods for each question are randomly sampled, ensuring fairness in the comparison. We collect over 2,000 comparison results and calculate the win rate of our method after excluding the cases where both methods are rated equally.

F Global and Local Editing

The mask $\boldsymbol{M}$ specifies where an edit takes effect and supports both local and global editing; for global edits, it covers the entire image. In this setting, spatial KV-mixing helps preserve image structure, while the AdaIN-based Latents-Shift module facilitates overall style transformations.

To evaluate the effectiveness of ProEdit on global edits (e.g., style transfer) and local edits, we partition PIE-Bench based on whether background metrics (PSNR/SSIM) can be computed. Samples with NaN background scores are treated as global (whole-image) edits, while the remaining samples are treated as local (object-attribute) edits. As shown in Tables 8 and 9, ProEdit improves the performance of all three baselines in both global and local editing, demonstrating that our framework is not restricted to localized masks. The reductions in structure distance are larger for local edits, consistent with KV-mix benefiting more from spatial selectivity in this setting.

Table 9: Quantitative experiments on local editing. The best results are shown in **bold** respectively.

Method	Structure Distance $\times 10^3$ (↓)	Whole CLIP (↑)	Edited CLIP (↑)
RF-Solver	31.49	25.89	21.96
w/ ProEdit(Ours)	27.55	26.16	**22.34**
FireFlow	28.52	25.82	22.00
w/ ProEdit(Ours)	27.62	26.14	22.31
UniEdit-Flow	26.71	26.55	22.19
w/ ProEdit(Ours)	**23.74**	**26.68**	**22.34**

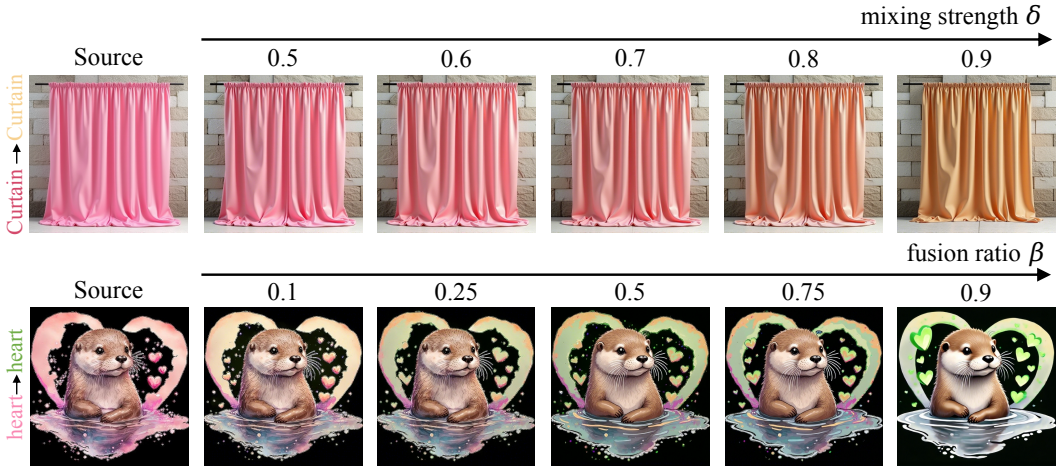

Figure 11: **Qualitative Results of Hyperparameter Experiment**.

G Hyperparameter Analysis

We conducted experiments and analyses on the hyperparameter settings to demonstrate how our method influences the editing results and assist users in better configuring the editing parameters. When conducting hyperparameter experiments on the mixing strength δ, we set the fusion ratio β to 0.25. Conversely, when experimenting with the fusion ratio β, we set the mixing strength δ to 0.9. As shown in Figure 11, regarding the mixing strength δ, our results indicate that a value exceeding 0.8 is necessary to initiate the semantic transition—such as the color shift from pink to yellow in curtains—while an optimal editing outcome is achieved at 0.9. For the fusion ratio β, we observed that a low value of 0.1 leads to editing failure; conversely, while a high ratio such as 0.9 can successfully execute the edit, it tends to adversely affect the image style. Balanced results were consistently obtained within the median range of 0.25 to 0.75. This is consistent with our analysis in Section 3.3 and Section 3.4 of the main text, excellent editing performance is achieved when the mixing strength δ is set to a higher value and the fusion ratio β is set to a moderate value.

Table 10: **Quantitative analysis of hyperparameter sensitivity**.

(δ, β)	RF-Solver	(0.85,0.25)	(0.85,0.5)	(0.85,0.75)	(0.9,0.25)	(0.9,0.5)	(0.9,0.75)	(0.95,0.25)	(0.95,0.5)	(0.95,0.75)
Structure Distance↓	31.10	25.67	26.7	28.38	27.82	28.72	28.96	28.66	29.05	29.96
CLIP Sim. Edited ↑	22.88	23.17	23.15	23.19	23.25	23.22	23.27	23.19	23.21	23.26

To further investigate the sensitivity of our method to hyperparameter configurations, we conducted qualitative experiments within the parameter range of 0.85 to 0.95 for the mixing strength δ, and 0.25 to 0.75 for the fusion ratio β. As shown in Table 10, our method is generally stable under different hyperparameter settings. All tested combinations outperform RF-Solver in both Structure Distance and CLIP similarity. The quantitative results further demonstrate that employing a higher mixing strength δ (0.85-0.95) alongside a moderate fusion ratio β (0.25-0.75) achieves an optimal balance

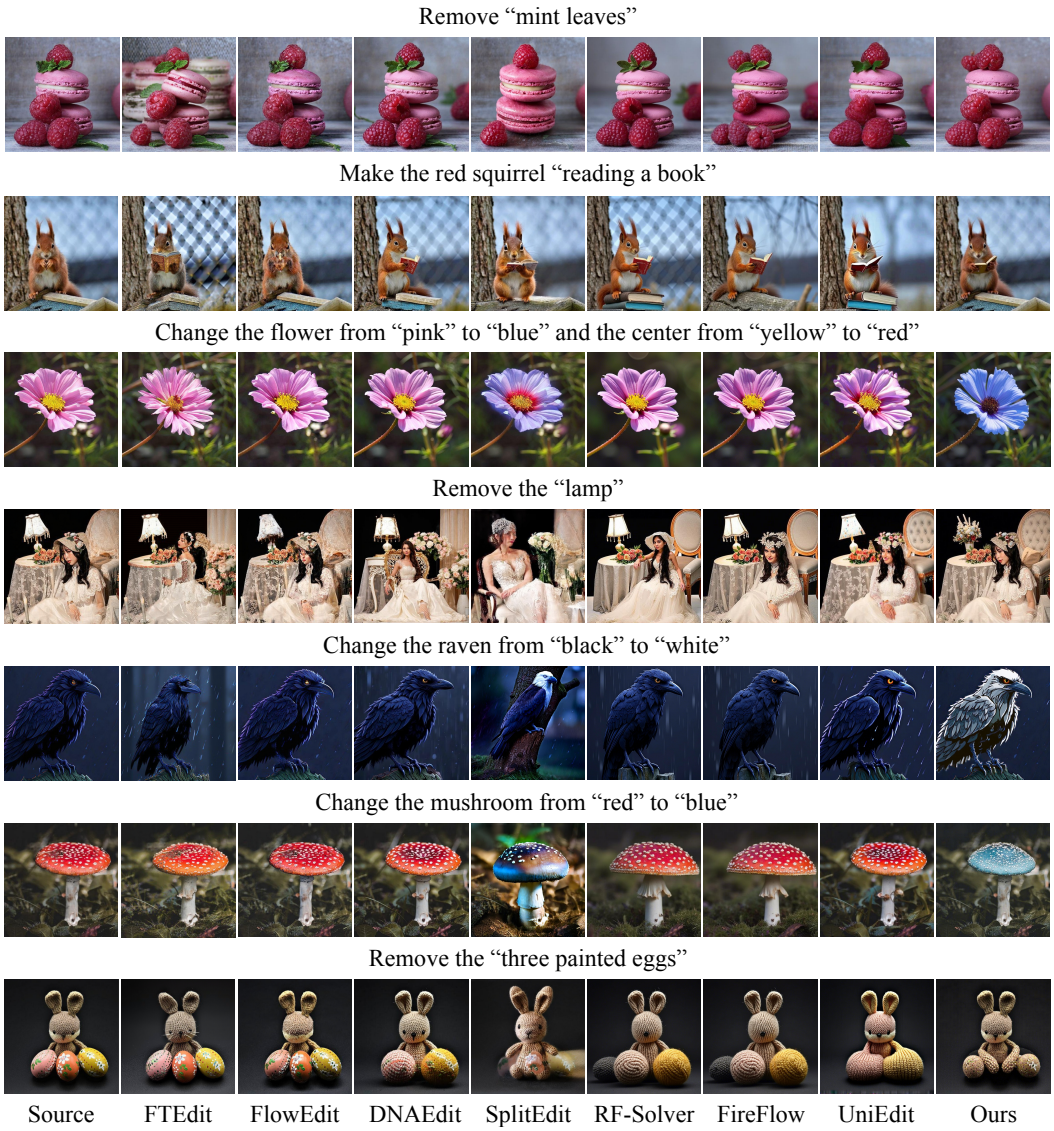

Figure 12: **More qualitative comparison of image editing on PIE-Bench[7].**

between editing accuracy and structural preservation. Within this parameter range, our method exhibits remarkable robustness, consistently delivering superior editing performance.

Hyperparameter selection. Within the tested ranges, we recommend slightly decreasing δ and β for mild edits (e.g., slightly darkening an object's red color) to better preserve the original image structure. For edits involving larger semantic changes (e.g., changing an object from deep red to deep blue), moderately increasing β can improve alignment with the target edit. Users can therefore adjust δ and β according to the editing requirements to balance editing accuracy and structural preservation.

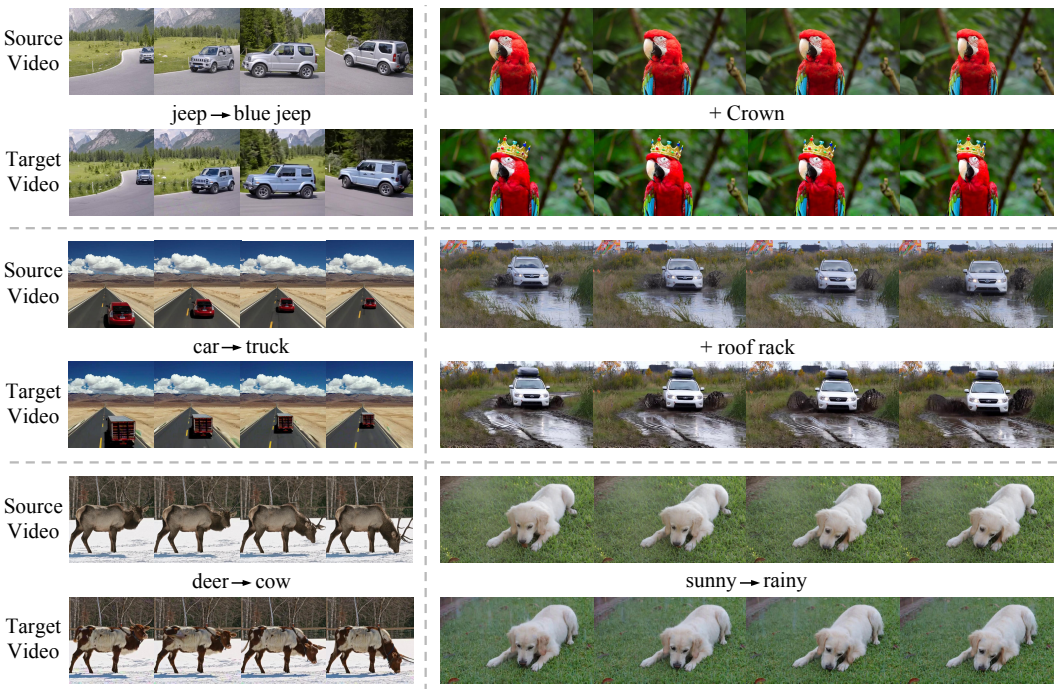

Figure 13: **More video editing results.**

H Prompts Sensitivity Analysis

To evaluate the robustness of our approach to prompt variations, we conducted a qualitative comparison between results generated using detailed, descriptive prompts (Long prompts) and those generated using simplified prompts containing only core key phrases (Short prompts).

As illustrated in Figure 14, our method exhibits remarkable stability across varying levels of prompt detail. Notably, ProEdit achieves precise semantic transformations using only concise, keyword-based prompts while strictly preserving the structural integrity of the source image, yielding results nearly indistinguishable from those generated with extensive descriptions. Nevertheless, detailed prompts provide beneficial explicit guidance that facilitates more refined and nuanced editing effects. Overall, the core editing capability of our approach remains robust across a broad spectrum of prompt complexities. Nevertheless, more detailed prompts can further unleash the full potential of our method, leading to even more refined editing results.

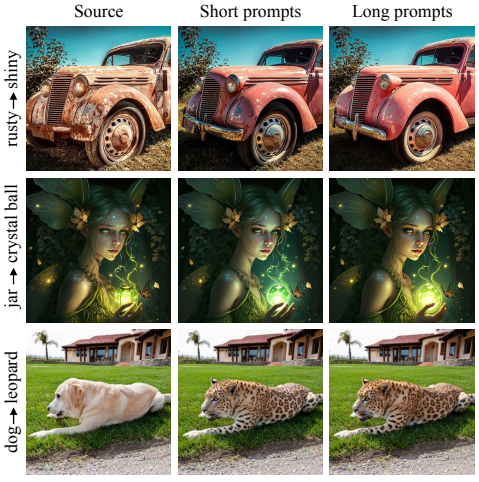

Figure 14: **Qualitative Results of Prompts Sensitivity Experiment**.

I More Qualitative Results

I.1 More Qualitative Results for Image Editing

Here we provide more qualitative results for image editing in Figure 12. In cases where other inversion-based editing methods [7, 6, 8, 10, 9] fail, result in insufficient edits, or fail to maintain

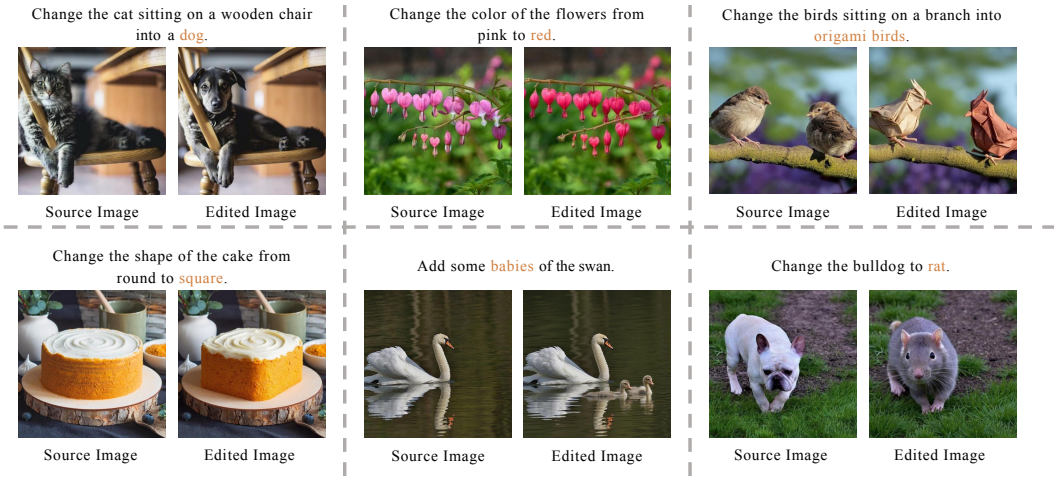

Figure 15: **Qualitative results of image editing based on editing instruction.** The actual input editing instruction are shown above each source image and its corresponding edited image.

consistency, our method successfully achieves semantically consistent editing and demonstrates impressive performance.

I.2 More Qualitative Results for Video Editing

Here we provide more qualitative results for video editing in Figure 13. Our method demonstrates impressive performance in various types of video editing tasks.

J Editing by Instruction

To lower the barrier for using our method and make it more user-friendly, we introduce a large language model Qwen3-8B [78] to enable editing based on editing instructions. Figure 15 shows the qualitative results of our method based on editing instruction. With the assistance of a large language model, our method can directly perform edits guided by editing instructions.

K Extension to Inversion-Free Editing

Excessive source-information injection may also occur in inversion-free editing methods. Taking FlowEdit as an example, although inversion is not performed, the source latent and source-conditioned velocity remain involved in every editing update, potentially limiting structural changes. Moreover, FlowEdit initializes editing directly from the source latent, retaining its distribution, which may hinder high-conflict color editing.

Table 11: Color editing results with FlowEdit. The best results are shown in **bold** respectively.

Method	Whole CLIP Sim. ↑	Edited CLIP Sim. ↑
Vanilla FlowEdit	26.65	21.31
FlowEdit + Latents-Shift	**26.98**	**21.54**

As described in Algorithm 1, we make a simple adaptation of our Latents-Shift module to FlowEdit and evaluate it on color editing tasks, with quantitative results reported in Table 11. These results support our view that excessive source-information injection can also arise in inversion-free methods and suggest a potential direction for improvement to the visual editing community.

Algorithm 1 Latents-Shift Adaptation for FlowEdit

Input: Source latent x_{src}, source prompt c_{src}, flow model m, shift ratio $\beta = 0.25$, initial timestep t_0
Output: Shifted initial editing latent z_{edit}

$z_{\mathrm{edit}} \leftarrow x_{\mathrm{src}}$ ▷ Initialize the editing latent
Sample $\epsilon_0 \sim \mathcal{N}(0, I)$
$z_{\mathrm{src}} \leftarrow (1 - t_0)x_{\mathrm{src}} + t_0\epsilon_0$
$(V_{\mathrm{src}}, Attn) \leftarrow m(z_{\mathrm{src}}, c_{\mathrm{src}}, t_0)$ ▷ Obtain source velocity and attention
$M \leftarrow \mathrm{ExtractMask}(Attn)$ ▷ Extract the editing-region mask
$z_{\mathrm{shift}} \leftarrow \mathrm{std}(\epsilon_0) \cdot \frac{z_{\mathrm{edit}} - \mathrm{mean}(z_{\mathrm{edit}})}{\mathrm{std}(z_{\mathrm{edit}})} + \mathrm{mean}(\epsilon_0)$
$z^M_{\mathrm{edit}} \leftarrow \beta z_{\mathrm{shift}} + (1 - \beta) z_{\mathrm{edit}}$
$z_{\mathrm{edit}} \leftarrow M \odot z^M_{\mathrm{edit}} + (1 - M) \odot z_{\mathrm{edit}}$ ▷ Preserve the non-edited region
return z_{edit}

L Discussions and Broader Impacts

L.1 Limitations

ProEdit is mainly designed for inversion-based editing with flow-based generative models. Although it can be integrated into several representative inversion and editing pipelines in a plug-and-play manner, its effectiveness may still vary across different model architectures, attention designs, solvers, resolutions, and sampling schedules. Extending ProEdit to a broader family of generative models, such as diffusion models with different architectures or future video foundation models, requires further validation. In addition, video editing remains challenging in more complex scenarios. For long videos, complex camera motion, severe occlusions, or large object deformation, the edited results may still suffer from temporal flickering or inconsistent edits across frames, since ProEdit does not explicitly model long-range temporal correspondence. Finally, real-world editing instructions can be more ambiguous, compositional, or subjective than benchmark prompts. For example, edits involving multiple interacting objects, global style changes together with local attribute changes, or instructions requiring physical and semantic reasoning may still be challenging. Future work may explore more comprehensive evaluation protocols, stronger instruction-understanding mechanisms, and motion-aware or tracking-based designs to improve robustness in broader editing scenarios.

L.2 Broader Impact

ProEdit provides a training-free and flexible approach for text-driven image and video editing. By improving the balance between editability and content preservation, it can benefit a wide range of creative and practical applications, such as visual content creation, design prototyping, film and video post-production, virtual try-on, data augmentation, and assistive visual editing tools. Since the method can be integrated into existing inversion-based editing pipelines, it may also reduce the cost of adapting powerful generative models to downstream editing tasks.

However, more powerful visual editing methods also raise potential risks. The ability to modify images and videos according to text prompts may be misused to create misleading, deceptive, or harmful visual content, including manipulated media, impersonation, or misinformation. These risks are particularly relevant for realistic human images, news-related content, and videos, where edited results may be difficult for viewers to distinguish from authentic recordings.

To mitigate these concerns, we encourage responsible use of ProEdit and related editing techniques. Possible safeguards include watermarking or provenance tracking for generated and edited content, clear disclosure when images or videos have been edited, restrictions on harmful use cases, and the development of detection tools for manipulated media. We also recommend that applications built on top of such methods include user-facing policies and safety filters, especially when editing human identities, sensitive events, or public-interest content.

In addition, biases inherited from the underlying generative models may appear in the edited outputs. Since ProEdit builds upon pretrained text-to-image and text-to-video models, it may reflect their limitations in terms of demographic representation, cultural stereotypes, object-text associations,

or visual quality across different domains. Careful auditing across diverse prompts, subjects, and cultural contexts is therefore important before deployment in real-world systems.

Overall, we believe that ProEdit contributes positively to controllable visual generation by improving instruction alignment while preserving source content. At the same time, its deployment should be accompanied by appropriate transparency, safety, and accountability mechanisms to reduce potential misuse and social harm.

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 main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: In Appendix section, we discuss the limitations of this work.

   Guidelines:

   - The answer [N/A] 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: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] 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 disclose all the information needed to reproduce the main experimental results of the paper in the Method, Experiment and Appendix sections. We will release the code and data upon acceptance of the paper.

   Guidelines:

   - The answer [N/A] 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: [No]

Justification: We will release the code and data upon acceptance of the paper. We disclose all the information needed to reproduce the main experimental results of the paper in the Method, Experiment and Appendix sections.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies all the training and test details necessary to understand the results.

   Guidelines:

   - The answer [N/A] 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: Error bars are not reported because it would be too computationally expensive.

   Guidelines:

   - The answer [N/A] 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: The paper provides sufficient information on the computer resources in Appendix.

   Guidelines:

   - The answer [N/A] 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: The research conducted in the paper conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer [N/A] 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: The paper discusses potential societal impacts of the paper in Appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not involve data or models that have a high risk for misuse.

    Guidelines:

    - The answer [N/A] 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 paper properly credits the creators or original owners of all used assets, and properly respects the license and terms of use.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] 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: We involve human labors in anontating test data and scoring the editing performance, and we include the instruction details for these two parts in Experiment and Appendix sections.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: There are no risks to human participants, and no risk disclosure or IRB approval is required.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.