DCI: Dual-Conditional Inversion for Boosting Diffusion-Based Image Editing

Zixiang Li[1,2], Haoyu Wang[1,2], Wei Wang[1,2], Chuangchuang Tan[1,2], Yunchao Wei[1,2], Yao Zhao[1,2]*
[1]Institute of Information Science, Beijing Jiaotong University
[2]Visual Intelligence +X International Cooperation Joint Laboratory of MOE

Abstract

Diffusion models have achieved remarkable success in image generation and editing tasks. Inversion within these models aims to recover the latent noise representation for a real or generated image, enabling reconstruction, editing, and other downstream tasks. However, to date, most inversion approaches suffer from an intrinsic trade-off between reconstruction accuracy and editing flexibility. This limitation arises from the difficulty of maintaining both semantic alignment and structural consistency during the inversion process. In this work, we introduce **Dual-Conditional Inversion (DCI)**, a novel framework that jointly conditions on the source prompt and reference image to guide the inversion process. Specifically, DCI formulates the inversion process as a dual-condition fixed-point optimization problem, minimizing both the latent noise gap and the reconstruction error under the joint guidance. This design anchors the inversion trajectory in both semantic and visual space, leading to more accurate and editable latent representations. Our novel setup brings new understanding to the inversion process. Extensive experiments demonstrate that DCI achieves state-of-the-art performance across multiple editing tasks, significantly improving both reconstruction quality and editing precision. Furthermore, we also demonstrate that our method achieves strong results in reconstruction tasks, implying a degree of robustness and generalizability approaching the ultimate goal of the inversion process. Our codes are available at: https://github.com/Lzxhh/Dual-Conditional-Inversion

1 Introduction

Diffusion models have made significant progress in the field of generative artificial intelligence. Among them, latent Diffusion Models (LDMs) [41] perform the diffusion process in a compressed latent space rather than the pixel space, enabling more efficient and high-quality image generation and editing. This architectural design has made LDMs a powerful and flexible backbone for a wide range of downstream tasks, such as text-to-image generation [36, 40, 43], image editing [31, 4, 48, 3], image restoration [29, 51, 54], style transfer [52, 50, 7], *etc*. In the image editing tasks, the editing is achieved by manipulating the diffusion latent representations. However, in most cases, the corresponding latent representation for a given image is not directly available, which means that we must first perform an inversion process to obtain their latent representations.

The earliest inversion method is DDPM [16], and it has inspired the development of numerous related methods [47, 2, 19]. DDPMs add random noise at each timestep, which leads to the loss of information contained in the original image, resulting in poor reconstruction and editing effects. DDIM inversion [45, 10] reformulates the diffusion process to be deterministic as solving an implicit equation under the assumption that consecutive points along the denoising trajectory remain close.

*Corresponding author

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

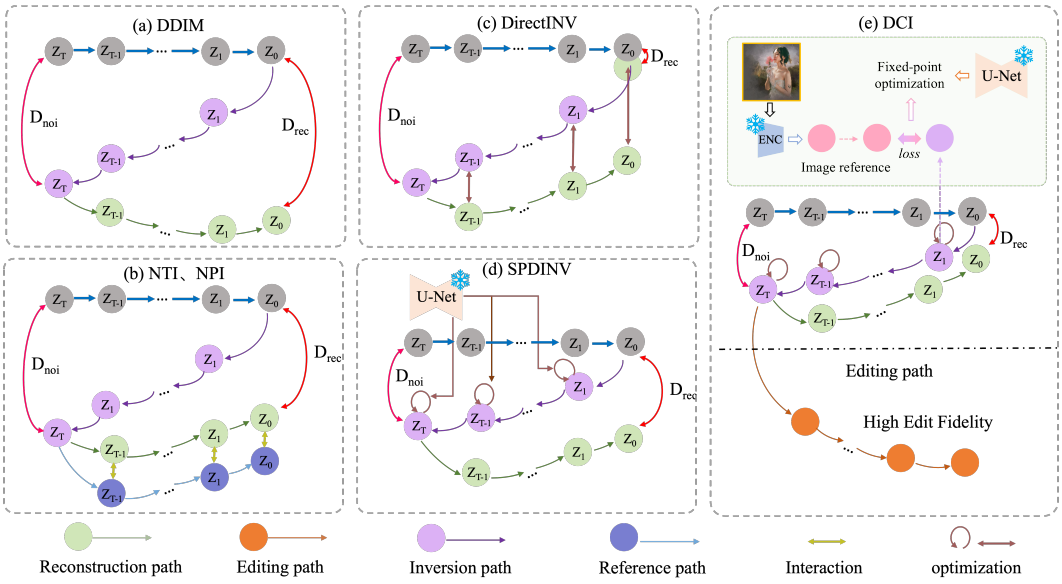

Figure 1: **Pipelines of different inversion methods in diffusion-based image editing.** Each sub-figure illustrates the specific process: (a) DDIM inversion; (b) NTI and NPI; (c) DirectInv; (d) SPDInv; (e) our Dual-Conditional Inversion(DCI). Obviously, DCI significantly reduces both latent noise gap(D_{noi}) and reconstruction error(D_{rec}).

However, in practice, especially when using a limited number of denoising steps, this assumption often breaks down, leading to significant inaccuracies in the inversion results. In order to improve the reconstruction effect of DDIM inversion, multiple works have proposed effective optimization methods, such as null-text embedding(NTI) [33] and negative prompt(NPI) [32] in the inversion process. As illustrated in figure 1, both NTI and NPI attempt to reduce the reconstruction gap(D_{rec}) by optimizing the text embeddings. In the meanwhile, the researchers have proposed some alternative solutions from a non-optimization perspective. For instance, DirectInv [20] introduces a target-aware branch to correct the source branch trajectory, improving reconstruction quality. It performs well especially in terms of content preservation, and it is faster than optimization-based inversion methods. Renoise [13] is based on the linear assumption that the direction from z_t to z_{t+1} can be approximated by the reverse direction from z_t to z_{t-1}. By calculating the direction from z_t to z_{t+1} multiple times and taking the average, a more accurate direction from z_t to z_{t+1} could be obtained. SPDInv [25] uses an optimization method to bridge the latent gap on each timestep, but the improvement of reconstruction gap (D_{rec}) is limited. Although these methods have achieved certain success, they still face an intrinsic trade-off between reconstruction accuracy and editing flexibility. As illustrated in figure 1, such approaches struggle to reconcile semantic precision with structural consistency, particularly when textual supervision is sparse or ambiguous.

In this work, we present **Dual-Conditional Inversion (DCI)**, a new perspective on diffusion-based image editing that unifies text and image conditioned inversion within a fixed-point optimization framework. DCI addresses this limitation by introducing a dual-conditioning mechanism: it jointly leverages the source prompt p_s and the reference image x_0 to guide the inversion process. At the core of our formulation is a two-stage iterative procedure. The first stage, *reference-guided noise correction*, refines the predicted noise at each timestep by anchoring it to a visually grounded reference derived from the source image. The second stage, *fixed-point latent refinement*, imposes self-consistency by optimizing each latent variable z_t as a fixed point of the generative trajectory defined by DDIM dynamics. Formally, we cast inversion as a dual-conditioned fixed-point optimization problem that minimizes two objectives: (1) the discrepancy between the predicted and reference noise vectors across timesteps, and (2) the reconstruction error between the generated image and the original reference. This formulation not only improves inversion stability but also yields latent representations that are inherently editable and semantically aligned.

To sum up, our framework enables a plug-and-play integration with a variety of existing diffusion models, requiring neither retraining nor any modification to the original model. Through extensive experiments across multiple editing tasks, DCI achieves superior reconstruction quality and editing

fidelity when compared to prior inversion baselines. Moreover, we demonstrate that the proposed dual-conditional fixed-point formulation facilitates stable convergence and generalizes well across a wide range of editing scenarios, highlighting the robustness and scalability of the proposed approach.

2 Related Work

2.1 Image Editing with Diffusion Models

In recent years, a large number of works based on diffusion models in the field of image editing demonstrate significant potential and adaptability across diverse tasks. These methods utilize diverse forms of guidance, such as text prompts, image references and segmentation maps to achieve editing objectives. [24, 22, 8, 17, 27] These advances better enable the ability to maintain editing precision and semantic consistency. The rapid development of diffusion models has significantly improved image generation capabilities. Among them, the widespread use of models such as GLIDE [36], Imagen [43], DALL·E2 [40], and Stable Diffusion(SD) [41] has gradually expanded downstream tasks based on image generation. Prompt-to-Prompt(P2P) [15] modifies cross-attention maps in diffusion models to enable text-driven image editing while preserving spatial structure through localized prompt adjustments. Pix2pix-zero [38] achieves zero-shot image-to-image translation by aligning latent features with text guidance. Plug-and-Play [48] integrates task-specific modules into pretrained diffusion backbones without retraining. MasaCtrl [4] enhances real-time spatial control in diffusion models by injecting mask-guided attention constraints for precise region-specific manipulation. IP-Adapter [56] injects visual features into the attention mechanism, enabling personalized generation without fine-tuning. ControlNet [57] introduces an auxiliary network to condition diffusion models on structural inputs like edges or poses. Some recent efforts have proposed different approaches to improve the precise of image editing from various perspectives [53, 34, 42, 21]. Despite these methods have shown promising results, they often suffer from editing failures due to inversion methods. Our DCI improves upstream inversion to enhance downstream editing fidelity.

2.2 Inversion methods of diffusion models

The earliest inversion methods include DDPM [19] and DDIM [45]. DDPM generates high-quality images by progressively adding noise in a forward process and learning the reverse denoising process. [9, 46] Building on this foundation, DDIM introduces a deterministic sampling mechanism. Its near-invertible properties provide a crucial foundation for subsequent image inversion and editing techniques. Researchers have conducted in-depth and extensive studies on the inversion process of diffusion models to achieve both efficiency and precision. Some methods focus on optimizing text embedding [33, 32, 14]. Null-Text Inversion (NTI) [33] adjusts latent encodings and text embeddings to reconstruct the original image. To improve efficiency, Negative-Prompt Inversion (NPI) [32] and its enhancements, including Proximal Guidance [14], have emerged to reduce the reliance on time-consuming optimization processes. EDICT [49], for example, achieves exact invertibility through coupling transformations, while methods like Direct Inversion [20] and Fixed-Point Inversion [30] focus on simplifying the inversion process. The former decouples the diffusion branches, while the latter utilizes fixed-point iteration theory to ensure high reconstruction quality while reducing computational overhead. Many inversion techniques also particularly focus on improving downstream editing tasks [25, 11]. For example, Source Prompt Disentangled Inversion (SPDInv) [25] aims to decouple image content from the original text prompt, enhancing editing flexibility and accuracy. Specialized inversion and editing frameworks have been developed for specific editing needs [26, 44]. Additionally, the concept of inversion has been extended to broader domains [12, 18, 11, 6, 59]. Textual Inversion proposes learning new text embeddings to represent user-specific concepts for personalized image generation [12]. ReVersion [18] further explores learning and inverting relational concepts from images. Meanwhile, works like Aligning Diffusion Inversion Chain [59] focus on generating high-quality image variants by aligning inversion chains.

Although the above methods have solved the reconstruction problem to a certain extent, they may bring artifacts and inconsistent details when applied to editing tasks. Most of the time, they only focus on the text prompt or the original image, but do not integrate them. In our work, we propose a simple but effective method to fuse the text prompt and source image in the form of fixed-point iteration. Our method improves the editing fidelity a lot and shows inspiring results.

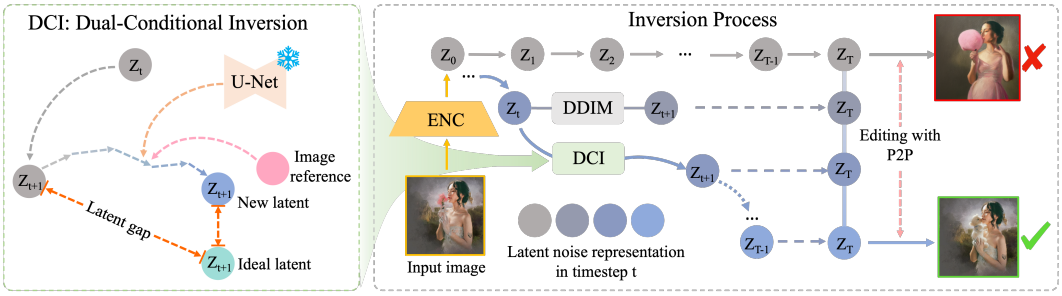

Figure 2: **Inversion process of DCI**. The green box on the left illustrates DCI, which use dual-conditional guidance to reduce the latent gap. The right describes how DCI modifies the inversion process and generate the latent noise code. It also shows our method can improve the editing method.

3 Dual-Conditional Inversion

3.1 Motivation and Problem Formulation

In most diffusion-based image editing frameworks, the inversion process plays a foundational role: it converts an image to the latent noise representation from which the image can be reconstructed and edited. However, diffusion models inherently lack an explicit and exact inverse process to convert an image back to its corresponding latent noise representation. Ideally, a successful inversion would yield a latent code z_T that faithfully preserves both the semantic content and structural details of the input image, thereby enabling accurate reconstruction and precise downstream editing. However, the information loss caused by repeated noise injection in inversion process makes perfect inversion unattainable, even when auxiliary constraints such as text prompts or reference images are employed.

To analyze the limitations of current inversion strategies, we begin with DDIM (Denoising Diffusion Implicit Models) [45], a deterministic variant of DDPM [16]. DDIM defines a closed-form sampling process that generates a latent image z_0 from Gaussian noise $z_T \sim \mathcal{N}(0, \mathbf{I})$ as follows:

$$z_{t-1} = \frac{\sqrt{\alpha_{t-1}}}{\sqrt{\alpha_t}} z_t + \sqrt{\alpha_{t-1}} \left(\sqrt{\frac{1}{\alpha_{t-1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1} \right) \epsilon_\theta(z_t, t, c), \tag{1}$$

Where α_t denotes the cumulative noise schedule, and ϵ_θ represents the noise predicted by a U-Net, conditioned on the current timestep t and a control input c (e.g. , a text prompt). However, using only a text prompt as c is insufficient for accurately reconstructing the original image. Recent methods such as ControlNet [57] and IP-Adapter [56] enrich the conditioning input c with visual features from the original image, thereby improving generation quality. Nevertheless, these methods are often computationally expensive and difficult to integrate into the inversion process. Ideally, inversion requires recovering z_t from a known z_{t-1}, which leads to the following "ideal inversion" formula:

$$z_t = C_{t,1} \cdot z_{t-1} + C_{t,2} \cdot \epsilon_\theta(z_t, t, c_{\text{ideal}}), \tag{2}$$

where the coefficients are defined as: $C_{t,1} = \frac{\sqrt{\alpha_t}}{\sqrt{\alpha_{t-1}}}$, $C_{t,2} = \sqrt{\alpha_t} \left(\sqrt{\frac{1}{\alpha_t} - 1} - \sqrt{\frac{1}{\alpha_{t-1}} - 1} \right)$.

However, in practice, this expected inversion is not feasible because the ideal latent z_t is not available when performing the inversion step from z_{t-1}. Thus, the DDIM inversion process approximates this update by feeding $(z_{t-1}, t-1, c)$ into the inversion process instead of (z_t, t, c), leading to the practical inversion formula:

$$z_t = C_{t,1} \cdot z_{t-1} + C_{t,2} \cdot \epsilon_\theta(z_{t-1}, t-1, c). \tag{3}$$

This approximation breaks the strict reversibility of the ODE-based formulation and introduces temporal mismatch error between the predicted noise and the actual generative trajectory. Since the diffusion model assumes infinitesimal step size for reversibility (akin to a continuous ODE), using coarse discrete steps and mismatched inputs (i.e., $\epsilon_\theta(z_{t-1}, t-1, c)$ instead of the ideal $\epsilon_\theta(z_t, t, c)$) induces systematic error at each timestep.

If a real image and its corresponding text prompt are given, the image generated directly using the text prompt will be very different from the real image. The reason arises from the inaccuracy of text

prompt and randomness in the generation process. From this perspective, there are also errors in the use of $\epsilon_\theta(z_t, t, c))$ for the inversion process. This error is also accumulated over time, resulting in the final z_t not being well applied to reconstruction and editing. In previous work, SPDInv [25] transforms the inversion process into a search problem that satisfies fixed-point constraints. The pre-trained diffusion model is used to make the inversion process as independent of the source prompt as possible, thereby reducing the gap between $\epsilon_\theta(z_t, t, c))$ and $\epsilon_\theta(z_{t-1}, t-1, c)$. Although SPDInv narrows the gap between $\epsilon_\theta(z_t, t, c))$ and $\epsilon_\theta(z_{t-1}, t-1, c)$. However, in the previous analysis, $\epsilon_\theta(z_t, t, c))$ is not an ideal noise. The ideal noise should not only be separated from the source prompt, but also retain more information of the original image. What needs to be reduced is the difference between $\epsilon_\theta(z_t, t, c_{ideal})$ and $\epsilon_\theta(z_{t-1}, t-1, c)$, and this difference will appear in each inversion process and accumulate in the final output.

To achieve high-fidelity inversion, it is essential to minimize the discrepancy between the predicted noise and the ideal generative direction at each timestep. This requires not only disentangling the inversion process from the source prompt(mentioned in [25]), but also preserving as much information from the original image as possible. Addressing both aspects simultaneously is key to reducing cumulative errors and improving the reconstruction and editability of the inverted latent noise representations in diffusion-based image editing.

3.2 Dual-Conditional Inversion (DCI)

To address the limitations of existing inversion methods, we propose Dual-Conditional Inversion (DCI), a novel framework that enhances the latent noise representations in diffusion models. DCI leverages both the original image and text prompt to guide the inversion process, ensuring high-fidelity reconstruction and improved editability. Unlike prior approaches, DCI integrates these into a dual-conditional fixed-point optimization pipeline. The method consists of two key stages: *reference-guided noise correction* that anchors the inversion to the source image, and *fixed-point latent refinement* that ensures self-consistency with the generative process.

3.2.1 Reference-Guided Noise Correction

The first stage of DCI introduces a reference-based constraint to align the predicted noise with the source image. At each DDIM timestep t, we compute an initial noise estimate conditioned on the source prompt p_s:

$$\hat{\epsilon}_{\text{raw}} = \epsilon_\theta(z_t, t, p_s). \tag{4}$$

where ϵ_θ is the noise prediction model (e.g., a U-Net) and z_t is the current latent. However, $\hat{\epsilon}_{\text{raw}}$ often deviates from the ideal noise due to the coarse constraint of p_s. While this prediction reflects prompt-level semantics, it often deviates from the actual noise corresponding to the input image due to limited grounding provided by textual information alone. To address this, we introduce a visual reference signal by extracting a reference noise vector ϵ_{ref} from the source image latent z_0, which is obtained via a pretrained VAE encoder E. The reference noise is defined as:

$$\epsilon_{\text{ref}} = E(z_0). \tag{5}$$

The ϵ_{ref} serves as an anchor to guide the correction of prompt-based noise estimation. To enforce alignment between the prompt-predicted noise and the image-derived reference, we define a reference alignment loss:

$$\mathcal{L}_{\text{ref}} = \left\| \hat{\epsilon}_{\text{raw}} - \epsilon_{\text{ref}} \right\|_2. \tag{6}$$

Equation 6 penalizes the discrepancy between the two noise vectors. A one-step gradient-based correction is then applied to refine the noise prediction:

$$\hat{\epsilon} = \hat{\epsilon}_{\text{raw}} - \lambda \cdot \nabla_{\hat{\epsilon}_{\text{raw}}} \mathcal{L}_{\text{ref}}. \tag{7}$$

where λ is a hyperparameter that controls the correction strength. This update adjusts the predicted noise in a direction that reduces its divergence from the reference signal, effectively grounding the inversion in visual structure. As a result, this correction improves reconstruction fidelity and ensures that the denoising trajectory remains semantically and perceptually consistent with the original image, particularly in scenarios where the prompt is ambiguous or underspecified.

3.2.2 Fixed-Point Latent Refinement

After correcting the noise estimate, we proceed to update the latent variable z_t using the DDIM inversion formula. This step changes the inversion trajectory from timestep $t-1$ to t, based on the corrected noise $\hat{\epsilon}$:

$$z_t = C_{t,1} \cdot z_{t-1} + C_{t,2} \cdot \hat{\epsilon}, \tag{8}$$

where $C_{t,1} = \frac{\sqrt{\alpha_t}}{\sqrt{\alpha_{t-1}}}$ and $C_{t,2} = \sqrt{\alpha_t}\left(\sqrt{\frac{1}{\alpha_t}-1} - \sqrt{\frac{1}{\alpha_{t-1}}-1}\right)$, and α_t is the noise schedule. While this deterministic update follows the DDIM trajectory, it remains sensitive to error accumulation during the inversion process. As such, it may introduce perturbations into the latent dynamics, ultimately affecting reconstruction and editing fidelity. To improve stability and enforce consistency with the forward generative process, DCI introduces a fixed-point refinement step that iteratively corrects the latent by treating it as a fixed-point problem of the DDIM inversion at each timestep. Specifically, we define the latent update function:

$$f_\theta(z_t) = C_{t,1} \cdot z_{t-1} + C_{t,2} \cdot \epsilon_\theta(z_t, t, p_s). \tag{9}$$

The objective is to find a latent z_t such that:

$$z_t = f_\theta(z_t). \tag{10}$$

To achieve this, we minimize the following fixed-point self-consistency loss:

$$\mathcal{L}_{\text{fix}} = \left\| f_\theta(z_t) - z_t \right\|_2 \tag{11}$$

We iteratively refine z_t using gradient descent:

$$z_t = z_t - \eta \cdot \nabla_{z_t} \mathcal{L}_{\text{fix}}, \tag{12}$$

where η is the learning rate of refinement process. This fixed-point update step is repeated for up to K iterations or until the convergence criterion $\mathcal{L}_{\text{fix}} < \delta$ is satisfied. In practice, our method converges rapidly within a few iterations(usually no more than 10 iterations), which ensures computational efficiency without compromising reconstruction quality. By explicitly enforcing this self-consistency constraint, DCI stabilizes the inversion trajectory and reduces artifacts that arise from misaligned latents. This refinement step not only enhances reconstruction quality but also improves the reliability and flexibility of downstream editing operations.

Algorithm 1 Dual-Conditional Inversion (DCI)

Input: Source image latent z_0, DDIM steps T, source prompt p_s, maximal optimization rounds K, threshold δ, image guidance strength λ, fixed-point learning rate η, reference noise ϵ_{ref}
Output: Inversion noise z_T

for $t = 1$ to T **do**
  for $i = 1$ to K **do**
    Get z_t from z_{t-1} based on (3)
    Predict noise $\hat{\epsilon}_{\text{raw}}$ based on (4)
    Compute $\mathcal{L}_{\text{ref}} = \left\| \hat{\epsilon}_{\text{raw}} - \epsilon_{\text{ref}} \right\|_2$
    Apply correction: $\hat{\epsilon} = \hat{\epsilon}_{\text{raw}} - \lambda \cdot \nabla_{\hat{\epsilon}_{\text{raw}}} \mathcal{L}_{\text{ref}}$
    Update z_t using $\hat{\epsilon}$
    Calculate $\mathcal{L}_{\text{fix}} = \left\| f_\theta(z_t) - z_t \right\|_2$
    Update $z_t = z_t - \eta \cdot \nabla_{z_t} \mathcal{L}_{\text{fix}}$
    if $\mathcal{L}_{\text{fix}} < \delta$ *then break* **end if**
  end for
end for

3.2.3 Algorithm Summary

The complete Dual-Conditional Inversion (DCI) process is summarized in Algorithm 1. At each DDIM timestep, DCI first performs *Reference-Guided Noise Correction* to obtain a visually grounded noise estimate $\hat{\epsilon}$ by combining prompt-based prediction and reference-derived supervision. Then it is followed by *Fixed-Point Latent Refinement*, which iteratively updates the latent z_t to satisfy a

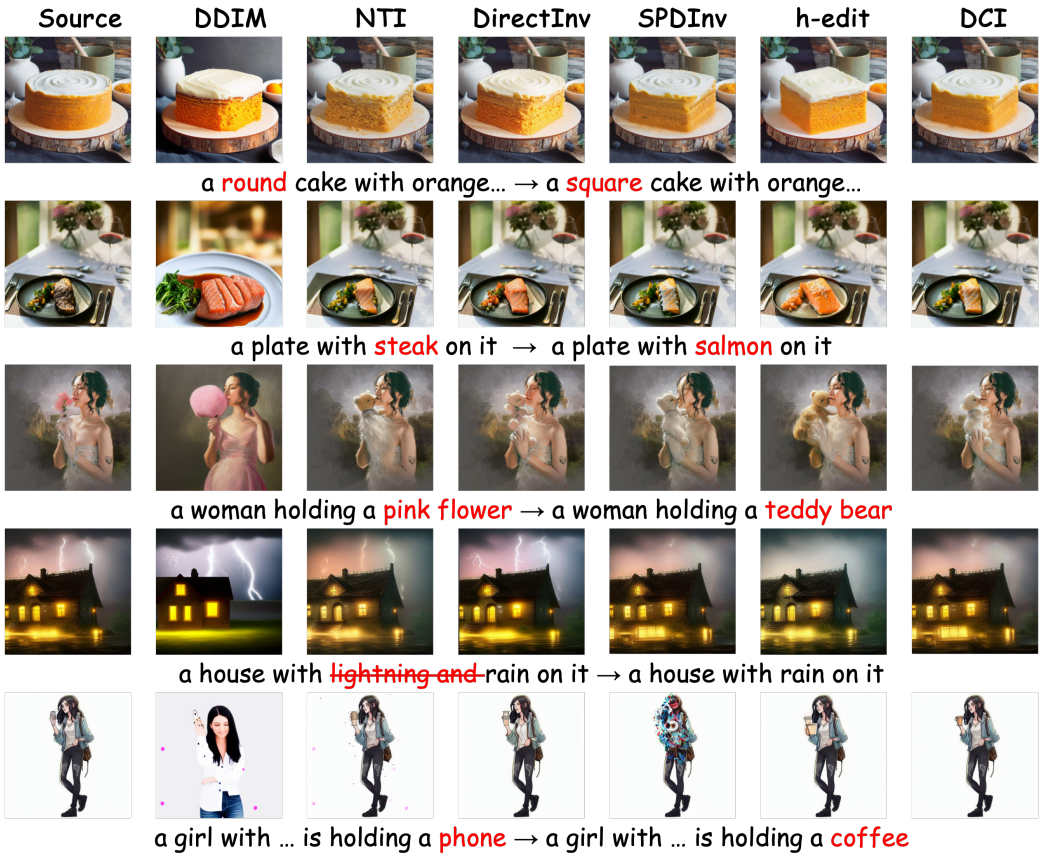

Figure 3: **Visual results of different inversion methods with P2P on PIE-Bench.** Each method is identified at the top of its respective column, while detailed editing information appears beneath each corresponding row. DCI(ours) demonstrates significant enhancements over existing methods.

self-consistency condition defined by the DDIM inversion dynamics. The dual conditioning on both the source prompt p_s and the reference image (via ϵ_{ref}) ensures that the final inverted latent z_T closely approximates the ideal generative noise z_T^*, which leads to reliable reconstruction and high-fidelity, better structure-preserving editing.

4 Experiments

We conduct extensive experiments to evaluate the effectiveness of Dual-Conditional Inversion (DCI). This section is organized as follows. In Section 4.1, we introduce the datasets, evaluation metrics and experimental settings. Section 4.2 compares DCI with representative inversion methods across multiple aspects quantitatively and qualitatively. In Section 4.3, we investigate how DCI reduces both the latent noise gap and the reconstruction error. Finally, Section 4.4 presents an ablation study to assess the impact of key hyperparameters and design choices.

4.1 Experimental Setups

Evaluation Metrics. We mainly use DINO score [5], Peak Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) [58] to evaluate the performance of DCI from multiple perspectives. We use the DINO score to evaluate the overall structural similarity of the generated images, while the CLIP score [39] is employed to quantify the alignment between the generated image and the given prompt. For background preservation and image fidelity, we report PSNR, MSE, SSIM, and LPIPS, with all metrics computed specifically over the annotated regions in the dataset from DirecInv [20]. Both the DINO and CLIP scores are calculated over the entire image to capture global consistency, whereas the remaining metrics focus on local quality within specified regions.

Table 1: Performance comparison of inversion-based methods under the Prompt-to-Prompt (P2P) editing engine [11] on PIE-Bench. Metrics include DINO (↓), PSNR (↑), LPIPS (↓), MSE (↓), SSIM (↑), and CLIP (↑). Best and second-best results are highlighted in red and blue, respectively. DCI (ours) achieves the best performance across all metrics.

Inversion	Editing Engine	DINO↓ $\times 10^3$	PSNR↑	LPIPS↓ $\times 10^3$	MSE↓ $\times 10^4$	SSIM↑ $\times 10^2$	CLIP↑
DDIM [45]	P2P	69.43	17.87	208.80	219.88	71.14	25.01
NTI [33]	P2P	13.44	27.03	60.67	35.86	84.11	24.75
NPI [32]	P2P	16.17	26.21	69.01	39.73	83.40	24.61
AIDI [37]	P2P	12.16	27.01	56.39	36.90	84.27	24.92
NMG [6]	P2P	23.50	25.83	81.58	107.95	82.31	24.05
DirectINV [20]	P2P	11.65	27.22	54.55	32.86	84.76	25.02
ProxEdit [14]	P2P	11.87	27.12	45.70	32.16	84.80	24.28
SPDInv [25]	P2P	8.81	28.60	36.01	24.54	86.23	25.26
h-Edit [35]	P2P	11.17	27.87	48.50	85.40	84.80	25.30
DCI(ours)	P2P	6.07	29.38	33.01	21.28	87.14	25.52

Datasets. We verifies the effectiveness of our proposed DCI method mainly on the PIE-Bench [20], which comprises 700 images featuring 10 distinct editing types. It provides five annotations on each image: source image prompt, target image prompt, editing instruction, main editing body, and the editing mask. The calculation of region-specific metrics heavily relies on the editing mask, as the editing is expected to occur only within the annotated region. We also use the *COCO2017* [28] to test the application of our method in a wider range of scenarios.

Other Settings. In our experiments, we utilize Stable Diffusion v1.4 as the base model with DDIM sampling steps of 50 and a Classifier-Free Guidance (CFG) scale of 7.5. These settings are the same as those used in the baselines. For DCI, we set the hyper-parameters to $K = 5$, $\lambda = 2$, and $\eta = 0.001$. All experiments and validations are conducted on a single NVIDIA RTX 4090 GPU.

4.2 Comparisons with Inversion-Based Editing Methods

We compare DCI with several inversion-based methods quantitatively and qualitatively. These methods includes DDIM inversion [45], Null-text inversion (NTI) [33], Negative prompt inversion (NPI) [32], AIDI [37], Noise Map Guidance (NMG) [6], Direct Inversion (DirectINV) [20], ProxEdit [14], SPDInv [25] and *h*-Edit [35]. We mainly evaluate under the Prompt-to-Prompt (P2P) editing engine on PIE-Bench. As Table 4 shows, DCI (ours) outperforms all methods across DINO, PSNR, LPIPS, MSE, SSIM, and CLIP metrics. Compared to the second-best method, SPDInv, DCI achieves significant improvements, including a 31.1% reduction in DINO (6.07vs.8.81), 8.3% reduction in LPIPS (33.01vs.36.01), and 13.3% reduction in MSE (21.28vs.24.54). At the same time, It is also higher than SPDInv in other metrics(PSNR,SSIM,CLIP). Compared with other methods listed in Table 4, our method has a greater improvement. These results underscore DCI's superior accuracy and robustness for high-fidelity image editing.

Figure 3 presents a visual comparison with the P2P engine. The first row presents cake images frequently used for comparative analysis in existing methods. Most approaches show satisfactory results. In contrast, the second row demonstrates that our method enhances detail representation in salmon. The third row illustrates when modifying features such as hands or mouth, previous methods will fail. However, our DCI achieves this task while maintaining high-quality output. In the fourth row, our method achieves better background color fidelity and reduces lighting artifacts compared to others. The fifth row highlights our method's robust performance in local part editing while preserving overall consistency across other image regions.

Human Preference Results For image editing task, human preference is an important part of evaluation metrics. We provide table 2 detailing both user study and human preferences metrics to demonstrate the effectiveness of our approach. For the user study, we collect 40 comparisons from 25 participants (aged 19 to 50). The table shows the mean scores for each participant (min:1, max:5). For human preferences metrics, ImageReward [55] and PickScore [23] are human preference–based

reward models to quantitatively evaluate the quality of image generation and editing. Both of them are higher metrics that represent better performance.

Table 2: Human preference results

	DDIM	SPDInv	DCI
Pickscore [23]	0.4416	0.4954	**0.5547**
ImageReward [55]	-0.0120	0.1564	**0.3674**
User	2.14	3.48	**4.10**

Time Consumption The running time of DCI is tied to the number of optimization iterations and the error threshold. However, since our method primarily focuses on nudging the inversion process back onto the correct path, we've empirically found that often only a few optimization steps at specific timesteps are needed to achieve significant improvements. As a result, the additional computational overhead compared with DDIM remains minimal.

Table 3: Comparison of inversion times (in seconds) across different methods.

	DDIM	NTI	NPI	AIDI	DirectINV	SPDInv	DCI(ours)
Time(s)	11.55	137.54	11.75	87.21	19.94	27.04	12.13

Results under different editing engines and base models. Other editing engines and other architectures of diffusion models can also be adopted for our DCI. We use LEDITS++ [2] as the multi-subject editing model and apply it with both DDIM and our DCI method for fair comparison. In our paper, the reported MSE is calculated for non-edited regions, thanks to the availability of appropriate mask annotations within the dataset. However, for multi-subject editing, we could only calculate the MSE between the edited image and the entire original image. Under these conditions, the MSE metric is not always an accurate reflection of editing quality, as DDIM frequently fails to produce any changes or only generates very minor alterations. Beyond multi-subject editing, we also test our method with Stable-Flow, a flow-based diffusion image editing method. The experimental results clearly indicate that our approach significantly enhances performance in flow-based methods as well.

Table 4: Performance under different editing engines and base models.

Inversion	Editing Engine	DINO↓ $\times 10^3$	PSNR↑	LPIPS↓ $\times 10^3$	MSE↓ $\times 10^4$	SSIM↑ $\times 10^2$	CLIP↑
DDIM [45]	LEDITS++ [2]	21.20	21.18	136.3	**76.00**	83.95	19.01
DCI(ours)	LEDITS++ [2]	**12.10**	**21.19**	**127.5**	125.00	**84.41**	**21.62**
DDIM [45]	Stable-Flow [1]	19.00	24.30	91.70	**37.00**	91.60	23.29
DCI(ours)	Stable-Flow [1]	**4.40**	**24.32**	**68.40**	**37.00**	**92.75**	**23.64**

Due to the page limit, we provide more visual and quantitative results under different editing engines in the **supplementary material**. We can draw similar conclusions to the above.

4.3 Reduction of Noise and Reconstruction Gap by DCI

We conduct experiments and confirm that our method can reduce the gap between noise and reconstruction (D_{noi} and D_{rec} as depicted in Figure 1). We randomly select 100 captions from the PIE-Bench and use Stable Diffusion V1.4 to generate images.

We initialize z_T with a fixed random seed, treating it as the ideal noise input for every image at the initial timestep of the diffusion process. The final generated image serves as a reference for reconstruction accuracy assessment. We visualize and evaluate the performance of our method with DDIM [45] and SPDInv [25].

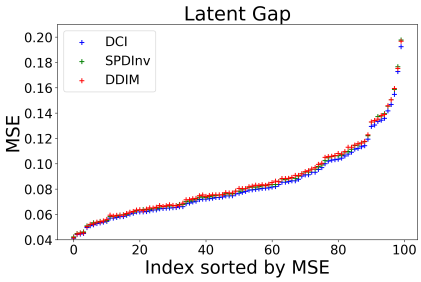

Figure 4: Illustration of Latent Gap.

Table 5: Ablation study on the hyper-parameters of DCI with PIE-Bench.

Hyper-parameter	DINO$_{\times 10^3}$ ↓	PSNR↑	LPIPS$_{\times 10^3}$ ↓	MSE$_{\times 10^4}$ ↓	SSIM$_{\times 10^2}$ ↑	CLIP↑
$K = 2$	6.13	29.32	33.10	21.50	87.11	25.49
$K = 5$	6.07	29.38	33.01	21.28	87.14	25.52
$K = 10$	6.17	29.29	33.17	21.56	87.12	25.51
$\lambda = 1$	6.19	29.29	33.12	21.62	87.12	25.53
$\lambda = 2$	6.07	29.38	33.01	21.28	87.14	25.52
$\lambda = 5$	9.29	28.25	41.05	26.18	86.26	25.38
$\eta = 0.0001$	6.72	28.80	35.93	23.72	86.70	25.50
$\eta = 0.001$	6.07	29.38	33.01	21.28	87.14	25.52
$\eta = 0.01$	35.29	23.05	88.90	83.84	81.18	25.02
Default	**6.07**	**29.38**	**33.01**	**21.28**	**87.14**	**25.52**

For latent gap analysis, we visualize the z_T gap obtained by these methods in figure 4. The concentration of the data shows that our method is closer to the ideal noise. For reconstruction gap evaluation, we use both MSE and CLIP scores. DDIM yields an MSE of 1.32×10^{-4}, SPDInv achieves 1.21×10^{-4}, while DCI obtains the lowest error at 1.12×10^{-4}. The CLIP Scores are 26.91 for DDIM, 26.92 for SPDInv, and 26.94 for DCI. Comparatively, our technique demonstrates superior performance over DDIM and SPDInv based on these metrics.

4.4 Ablation Study

Table 5 presents an ablation study on three key hyper-parameters of DCI: the number of optimization rounds ($K \in \{2, 5, 10\}$), the reference-guided noise correction weight ($\lambda \in \{1, 2, 5\}$), and the learning rate ($\eta \in \{0.0001, 0.001, 0.01\}$). The method converges quickly, as even a small number of rounds ($K = 2$) shows competitive results, and performance saturates by $K = 5$. $\lambda = 2$ achieves the best trade-off, while higher values such as $\lambda = 5$ lead to significant degradation across all metrics, indicating over-dependence on inversion constraints. The learning rate $\eta = 0.001$ provides the most stable and effective optimization; both smaller and larger values reduce reconstruction quality, with $\eta = 0.01$ causing severe performance collapse. These results support the choice of the default configuration ($K = 5$, $\lambda = 2$, $\eta = 0.001$) as optimal for fidelity and stability.

5 Conclusion

In this paper, we introduce Dual-Conditional Inversion (DCI), a novel method that combines both the source prompt and the reference image to guide the inversion process. By formulating inversion as a dual-conditioned fixed-point optimization problem, DCI reduces both latent noise gap and reconstruction errors in diffusion models. Notably, DCI exhibits strong plug-and-play capability: it can be seamlessly integrated into existing diffusion-based editing pipelines without requiring model retraining or architecture modification. Extensive experiments demonstrate that our method achieves superior edit quality on benchmark datasets. Overall, DCI provides a robust, flexible, and easily deployable foundation for future research in diffusion-based tasks.

6 Acknowledgements

This research is supported by the National Natural Science Foundation of China (62120106009, 62372033, U24B20179, 92470203, U23A20314), Natural Science Foundation of Beijing, China(No.L252025, No. L242022) and the Fundamental Research Funds for the Central Universities (No. 2024XKRC082).

References

[1] 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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7877–7888, 2025.

[2] Manuel Brack, Felix Friedrich, Katharina Kornmeier, Linoy Tsaban, P. Schramowski, K. Kersting, and Apolin'ario Passos. Ledits++: Limitless image editing using text-to-image models. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8861–8870, 2023.

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

[4] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. *arXiv preprint arXiv:2304.08465*, 2023.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[6] Hansam Cho, Jonghyun Lee, Seoung Bum Kim, Tae-Hyun Oh, and Yonghyun Jeong. Noise map guidance: Inversion with spatial context for real image editing. *arXiv preprint arXiv:2402.04625*, 2024.

[7] Jiwoo Chung, Sangeek Hyun, and Jae-Pil Heo. Style injection in diffusion: A training-free approach for adapting large-scale diffusion models for style transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8795–8805, 2024.

[8] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022.

[9] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:8780–8794, 2021.

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

[12] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

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

[14] Ligong Han, Song Wen, Qi Chen, Zhixing Zhang, Kunpeng Song, Mengwei Ren, Ruijiang Gao, Yuxiao Chen, Di Liu, Qilong Zhangli, et al. Improving negative-prompt inversion via proximal guidance. *arXiv preprint arXiv:2306.05414*, 2023.

[15] 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.

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[17] Nisha Huang, Fan Tang, Weiming Dong, Tong-Yee Lee, and Changsheng Xu. Region-aware diffusion for zero-shot text-driven image editing. *arXiv preprint arXiv:2302.11797*, 2023.

[18] Ziqi Huang, Tianxing Wu, Yuming Jiang, Kelvin CK Chan, and Ziwei Liu. Reversion: Diffusion-based relation inversion from images. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

[19] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. *arXiv preprint arXiv:2304.06140*, 2023.

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

[21] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Hui-Wen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[22] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2426–2435, 2022.

[23] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

[24] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. *arXiv preprint arXiv:2210.10960*, 2022.

[25] Ruibin Li, Ruihuang Li, Song Guo, and Lei Zhang. Source prompt disentangled inversion for boosting image editability with diffusion models. In *European Conference on Computer Vision*, pages 404–421. Springer, 2024.

[26] 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. *arXiv preprint arXiv:2303.15649*, 2023.

[27] Zixiang Li, Yue Song, Renshuai Tao, Xiaohong Jia, Yao Zhao, and Wei Wang. Unsupervised region-based image editing of denoising diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18638–18646, 2025.

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014.

[29] Xinqi Lin, Jingwen He, Ziyan Chen, Zhaoyang Lyu, Bo Dai, Fanghua Yu, Yu Qiao, Wanli Ouyang, and Chao Dong. Diffbir: Toward blind image restoration with generative diffusion prior. In *European Conference on Computer Vision*, pages 430–448. Springer, 2024.

[30] Barak Meiri, Dvir Samuel, Nir Darshan, Gal Chechik, Shai Avidan, and Rami Ben-Ari. Fixed-point inversion for text-to-image diffusion models. *CoRR*, 2023.

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

[32] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. *arXiv preprint arXiv:2305.16807*, 2023.

[33] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023.

[34] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Diffeditor: Boosting accuracy and flexibility on diffusion-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8488–8497, 2024.

[35] Toan Nguyen, Kien Do, Duc Kieu, and Thin Nguyen. h-edit: Effective and flexible diffusion-based editing via doob's h-transform. *arXiv preprint arXiv:2503.02187*, 2025.

[36] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[37] Zhihong Pan, Riccardo Gherardi, Xiufeng Xie, and Stephen Huang. Effective real image editing with accelerated iterative diffusion inversion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15912–15921, 2023.

[38] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–11, 2023.

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

[40] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

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

[42] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023.

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

[44] Yujun Shi, Chuhui Xue, Jun Hao Liew, Jiachun Pan, Hanshu Yan, Wenqing Zhang, Vincent YF Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8839–8849, 2024.

[45] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020.

[46] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[47] Linoy Tsaban and Apolinário Passos. Ledits: Real image editing with ddpm inversion and semantic guidance. *arXiv preprint arXiv:2307.00522*, 2023.

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

[49] Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22532–22541, 2023.

[50] Haofan Wang, Matteo Spinelli, Qixun Wang, Xu Bai, Zekui Qin, and Anthony Chen. Instantstyle: Free lunch towards style-preserving in text-to-image generation. *arXiv preprint arXiv:2404.02733*, 2024.

[51] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *International Journal of Computer Vision*, 132(12):5929–5949, 2024.

[52] Zhizhong Wang, Lei Zhao, and Wei Xing. Stylediffusion: Controllable disentangled style transfer via diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7677–7689, 2023.

[53] Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15943–15953, 2023.

[54] Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. One-step effective diffusion network for real-world image super-resolution. *Advances in Neural Information Processing Systems*, 37:92529–92553, 2024.

[55] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

[56] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

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

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

[59] Yuechen Zhang, Jinbo Xing, Eric Lo, and Jiaya Jia. Real-world image variation by aligning diffusion inversion chain. *Advances in Neural Information Processing Systems*, 36:30641–30661, 2023.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction summarize the core contributions behind Dual-Conditional Inversion (DCI), the methodological innovation of incorporating both source prompt and image reference as dual conditions, and the demonstrated improvements in both reconstruction and editing performance. They accurately reflect the paper's described scope and contributions.

   Guidelines:

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

2. **Limitations**

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

   Answer: [No]

   Justification: Due to page limit, we present this part in supplementary materials.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: The cited literature and experimental results fully confirm our theoretical hypothesis and proof.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We disclosed all of the main experimental results in the part of the experiments.

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: The data and code of our paper are convenient for reproducing the experimental results.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We demonstrated all the training and test details.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Due to the resource limitation, we do not report error bars. Our experiments on models and hyperparameters are so numerous that they are impossible to repeat.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: In the experimental section, we declare the setups.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: All experiments comply with ethical standards.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Our work already discusses the impacts of society.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: Our work carries no risk.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Our work is based on public datasets and code.

    Guidelines:

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

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [Yes]

    Justification: We have an human-preference experiment.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [No]

    Justification: We have no crowdsourcing or research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: In this study, the LLM is not the target of this paper.

Guidelines:

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

In this supplementary file, we provide the following materials:

- **Proof of Dual-Conditional Inversion**
- **Experimental Results on Different Datasets and Editing Engines**
- **Inversion Time Comparison**
- **Failure Cases**
- **Limitations and Future Work**

A Proof of Dual-Conditional Inversion

A.1 Stage-Wise Convergence Analysis

The Dual-Conditional Inversion (DCI) algorithm operates in two alternating stages: (1) reference-guided noise correction and (2) fixed-point latent refinement. Since these two stages target separate objective functions and update different variables ($\hat{\epsilon}_{\text{raw}}$ and z_t, respectively), their convergence must be analyzed independently.

A.1.1 Stage 1: Reference-Guided Noise Correction

This stage minimizes the noise alignment loss:

$$\mathcal{L}_{\text{ref}}(\hat{\epsilon}_{\text{raw}}) = \|\hat{\epsilon}_{\text{raw}} - \epsilon_{\text{ref}}\|_2^2 , \tag{13}$$

via a one-step gradient descent update:

$$\hat{\epsilon} = \hat{\epsilon}_{\text{raw}} - \lambda \cdot \nabla_{\hat{\epsilon}_{\text{raw}}} \mathcal{L}_{\text{ref}} = (1-\lambda)\hat{\epsilon}_{\text{raw}} + \lambda \epsilon_{\text{ref}}. \tag{14}$$

This is equivalent to a convex interpolation between $\hat{\epsilon}_{\text{raw}}$ and ϵ_{ref}, and always satisfies:

$$\mathcal{L}_{\text{ref}}(\hat{\epsilon}) \leq \mathcal{L}_{\text{ref}}(\hat{\epsilon}_{\text{raw}}), \quad \forall \lambda \in (0, 1]. \tag{15}$$

Hence, the correction step is guaranteed to reduce noise misalignment in a single iteration, and can be seen as a contractive projection in noise space.

A.1.2 Stage 2: Fixed-Point Latent Refinement

Given the corrected noise $\hat{\epsilon}$, we update the latent z_t using the DDIM inversion formula:

$$z_t^{(0)} = C_{t,1} z_{t-1} + C_{t,2} \cdot \hat{\epsilon}. \tag{16}$$

Then, we refine z_t by minimizing the self-consistency fixed-point loss:

$$\mathcal{L}_{\text{fix}}(z_t) = \|f_\theta(z_t) - z_t\|_2^2 , \quad \text{where} \quad f_\theta(z_t) := C_{t,1} z_{t-1} + C_{t,2} \cdot \epsilon_\theta(z_t, t, p_s). \tag{17}$$

We apply gradient descent:

$$z_t^{(k+1)} = z_t^{(k)} - \eta \cdot \nabla \mathcal{L}_{\text{fix}}(z_t^{(k)}). \tag{18}$$

Assume:

- $\epsilon_\theta(z)$ is L-Lipschitz smooth;
- $f_\theta(z)$ is locally contractive near the solution;
- $\mathcal{L}_{\text{fix}}$ is bounded below.

Then from smooth non-convex optimization theory:

Theorem A.1 (Local Convergence of Fixed-Point Refinement)**.** *The sequence* $\{z_t^{(k)}\}$ *satisfies:*

$$\min_{0 \leq k < K} \left\| \nabla \mathcal{L}_{\textit{fix}}(z_t^{(k)}) \right\|^2 \leq \frac{2(\mathcal{L}_{\textit{fix}}(z_t^{(0)}) - \mathcal{L}_{\min})}{\eta K}, \tag{19}$$

and converges to a stationary point z_t^* *as* $K \to \infty$.

A.1.3 Alternating Convergence

In DCI, the noise correction step anchors the predicted noise toward a semantically and visually meaningful reference, ensuring that the initialization for the latent refinement step falls within the contraction region of $f_\theta(z)$. Thus, the two-stage alternating optimization benefits from mutual regularization:

- Stage 1 reduces semantic deviation from image-derived noise;
- Stage 2 reduces structural inconsistency via fixed-point updates;

This design avoids the need to jointly optimize a non-separable loss and enables fast convergence with high reconstruction fidelity.

A.1.4 Additional Analysis

Beyond convergence guarantees, we analyze several critical properties of the two-stage optimization process to better understand the behavior of DCI in practice.

Stability of the Correction Step. The reference-guided noise correction step performs a convex interpolation between $\hat{\epsilon}_{\text{raw}}$ and ϵ_{ref}, ensuring that the corrected noise $\hat{\epsilon}$ remains within the convex hull of the input and the reference:

$$\hat{\epsilon} \in \text{Conv}\left(\hat{\epsilon}_{\text{raw}}, \epsilon_{\text{ref}}\right). \tag{20}$$

This guarantees bounded updates and avoids divergence, even when $\hat{\epsilon}_{\text{raw}}$ contains large errors. Moreover, by interpreting λ as a soft trust coefficient, we can view the update as a controllable balance between prompt semantics and visual fidelity.

Propagation of Residual Noise Error. Let $\delta_t := \hat{\epsilon} - \epsilon_t^*$ denote the residual noise error at timestep t with respect to the ideal generative noise ϵ_t^*. The DDIM update propagates this noise error linearly into the latent space:

$$z_t = z_t^* + C_{t,2} \cdot \delta_t, \tag{21}$$

where z_t^* denotes the latent corresponding to ideal inversion. Hence, even if $\mathcal{L}_{\text{ref}}$ is not minimized to zero, the resulting latent perturbation is bounded and scales linearly with $\|\delta_t\|$, which DCI attempts to iteratively reduce via fixed-point refinement.

Editability Preservation. Unlike optimization methods that overly constrain z_t toward reconstruction, DCI balances reconstruction and generative semantics. The fixed-point loss $\mathcal{L}_{\text{fix}}$ enforces consistency with the model's forward trajectory rather than a hard projection to a reconstruction target, which helps preserve the generative flexibility required for downstream editing. Formally, if $f_\theta(z)$ approximates the forward generative trajectory, minimizing $\|f_\theta(z_t) - z_t\|$ ensures that z_t lies on a semantically meaningful denoising path, rather than collapsing to a static reconstruction point.

Numerical Robustness. Empirically, the fixed-point refinement converges within 3–10 iterations under a moderate learning rate $\eta \in [10^{-4}, 10^{-2}]$. As $\nabla\mathcal{L}_{\text{fix}}$ involves only first-order derivatives of the denoiser ϵ_θ, the update is numerically stable under automatic differentiation and does not amplify high-frequency errors.

Impact of λ and η on Optimization Dynamics. DCI offers explicit knobs to trade off visual grounding (λ) and convergence aggressiveness (η). Large λ may overfit to the reference signal and degrade semantic consistency; large η may induce oscillation or overshoot in latent updates. As shown in the ablation (Table 2), default values $\lambda = 2$ and $\eta = 0.001$ yield a stable equilibrium across editing tasks.

B Experimental Results on Different Datasets and Editing Engines

We first present additional results of our method on the PIE-Bench benchmark. As shown in Figure 6, our approach clearly outperforms existing baselines. The red circles highlight undesirable artifacts and imprecise edits introduced by other methods, while our method achieves target edits with high

fidelity. These results demonstrate that our approach excels in terms of editing precision, artifact suppression, and background consistency.

We then evaluate our method on a broader dataset, specifically in *COCO2017* [28], to assess generalization to open-world scenarios. We employ our Dual-Conditional Inversion (DCI) for the inversion stage and adopt P2P [15] as the editing engine. The text prompts are generated by a large language model, with only the desired editing attribute manually modified. As shown in Figure 7, our method maintains strong editing performance even in diverse and unconstrained real-world contexts.

Finally, we examine the compatibility of our inversion method with alternative downstream editing engines. We use the popular Masactrl [4] framework as a representative case. As shown in Figure 8, the results demonstrate that our method performs robustly across different editing pipelines, highlighting its generalizability and adaptability.

C Inversion Time Comparison

Despite involving an iterative optimization procedure, our DCI method maintains competitive runtime performance. On a single NVIDIA RTX 4090 GPU, a full DCI inversion-editing cycle takes only average 12.1 seconds per image, which is comparable to the baseline DDIM inversion method [45]. This efficiency is largely due to the lightweight design of our fixed-point refinement and the use of one-step noise correction per iteration. Compared to other methods that involve intensive text

Table 6: Comparison of inversion times (in seconds) across different methods.

Inversion Method	Inversion Time (s)
DDIM	11.55
NTI	137.54
NPI	11.75
AIDI	87.21
NMG	16.71
DirectINV	19.94
ProxEdit	11.75
SPDInv	27.04
DCI(ours)	12.13

embedding optimization or complex auxiliary modules, such as Null-text inversion (NTI) [33], Negative prompt inversion (NPI) [32], AIDI [37], Noise Map Guidance (NMG) [6], Direct Inversion (DirectINV) [20], ProxEdit [14], and SPDInv [25]—our method achieves a favorable balance between quality and speed. Some of these baselines require additional optimization rounds or rely on extra network branches. DCI requires only a small number of optimization steps and converges quickly, while still avoiding heavy architectural changes, making it more practical for real-world deployment.

D Failure Cases

While DCI significantly improves inversion quality and editing controllability, it still exhibits limitations in certain scenarios. Since our method is designed for an independent optimization method of inversion process without being aligned with downstream editing objectives. Its performance can be adversely affected by the characteristics and limitations of the editing engine itself. In particular, failure cases may arise when the editing model lacks sufficient semantic alignment or spatial precision, resulting in incomplete edits or distorted outputs.

As shown in Figure 5, one failure example occurs when the target edit conflicts with the original content. In this case, the edited image either fails to reflect the desired changes or introduces unwanted artifacts. Such outcomes highlight the dependency of DCI on the quality and specificity of downstream editing models.

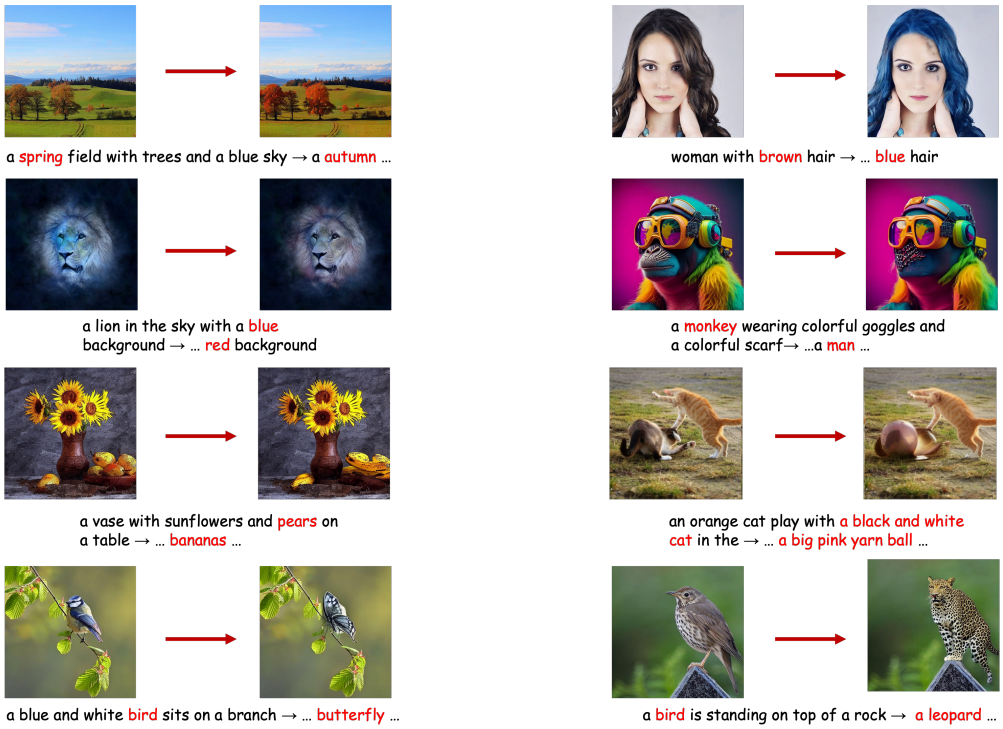

Figure 5: Failure cases

E Limitations and Future Work

Limitations. While our proposed Dual-Conditional Inversion (DCI) framework demonstrates superior performance in diffusion-based image editing, several limitations remain.

First, DCI introduces additional computational overhead due to its dual-stage optimization process, including reference-guided noise correction and fixed-point refinement. Although this leads to increased inference time compared to purely feed-forward inversion methods, we find the overhead to be acceptable in most practical applications, especially those prioritizing editing fidelity over real-time speed. Second, the effectiveness of DCI depends on the quality and semantic alignment of both the source prompt and the reference image. In cases where the prompt is overly ambiguous or poorly aligned with the image content, the dual-conditioning mechanism may lead to suboptimal inversion performance or conflicting guidance signals. Lastly, the extension of DCI to other modalities such as video, 3D scenes, or multi-view data remains unexplored.

Applicability Conditions

- **Editing with ambiguous or weak prompts.** The incorporation of reference images enables stable inversion when text prompts alone are insufficient for guiding precise edits.
- **Applications requiring high reconstruction fidelity.** Tasks such as identity preservation, localized image edits, or photo-realistic retouching benefit from the semantic and structural anchoring provided by DCI.
- **DDIM-based architectures.** DCI is currently implemented and evaluated with DDIM. Compatibility with other samplers (e.g., DPM-Solver) may require re-derivation or empirical verification.

Future Work. We plan to explore several promising directions: (1) improving computational efficiency through adaptive early stopping or learned refinement modules; (2) extending DCI to higher-resolution pipelines and multi-modal inputs such as text-image-mask triplets; and (3) evaluating the applicability of DCI in diverse generative backbones, including DiT and other transformer-based diffusion models. We also aim to conduct user studies in practical editing tools to assess robustness, usability, and real-world performance.

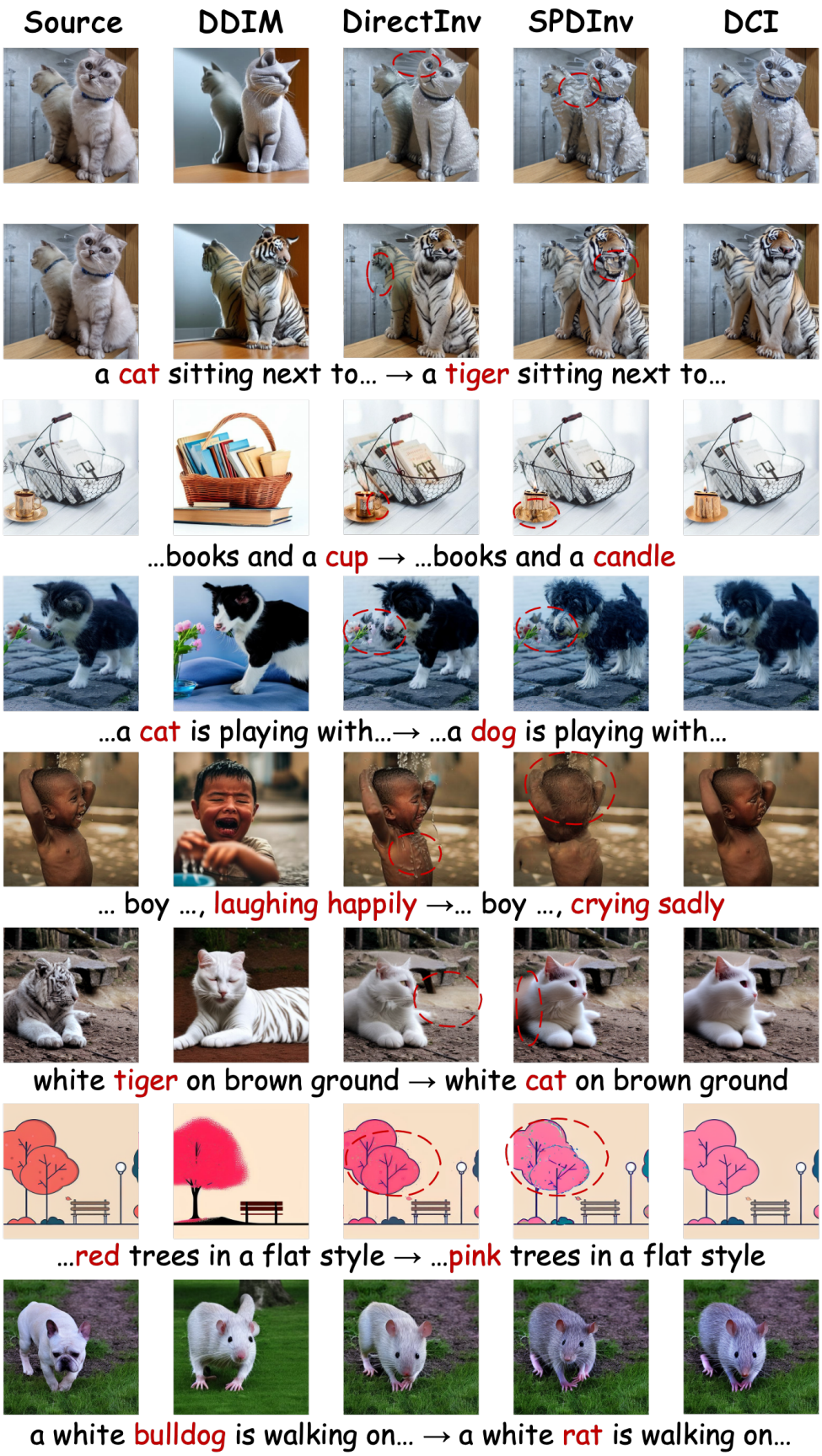

Figure 6: More visual results in PIE-Bench with P2P

Figure 7: Visual results on *COCO2017* dataset

Figure 8: Visual results of DCI with Masactrl