

Multi-History-Step SDE Inversion for Image Editing with Superior Regional Awareness

Haiyan Wei^{1,2}, Yunlong Wang¹, Huaibo Huang¹, Zhenan Sun¹, and Kunbo Zhang^{1*}

¹ New Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences

² University of Chinese Academy of Sciences
{weihaiyan2026,yunlong.wang,kunbo.zhang}@ia.ac.cn,
huaibo.huang@cripac.ia.ac.cn, znsun@nlpr.ia.ac.cn

Abstract. In recent years, diffusion stochastic differential equation (SDE) inversion and inversion-free methods have become prevalent for training-free image editing, as they can achieve faithful reconstruction without tuning. However, existing approaches remain inefficient, exhibit limited plasticity, and struggle to accurately preserve unedited regions. To address these issues, we propose MIEdit, a training-free editing framework based on SDE inversion. MIEdit introduces a predictor-corrector multi-history-step scheme to achieve superior editing quality with fewer steps. We further mitigate heterogeneity and conflict between the multi-conditioned noise residuals and gradient terms during sampling, improving stability and editing plasticity under large edits. MIEdit also includes Inversion-Time Automatic Semantic Angle Masking (IASM); it leverages classifier-free guidance to automatically generate semantic angle masks during inversion and applies them throughout the sampling process for regional constraints, without extra user inputs. We additionally construct EditEval++ (30 fine-grained tasks, 1,000+ image-text-mask triplets) for comprehensive evaluation; experiments show that MIEdit outperforms state-of-the-art techniques. Project page: <https://whywwwzzzg.github.io/MIEdit/>.

Keywords: image editing · image generation · diffusion models

1 Introduction

Recently, pretrained text-to-image diffusion models have driven numerous training-free image editing methods [10, 12, 24, 26, 27, 42, 54]. Many image editing methods rely on deterministic ordinary differential equation (ODE) process inversion, such as Denoising Diffusion Implicit Models (DDIM) inversion and flow inversion [10, 27, 37, 39]. These approaches map an input image back to latent variables or a noise trajectory of the diffusion process, enabling the synthesis of an edited image under a new prompt [9, 25]. However, ODE inversion typically involves

* Corresponding author.

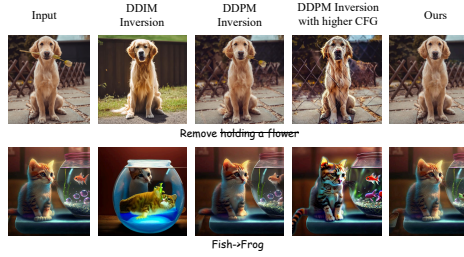


Fig. 1: Editing results under different methods and settings. DDIM inversion (ODE inversion) often suffers from poor consistency, while DDPM inversion (SDE inversion) may have limited plasticity: increasing CFG can degrade image quality before improving prompt adherence.

implicit equations over its own variables, and explicit approximations may introduce severe reconstruction errors, particularly when conditioning is incorporated [2, 21, 23]; consequently, many methods require tuning or multiple network evaluations within a single inversion step to enhance reconstruction consistency, which is highly time-consuming [11, 16, 33, 38]. Recently, approaches based on model prediction differences, including inversion-free editing and denoising diffusion probabilistic model (DDPM) inversion, have emerged, theoretically enabling error-free reconstruction, avoiding optimization and often achieving better background preservation [4, 51].

We generalize DDPM inversion as a form of stochastic differential equation (SDE) inversion. It leverages latents precomputed from the forward process to compute the noise terms required for reconstruction (sampling), thereby correcting the generation process [4, 18, 49]. Nevertheless, existing SDE inversion methods mainly focus on low-order SDE samplers, leaving room for improvement in both convergence speed and generation quality [18]. In particular, the generation quality (i.e., editing plasticity) of SDE-inversion-based editing sometimes suffers from insufficient stability and may even exhibit artifacts, as shown in Fig. 1. Moreover, both ODE- and SDE-inversion-based editing methods remain weak in preserving non-edited regions, making it difficult to precisely handle editing tasks with different modification extents [5, 13, 28, 51]. Some mask-control strategies attempt to alleviate this issue, yet they often require additional inputs or an independent mask branch [8, 56].

To address these issues, we propose **MIEdit**, an image editing method based on arbitrary multi-history-step (at least two history terms per step) diffusion SDE inversion that integrates a predictor-corrector mechanism for higher-quality image generation. MIEdit can reconstruct the input exactly (up to numerical precision) and requires no tuning. Compared to existing SDE inversion methods, our approach generates diverse samples of higher quality with fewer timesteps. Another key improvement is that we carefully modify the inference-time noise schedule, which substantially improves the generation quality of naive multi-history-step inversion in large-edit scenarios and effectively mitigates the artifact issue of SDE inversion.

To improve non-edited region control, we further propose Inversion-Time Automatic Semantic Angle Masking (IASM) without any external conditions. Specifically, we move mask generation from a byproduct of sampling to the early phase of inversion, and introduce a direction-based metric to obtain fine-grained mask control that is available throughout all generation steps. Importantly, the proposed mask does not rely on image quality during generation, nor on the gradual structural alignment between the inversion and generation stages. Our contributions are summarized as follows:

- We reformulate DDPM inversion under a unified SDE inversion framework, and propose a predictor–corrector two-stage, multi-history-step SDE-based inversion-and-generation image editing method, improving both the efficiency and generation quality of current training-free image editing.
- We address condition-induced mismatch and interference between the noise part and gradient terms, significantly improving the plasticity of multi-history-step SDE inversion.
- We propose an inversion-time automatic semantic angle masking that leverages the classifier-free guidance (CFG) process within inversion itself to produce accurate semantic masks for guiding the entire generation process. It requires no extra inputs or inference branches.
- We introduce a new benchmark, EditEval++, which expands EditEval’s original 7 tasks [17] to 30 fine-grained editing tasks and contains 1,000+ image–text–mask triplets, featuring more diverse scenes and descriptions. Experiments on PIE-Bench and EditEval++ demonstrate the advantages of our method.

2 Related Work

2.1 Diffusion Inversion for Image Editing

Many training-free methods enable controllable image editing via diffusion inversion [6, 10, 24, 34, 35]. The key idea is to map an image back to initial noise or intermediate states at selected timesteps so that sampling from the same latent reconstructs the input, dating back to DDIM [39]. Methods like RF-Inv further adopt a flow-based ODE inversion strategy [37, 50]. Faithful reconstruction, however, often requires extra optimization or multiple function evaluations per step for error correction [2, 32, 33, 44, 45]. Methods like CycleDiffusion instead adopt DDPM inversion by recovering a sequence of noise vectors such that the image can be exactly reconstructed by the DDPM sampling process [18, 49]. These SDE methods are closely related to recent inversion-free approaches [21, 23, 51], where network evaluations apply differential corrections to adjust the original trajectory or image [18], mitigating error accumulation in iterative ODE inversion and improving reconstruction fidelity. Our method, inspired by the high-order SA-Solver [52], achieves higher efficiency and quality with exact reconstruction. SA-Solver admits a closed-form arbitrary-order SDE formulation, with DDPM and SDE-DPM-Solver++ as low-order cases [4, 29, 52].

2.2 Precisely Localized Image Editing

Image editing typically requires localized changes while keeping the rest of the image intact. Prior work often reuses inversion/reconstruction cues during generation (e.g., attention-feature injection [56] and blended diffusion [1]), which often requires specifying regions using masks. Some studies rely on user-provided external masks, which incur substantial annotation overhead [1, 30, 56]. Other methods infer masks from attention maps during diffusion [5, 13, 43, 48, 51]. Nevertheless, this strategy typically requires an external “blended-words” prompt to select which token(s)’ attention to use, and it often identifies the editing subject rather than the precise editing region. Moreover, existing automated mask generation methods either cannot reliably provide high-quality control throughout the full sampling trajectory, or incur additional overhead when implemented as an extra inference branch [8, 28, 41].

3 Methods

3.1 Diffusion SDE and Its Inversion

In the DDPM framework [14], we are given a noise schedule $\{(\alpha_t, \sigma_t)\}_{t=0}^T$ with $\alpha_t > 0$ and $\sigma_t > 0$. The forward diffusion perturbs a clean sample x_0 into noise; its marginal form is:

$$x_t = \alpha_t x_0 + \sigma_t \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \mathbf{I}). \quad (1)$$

The corresponding reverse sampling (generation) process starts from $x_T \sim \mathcal{N}(0, I)$ and iterates $t = T, \dots, 1$:

$$x_{t-1} = \mu_t(x_t, c) + \tilde{\sigma}_t z_t, \quad z_t \sim \mathcal{N}(0, I). \quad (2)$$

which yields x_0 at the end. Here $\mu_t(\cdot, c)$ is computed from a denoising network, and $\tilde{\sigma}_t$ is determined by the sampler and schedule.

In the continuous limit, the diffusion can be written as a linear SDE [40]:

$$dx_t = f(t) x_t dt + g(t) dw_t, \quad x_t | x_0 \sim \mathcal{N}(\alpha_t x_0, \sigma_t^2 \mathbf{I}). \quad (3)$$

The equivalent reverse-time generative process is:

$$dx_t = (f(t)x_t - g(t)^2 \nabla_x \log p_t(x_t)) dt + g(t) d\bar{w}_t. \quad (4)$$

We approximate the score by $s_\theta(x_t, t)$. A common parameterization is noise prediction $\epsilon_\theta(x_t, t) = -\sigma_t s_\theta(x_t, t)$, with the corresponding data prediction:

$$x_\theta(x_t, t) = \frac{x_t - \sigma_t \epsilon_\theta(x_t, t)}{\alpha_t}. \quad (5)$$

Based on the above, inversion aims to construct latent codes for a given image x_0 such that the reverse trajectory reconstructs x_0 . For a particular SDE generation, under a given noise level and condition c , the initial random state and the random drives injected throughout the process (in the DDPM case, the per-step noises $\{z_t\}_{t=1}^T$ in Eq. (2)) uniquely determine the final generated sample x_0 . Therefore, they can be viewed as latent codes. Correspondingly, DDPM inversion aims to, given a sample x_0 , construct such latent variables.

3.2 Naive SA-Inversion

We further improve the efficiency of DDPM inversion [14, 18] by optimizing the sampling strategy. Specifically, we rephrase DDPM inversion as SDE inversion and derive an efficient inversion scheme using a refined multi-history-step stochastic diffusion sampler together with its sampling recursion [52].

Under the SDE formulation, the latent code to be inverted corresponds to a Wiener process path $w_{[0,T]}$. In practical numerical implementation, it can be equivalently represented as the Brownian increment sequence $\{\Delta w_i\}$ on a discrete time grid. This sequence can deterministically reconstruct the input image from the noise endpoint. If we further replace the conditioning text, we obtain the corresponding image editing result.

For our method, given an input image x_0 , we choose timesteps $\{t_i\}_{i=0}^M$ and construct an auxiliary noisy reconstruction sequence:

$$x_{t_i}^{\text{inv}} = \alpha_{t_i} x_0 + \sigma_{t_i} \tilde{\epsilon}, \quad \tilde{\epsilon} \sim \mathcal{N}(0, I). \quad (6)$$

where α_{t_i} and σ_{t_i} are respectively the signal coefficient and the noise standard deviation at time t_i . We define $\lambda_t = \log(\alpha_t/\sigma_t)$ and $\lambda_i \triangleq \lambda(t_i)$.

Taking the predictor in SA-Solver [52] as an example (the corrector is analogous), we compute and store the per-step noise term η_i^P :

$$\eta_i^P = x_{t_{i+1}}^{\text{inv}} - \frac{\sigma_{t_{i+1}}}{\sigma_{t_i}} \exp\left(-\int_{\lambda_i}^{\lambda_{i+1}} \tau^2(\bar{\lambda}) d\bar{\lambda}\right) x_{t_i}^{\text{inv}} - \sum_{j=0}^{s-1} b_{i-j} x_{\theta}\left(x_{t_{i-j}}^{\text{inv}}, t_{i-j}\right). \quad (7)$$

Here $x_{\theta}(\cdot, \cdot)$ is the data-prediction model; s is the number of model evaluations used by the multi-history-step method (i.e., its order). In this setting, “multi-history-step” indicates that the current sampling step uses not only the model output from the current latent variable, but also the outputs from the preceding $s-1$ time steps for interpolation. $\tau(\cdot)$ is a variance control function; and the coefficients b_{i-j} are integral weights derived from the Lagrange interpolation basis functions [52]. During generation, we use the stored per-step noise term η_i^P from the inversion process to replace the random noise component $\tilde{\sigma}_i \xi$:

$$x_{t_{i+1}} = \frac{\sigma_{t_{i+1}}}{\sigma_{t_i}} \exp\left(-\int_{\lambda_i}^{\lambda_{i+1}} \tau^2(\bar{\lambda}) d\bar{\lambda}\right) x_{t_i} + \sum_{j=0}^{s-1} b_{i-j} x_{\theta}(x_{t_{i-j}}, t_{i-j}) + \eta_i^P. \quad (8)$$

The original noise scale is

$$\tilde{\sigma}_i = \sigma_{t_{i+1}} \sqrt{1 - \exp\left(-2 \int_{\lambda_i}^{\lambda_{i+1}} \tau^2(\bar{\lambda}) d\bar{\lambda}\right)}. \quad (9)$$

Overall pseudocode for the process is provided in the supplementary material.

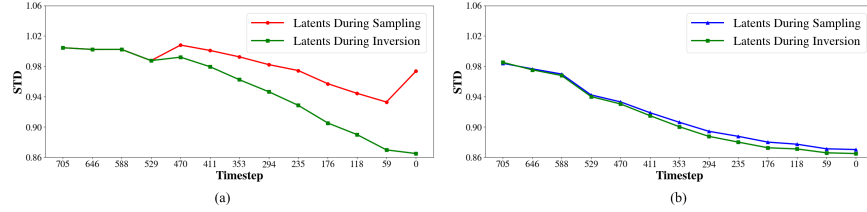


Fig. 2: (a) Latent standard deviation at each timestep during inversion and sampling with Naive SA-Inversion. (b) Latent standard deviation at each timestep with our method, which prevents latent-scale explosion and keeps the trajectory within a reasonable range by closely following inversion statistics.

3.3 Granting Higher Plasticity to Multi-History-Step SDE Inversion

We observe that although the Naive SA-Inversion method achieves nearly perfect reconstruction and requires only a few iterations, its editing plasticity—*i.e.*, the ability to alter the output as instructed—is insufficient. Specifically, when the desired edit deviates significantly from the source image, the generation result often responds weakly to the condition, and may even produce artifacts. We examine the variance of the latent trajectory $\{x_{t_i}\}_{i=0}^M$ and find that it diverges from the forward noising distribution, especially at middle-to-late timesteps: the latent scale becomes biased, and the latent standard deviation is close to 1, as shown in Fig. 2. The more historical steps used in SDE inversion, the more pronounced this deviation tends to be.

The key issue stems from the “stored-and-reused” η_i^P . In standard generation, the noise term follows a zero-mean Gaussian distribution with standard deviation $\tilde{\sigma}_i$ given in (9). In contrast, during image editing, the noise term η_i^P exhibits a higher variance in practice due to encompassing conditional information correlated with the source image. While multi-step history terms improve fitting accuracy, they also introduce heterogeneity in the noise term across different latent and timestep conditions. On the other hand, when the target image differs significantly from the source image, the correlation between the noise term and the gradient term becomes weaker. These two aspects lead to superposition and conflict between the multi-step noise term and the multi-step gradient term, causing $x_{t_{i+1}}$ computed by Eq. (8) to have abnormally large variance. This inflated variance then propagates to later steps and can ultimately cause the final sample to collapse. With the predictor step, we can rewrite:

$$\eta_i^P = x_{t_{i+1}}^{\text{inv}} - a_i x_{t_i}^{\text{inv}} - g_i, \quad (10)$$

where

$$a_i = \frac{\sigma_{t_{i+1}}}{\sigma_{t_i}} \exp\left(-\int_{\lambda_i}^{\lambda_{i+1}} \tau^2(\bar{\lambda}) d\bar{\lambda}\right), \quad g_i = \sum_{j=0}^{s-1} b_{i-j} x_{\theta}\left(x_{t_{i-j}}^{\text{inv}}, t_{i-j}\right). \quad (11)$$

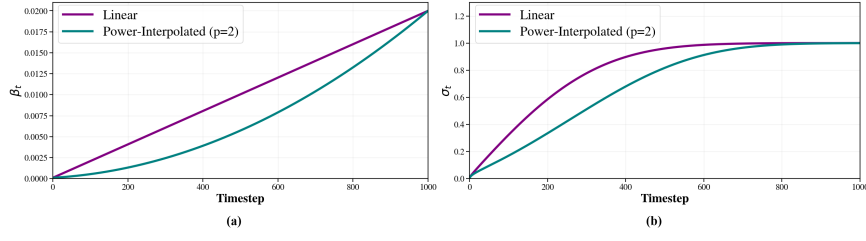


Fig. 3: (a) The linear β schedule used in other inversion methods, and our power-interpolated β schedule. (b) The corresponding σ schedules for the two β schedules (Eq. (14)), indicating the noise level at each timestep during inversion and sampling. Our power-interpolated β schedule yields a lower noise level at the same timestep.

Then

$$\begin{aligned} \text{Var}(\eta_i^P) = & \text{Var}(x_{t_{i+1}}^{\text{inv}}) + a_i^2 \text{Var}(x_{t_i}^{\text{inv}}) + \text{Var}(g_i) - 2a_i \text{Cov}(x_{t_{i+1}}^{\text{inv}}, x_{t_i}^{\text{inv}}) \\ & - 2 \text{Cov}(x_{t_{i+1}}^{\text{inv}}, g_i) + 2a_i \text{Cov}(x_{t_i}^{\text{inv}}, g_i). \end{aligned} \quad (12)$$

It can be seen that $\text{Var}(\eta_i^P)$ is mainly governed by: (1) the endpoint variances, (2) $\text{Var}(g_i)$ (whose scale is strongly tied to $\sigma_{t_{i+1}}$), and (3) the covariance terms, among which $-2a_i \text{Cov}(x_{t_{i+1}}^{\text{inv}}, x_{t_i}^{\text{inv}})$ is often dominant.

Therefore, to reduce the overall variance and stabilize η_i^P , we lower the noise level at each timestep: this simultaneously decreases $\text{Var}(x_{t_{i+1}}^{\text{inv}})$ and $\text{Var}(x_{t_i}^{\text{inv}})$; moreover, a lower noise level also reduces $\text{Var}(g_i)$ through its strong dependence on $\sigma_{t_{i+1}}$; and it increases the correlation between adjacent latents, thereby decreasing the dominant covariance contribution $-2a_i \text{Cov}(x_{t_{i+1}}^{\text{inv}}, x_{t_i}^{\text{inv}})$.

Thus, we adopt a simple yet effective approach: we replace the standard linear β schedule (used to define the forward process) with a power-interpolated schedule at inference time:

$$\beta_t = \left((1-u) \beta_{\text{start}}^{1/p} + u \beta_{\text{end}}^{1/p} \right)^p, \quad p \geq 1, \quad (13)$$

where p is a shape-controlling parameter and $u \in [0, 1]$ is the normalized time. This schedule injects noise more smoothly at early timesteps and yields a smaller β_{t_i} at the same timestep, as illustrated in Fig. 3. We have:

$$\sigma_{t_i} = \sqrt{1 - \prod_{j=1}^{t_i} (1 - \beta_j)}. \quad (14)$$

This results in a smaller σ_{t_i} at the same timestep compared to a linear schedule. It makes the sampling trajectory better match the training-time noise statistics. Ultimately, it retains perfect reconstruction (because the noise schedule used for inversion is still the same as the one used for sampling) while improving the editing plasticity of multi-history-step SDE inversion under large edits and reducing artifacts in the generated images.

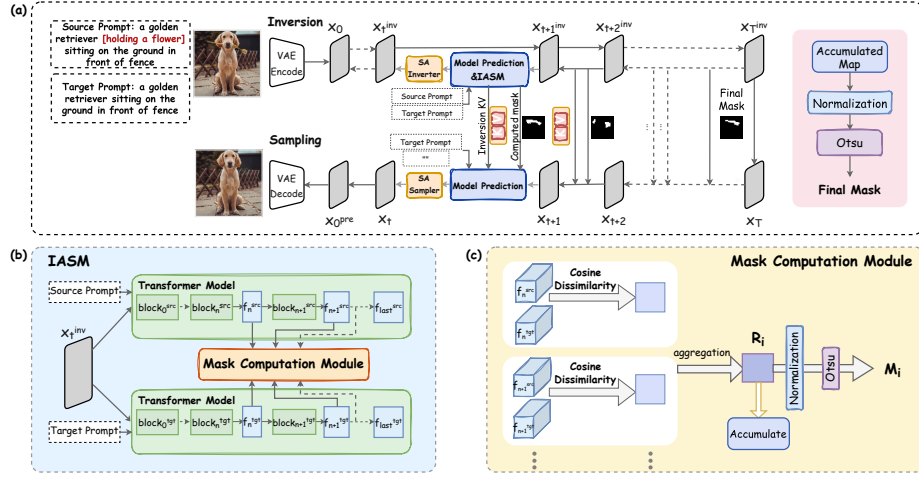


Fig. 4: (a) Pipeline overview. Given a source image and prompt, we invert to obtain step-wise noise terms, masks, and key-value (KV) features, which guide generation. (b) IASM Module. Using CFG during inversion, we set the inversion negative prompt to the target prompt for mask computation. (c) Mask Computation Module. We compute a semantic angle mask via cosine dissimilarity between corresponding feature maps.

3.4 Inversion-Time Automatic Semantic Masking

Most image-editing pipelines derive masks from external cues, such as user-drawn regions or blended-word tokens for attention masking [13, 51]. In addition, attention masks can be noisy, and edited regions may not align with explicit prompt tokens, making blended-word specification infeasible. Recent methods attempt to auto-generate masks without manual annotation, yet limitations persist. A common approach derives a mask from the prediction discrepancy between inversion and generation at matched timesteps [28]. This method produces reliable masks only at a few mid-stage steps (early steps are too noisy), and fails to constrain early/mid generation, so small errors accumulate. Moreover, masks generated in the early and mid stages tend to be of low quality because the latent variables still contain substantial noise at these steps.

To address the need for adaptive region control, we propose *Inversion-Time Automatic Semantic Masking (IASM)*. We generate masks during inversion, particularly at early and mid-stage inversion timesteps, to guide generation using inversion-derived information. Such masks originate from lower-noise latents, providing higher-confidence indications of the editing regions, and can precisely control the entire generation process, including the early generation stages. Our method requires no external conditions and remains effective even under methods with weak inversion-generation consistency; an overview is shown in Fig. 4.

CFG for Masking For SDE inversion and inversion-free methods, classifier-free guidance (CFG) [15] is widely used to strengthen text conditioning by combining

unconditional and conditional predictions:

$$\hat{x}_\theta(x_{t_i}; C, C_{\text{neg}}) = x_\theta(x_{t_i}, C_{\text{neg}}) + w_g (x_\theta(x_{t_i}, C) - x_\theta(x_{t_i}, C_{\text{neg}})), \quad (15)$$

where w_g denotes the guidance scale; larger values strengthen conditional consistency. Here x_θ denotes the model predictor parameterized by θ . In image editing, let C_{src} and C_{tgt} be the source and target prompts. Inversion and generation use guidance prompts C_{neg}^i and C_{neg}^g , with strengths w_i and w_g . We take a first-order SDE as an example (omitting time-dependent parameters and constants), applying the inverted latent from inversion to generation yields:

$$\begin{aligned} x_{t_{i+1}} = & x_{t_{i+1}}^{\text{inv}} + a_i(x_{t_i} - x_{t_i}^{\text{inv}}) + b_i \left[w_g x_\theta(x_{t_i}, C_{\text{tgt}}) + (w_i - 1) x_\theta(x_{t_i}^{\text{inv}}, C_{\text{neg}}^i) \right. \\ & \left. - w_i x_\theta(x_{t_i}^{\text{inv}}, C_{\text{src}}) - (w_g - 1) x_\theta(x_{t_i}, C_{\text{neg}}^g) \right]. \end{aligned} \quad (16)$$

Intuitively, CFG drives the next latent toward C_{tgt} and C_{neg}^i , and away from C_{src} and C_{neg}^g . However, most existing methods set C_{neg}^i to empty text. This can introduce ambiguity because the unconditional (empty-text) branch is implicitly used as a positive reference during inversion, which is misaligned with the editing objective. Therefore, we set the inversion-stage guidance prompt as $C_{\text{neg}}^i = C_{\text{tgt}}$, which improves semantic consistency while maintaining background preservation; more importantly, it enables mask acquisition through CFG computation within the IASM framework. Concretely, at inversion state $x_{t_i}^{\text{inv}}$ we compute conditional predictions for both branches, $x_\theta(x_{t_i}^{\text{inv}}, C_{\text{src}})$ and $x_\theta(x_{t_i}^{\text{inv}}, C_{\text{tgt}})$. Small differences indicate strong source–target agreement (keep unchanged as mask background), while large differences indicate semantic conflict—regions supported by the source but not by the target—and should be edited. Since this relies on the semantic gap rather than fully specified prompts, IASM remains robust to partially descriptive prompts as long as the intended edit is expressed.

Temporal Scheduling of Masks. We compute masks mainly on lower-noise latents in early inversion, where structures and details are clearer. Furthermore, latents at different noise intensities exhibit distinct generation focuses, and network prediction differences at corresponding noise levels indicate editing priorities for those noise scales.

We adopt temporally aligned mask scheduling in IASM. We compute single-step masks only at a few high-confidence timesteps within $[d_s, d_e)$ and apply them to the corresponding sampling steps. For the remaining (higher-noise) steps, we use an averaged mask M_{avg} obtained by averaging dissimilarity maps over $[d_s, d_e)$, which reduces variance and improves stability. Formally, given single-step masks obtained at timesteps d_s through $d_e - 1$, the mask used at generation step t is:

$$M_t^{\text{gen}} = \begin{cases} M_t, & d_s \leq t < d_e, \\ M_{\text{avg}}, & \text{otherwise,} \end{cases} \quad (17)$$

This lets masks from clearer intermediate representations be used where most beneficial and stably guide early generation.

Mask Computation Method. Pixel- or latent-space prediction changes mostly capture appearance differences and may miss cases where pixels look similar but semantics differ. Attention maps are more semantic but are often noisy and frequently depend on specific network designs. IASM combines the strengths by comparing internal multi-layer feature-map differences between the two network branches. On the other hand, since L2/absolute differences mainly reflect magnitude changes while semantic gaps often appear as directional changes, we use **cosine dissimilarity** to measure semantic direction consistency. At inversion state $x_{t_i}^{\text{inv}}$, we extract feature maps under source/target conditions:

$$f_k^{\text{src}} = F_k(x_{t_i}^{\text{inv}}, C_{\text{src}}), \quad f_k^{\text{tgt}} = F_k(x_{t_i}^{\text{inv}}, C_{\text{tgt}}). \quad (18)$$

where $F_k(\cdot)$ denotes the feature extractor for layer k , and f_k represents the spatial feature map. To measure semantic conflict, we compute the cosine similarity and define the angle-response map:

$$S_{\cos}^{(k)}(i, j) = \frac{f_k^{\text{src}}(i, j) \cdot f_k^{\text{tgt}}(i, j)}{\|f_k^{\text{src}}(i, j)\|_2 \|f_k^{\text{tgt}}(i, j)\|_2}, \quad M_{\text{angle}}^{(k)}(i, j) = 1 - S_{\cos}^{(k)}(i, j). \quad (19)$$

Larger values indicate greater directional inconsistency and thus a higher likelihood that the position belongs to the edit region. Averaging over layers gives the dissimilarity map R_i for step i , and applying post-processing to R_i yields the single-step mask M_i [22].

4 Experiments

4.1 Setup

Baselines and Implementation. We compare our method with representative training-free image editing approaches: (i) feature-control methods (P2P [13], PnP [43], MasaCtrl [5]); (ii) SDE-inversion and inversion-free methods (CycleDiffusion [49], Edit-Friendly DDPM Inversion (EF) [18], LEDITS++ [4], InEdit [51], FlowEdit [23], FlowAlign [21], FlowCycle [46]); and (iii) ODE-inversion methods (DDIM Inversion [39], Direct Inversion(DI) [19], FTEdit [50], iRFDS [53], FreeInv [2]). For fairness, we follow the official evaluation protocols of all baselines; for FlowEdit, we set CFG to 7.0 during editing to improve background consistency, while keeping other settings identical to its official implementation. Our method is implemented on PixArt- Σ [6], a DiT-based model [34] that substantially improves over PixArt- α [7]. We set CFG to 3.1/9.0 for inversion/editing, and use a shape-control parameter $p = 1.9$. To evaluate stronger backbones and ensure fair comparison with recent methods, we also implement our approach on Stable Diffusion 3 Medium and Stable Diffusion 3.5 Medium [10], using CFG 1.6/3.6 for inversion/editing. All experiments are conducted in PyTorch.

Evaluation Datasets and Metrics. We evaluate general editing performance on PIE-Bench [19] and EditEval [17]. PIE-Bench contains 700 natural and artificial images covering 9 dimensions; each sample provides source/target

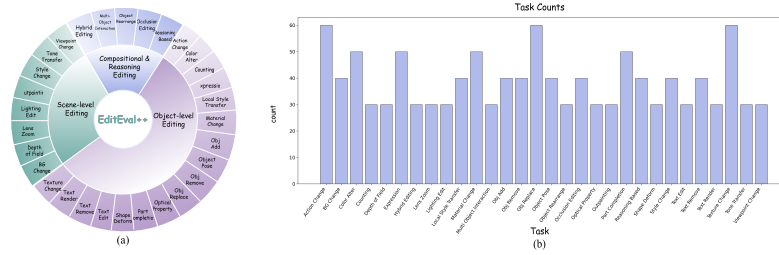


Fig. 5: (a) We propose EditEval++, which covers 30 tasks across three major categories: Object-level Editing, Scene-level Editing, and Compositional & Reasoning Editing. (b) The number of (image, prompt, mask) triplets for each task.

prompts, and region masks to assess background preservation and local edits. EditEval includes 150 high-quality images across 7 task types with text prompts, but its coverage is limited and it provides no region masks for quantitative evaluation; compared to EditEval, PIE-Bench has slightly lower resolution (512×512) and visual quality, and a higher proportion of artistic images.

Therefore, we propose **EditEval++**, a more challenging large-scale image editing benchmark that expands EditEval’s original 7 tasks to 30 fine-grained tasks with varying difficulty levels—positioning it among the most comprehensive description-based image editing benchmarks currently available, as shown in Fig. 5. Other benchmarks remain limited: TedBench [20] covers only 4 tasks with 100 samples, and EditVal [3] spans 12 tasks but offers fewer samples and is instruction-based rather than description-based. EditEval++ includes classic tasks such as shape changes, object additions, and challenging operations that push the limits of existing methods, including lighting edits and depth-of-field modifications. We expand the sample size from EditEval’s 150 to 1160 samples and provide corresponding masks for each sample (for evaluation purposes only). The detailed construction pipeline is provided in the supplementary material.

For metrics, following PIE-Bench, we measure non-edited region preservation using structure distance [19], PSNR, LPIPS [55], MSE, and SSIM [47], and measure edit success using CLIP similarity [36] between the edited region and the target text prompt. Some metrics in the tables are rescaled. For EditEval++, we assess alignment with human preferences via EditScore [31]. We also report the number of function evaluations (NFE) for computational efficiency. For fairness, we define one NFE as one forward pass on a single sample; for example, methods that perform both reconstruction and generation incur two NFEs per step, since the actual computational cost is doubled.

4.2 Comparison with Previous Editing Methods

We qualitatively evaluate our method with baselines on image editing tasks, with results shown in Fig. 6. Our method successfully executes edits according to prompts with high fidelity while better preserving non-target content, validat-

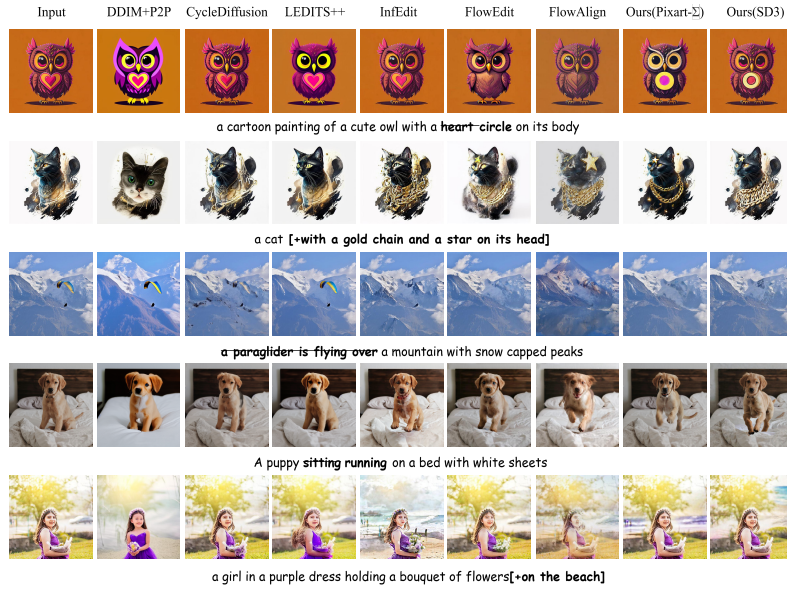


Fig. 6: Qualitative results of different methods.

ing its comprehensive capability for image editing. Quantitative results on PIE-Bench are summarized in Table 1. Our method achieves excellent performance across most metrics and outperforms all baselines in background preservation and instruction fidelity, with instruction fidelity measured by CLIP scores. In particular, our method attains substantially better background preservation than competing approaches. Notably, there exists a fundamental trade-off between preserving source content and achieving strong editing capability. Furthermore, our method demonstrates significantly higher efficiency compared to other methods. We evaluate SDE inversion and inversion-free methods on EditEval++, with results shown in Table 2 and detailed task-by-task performance shown in the supplementary material. Our method maintains superior performance—achieving higher CLIP scores while better preserving background content—further validating its robustness across diverse tasks. As reported in Table 2, our method (SD3) achieves the highest EditScore of 4.30, demonstrating its superior perceptual quality and consistency with human judgment. Moreover, among the methods built on SD3, our approach is the fastest and requires only 1.8 seconds per image, as shown in Table 4.

4.3 Ablation Study and Analysis

We conduct ablation studies on key components of our algorithm on PIE-Bench. First, we study the role of historical terms in the stochastic linear multistep sampling and inversion scheme with integrated predictor-corrector updates. With a fixed number of sampling steps in our PixArt- Σ implementation, Table 3a shows

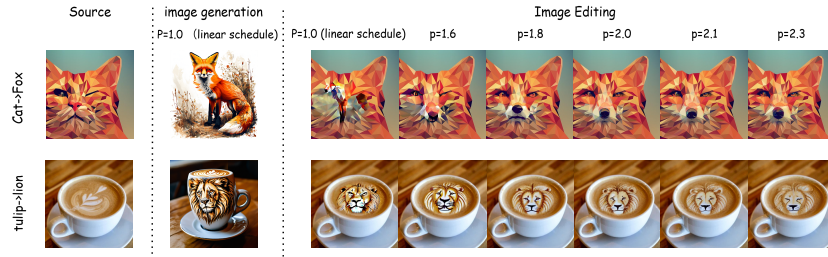
Table 1: Quantitative comparison on PIE-Bench. (best: bold; second: underlined)

Method	Model	Structure	Background Preservation				CLIP	NFE↓
		Distance↓	PSNR↑	LPIPS↓	MSE↓	SSIM↑	Similarity↑	
DDIM+P2P	SD1.4	69.43	17.87	208.80	219.88	71.14	22.44	150
CycleDiffusion	SD1.4	<u>11.15</u>	26.93	62.43	35.41	84.29	21.68	118
DI+MasaCtrl	SD1.4	23.58	22.68	87.41	80.63	81.51	21.41	200
DI+PnP	SD1.5	24.29	22.46	106.06	80.40	79.68	22.62	150
EF+P2P	SD1.4	18.05	24.55	91.88	94.58	81.57	21.03	126
InfEdit	SD1.4	14.22	27.52	47.98	34.17	85.05	22.03	64
LEDITS++	SD1.5	15.76	24.54	71.25	53.05	83.08	20.92	60
iRFDS	SD3	62.72	19.61	186.39	179.76	74.59	21.67	2816
FTEdit	SD3.5	30.12	21.94	118.04	88.09	83.35	<u>22.65</u>	150
FlowEdit	SD3	12.29	26.56	55.30	34.23	89.45	22.50	66
FreeInv+P2P	SD1.5	17.13	26.03	67.90	41.70	83.00	22.33	150
FlowAlign	SD3	28.30	25.50	53.28	43.78	87.92	22.00	66
FlowCycle	SD3	13.30	26.83	63.24	33.56	88.63	22.46	6633
Ours	PixArt- Σ	11.10	28.01	48.13	36.18	87.11	22.26	<u>48</u>
Ours	SD3	12.07	29.44	40.03	23.49	91.17	22.41	38
Ours	SD3.5	13.21	<u>29.35</u>	<u>42.36</u>	<u>25.67</u>	<u>90.86</u>	22.73	38

Table 2: Quantitative comparison on EditEval++ (best in bold).

Method	Structure	PSNR↑	LPIPS↓	MSE↓	SSIM↑	CLIP	EditScore↑
	Distance↓					Similarity↑	
CycleDiffusion	11.15	28.06	67.13	33.24	85.98	22.55	3.66
LEDITS++	16.06	25.41	71.19	50.12	83.96	22.22	3.09
InfEdit (LCM)	33.88	20.48	131.37	243.20	76.13	22.79	4.05
FlowEdit	11.21	27.31	58.40	32.64	89.53	22.62	3.84
FlowAlign	23.50	26.36	71.58	42.53	87.91	22.91	4.02
Ours (PixArt- Σ)	10.75	28.21	52.21	37.51	87.75	22.85	3.98
Ours (SD3)	11.58	29.58	50.42	25.75	90.52	23.30	4.30

that using 2 historical terms for prediction (a third-order predictor) and 3 for correction (a fourth-order corrector) achieves the best editing performance. Next, we analyze our proposed β -schedule design by adjusting the shape-controlling parameter p . linear β schedule can achieve high-quality text-to-image generation; however, for large-scale image editing, it tends to introduce artifacts. In contrast, our power-interpolated β schedule yields stable and high-quality results. Finally, we evaluate IASM for mask localization by comparing the Attention Mask from InfEdit [51] and the Sampling-generated Mask from Follow-Your-Shape [28] within our SD3.5-based SA-Inversion pipeline. Due to the absence of corresponding blended words in many editing prompts and the stronger structural guarantees provided by our algorithm compared to FollowYourShape, the results in Table 3b and Fig. 8 demonstrate our superior capability in localizing editing regions. The last two rows of Table 3b further validate our temporal mask scheduling and the advantage of semantic angle masks over magnitude-based masks.

**Fig. 7:** The impact of the shape-controlling parameter p on the results.**Table 3:** Ablations on history terms and mask design.

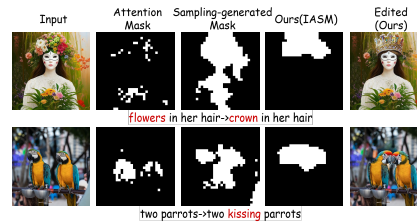
	(a)					(b)		
	$\begin{pmatrix} \text{Predictor} & \text{Corrector} \\ \text{history terms, history terms} \end{pmatrix}$					Method	PSNR $\uparrow$	LPIPS $\downarrow$ CLIP $\uparrow$
Metric	(1, 1)	(2, 2)	(3, 2)	(2, 3)	(3, 3)	Attention Mask (from InfEdit)	28.86	48.06 22.47
PSNR $\uparrow$	27.35	27.80	27.95	28.01	28.08	Sampling-generated Mask (from Follow-Your-Shape)	26.82	60.44 22.39
CLIP $\uparrow$	21.73	22.01	22.06	22.26	22.17	IASM (Ours)	29.35	42.36 22.73
						Full-Step IASM	29.23	43.20 22.68
						Magnitude-Based Mask	28.46	44.63 22.63

5 Conclusion

We present MIEdit, a training-free image editing framework based on high-order SDE inversion. A predictor-corrector multi-history scheme enables exact reconstruction with fewer steps, while an optimized inference noise schedule improves stability and quality for large edits. IASM automatically derives semantic angle masks during inversion from CFG feature discrepancies to constrain regions throughout sampling without extra inputs. Extensive experiments and the new EditEval++ benchmark demonstrate consistent gains, especially on challenging structural edits, and suggest promising extensions to video and 3D generation.

Table 4: Average inference time per image on NVIDIA RTX 5090. Comparison among Stable Diffusion 3 based methods.

Method	FlowEdit	FlowAlign	FlowCycle	Ours
Time (s/img)	2.6	2.4	270.0	1.8

**Fig. 8:** Comparison of three types of localization strategies.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 62476272 and 62576342), and by the Tianjin Key Research and Development Program CAS-Cooperation Project (Grant No. 24YFYSHZ00290).

References

1. Avrahami, O., Fried, O., Lischinski, D.: Blended latent diffusion. *ACM transactions on graphics (TOG)* **42**(4), 1–11 (2023)
2. Bao, Y., Liu, H., Gao, X., Fu, H., Kang, G.: Freeinv: Free lunch for improving ddim inversion. *arXiv preprint arXiv:2503.23035* (2025)
3. Basu, S., Saberi, M., Bhardwaj, S., Chegini, A.M., Massiceti, D., Sanjabi, M., Hu, S.X., Feizi, S.: Editval: Benchmarking diffusion based text-guided image editing methods. *arXiv preprint arXiv:2310.02426* (2023)
4. Brack, M., Friedrich, F., Kornmeier, K., Tsaban, L., Schramowski, P., Kersting, K., Passos, A.: Ledit++: Limitless image editing using text-to-image models. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 8861–8870 (2024)
5. Cao, M., Wang, X., Qi, Z., Shan, Y., Qie, X., Zheng, Y.: Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 22560–22570 (2023)
6. Chen, J., Ge, C., Xie, E., Wu, Y., Yao, L., Ren, X., Wang, Z., Luo, P., Lu, H., Li, Z.: Pixart- σ : Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In: *European Conference on Computer Vision*. pp. 74–91. Springer (2024)
7. Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wu, Y., Wang, Z., Kwok, J., Luo, P., Lu, H., Li, Z.: Pixart- α : Fast training of diffusion transformer for photorealistic text-to-image synthesis (2023)
8. Couairon, G., Verbeek, J., Schwenk, H., Cord, M.: Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427* (2022)
9. Dong, W., Xue, S., Duan, X., Han, S.: Prompt tuning inversion for text-driven image editing using diffusion models. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 7430–7440 (2023)
10. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: *Forty-first international conference on machine learning* (2024)
11. Garibi, D., Patashnik, O., Voynov, A., Averbuch-Elor, H., Cohen-Or, D.: Renoise: Real image inversion through iterative noising. In: *European Conference on Computer Vision*. pp. 395–413. Springer (2024)
12. Ge, S., Huang, C., Ai, Y., Fan, Q., Huang, H., He, R.: Expand and prune: Maximizing trajectory diversity for effective grpo in generative models. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 41913–41922 (2026)
13. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626* (2022)

14. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. *Advances in neural information processing systems* **33**, 6840–6851 (2020)
15. Ho, J., Salimans, T.: Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022)
16. Hong, S., Lee, K., Jeon, S.Y., Bae, H., Chun, S.Y.: On exact inversion of dpm-solvers. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 7069–7078 (2024)
17. Huang, Y., Huang, J., Liu, Y., Yan, M., Lv, J., Liu, J., Xiong, W., Zhang, H., Cao, L., Chen, S.: Diffusion model-based image editing: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025)
18. Huberman-Spiegelglas, I., Kulikov, V., Michaeli, T.: An edit friendly ddpm noise space: Inversion and manipulations. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 12469–12478 (2024)
19. Ju, X., Zeng, A., Bian, Y., Liu, S., Xu, Q.: Direct inversion: Boosting diffusion-based editing with 3 lines of code. *arXiv preprint arXiv:2310.01506* (2023)
20. Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 6007–6017 (2023)
21. Kim, J., Hong, Y., Park, J., Ye, J.C.: Flowalign: Trajectory-regularized, inversion-free flow-based image editing. *arXiv preprint arXiv:2505.23145* (2025)
22. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems* **24** (2011)
23. Kulikov, V., Kleiner, M., Huberman-Spiegelglas, I., Michaeli, T.: Flowedit: Inversion-free text-based editing using pre-trained flow models. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 19721–19730 (2025)
24. Labs, B.F., Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., Kulal, S., Lacey, K., Levi, Y., Li, C., Lorenz, D., Müller, J., Podell, D., Rombach, R., Saini, H., Sauer, A., Smith, L.: Flux.1 kontext: Flow matching for in-context image generation and editing in latent space (2025), <https://arxiv.org/abs/2506.15742>
25. Li, S., Van De Weijer, J., Hu, T., Khan, F.S., Hou, Q., Wang, Y., Yang, J., Cheng, M.M.: Stylediffusion: Prompt-embedding inversion for text-based editing. *arXiv preprint arXiv:2303.15649* (2023)
26. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747* (2022)
27. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003* (2022)
28. Long, Z., Zheng, M., Feng, K., Zhang, X., Liu, H., Yang, H., Zhang, L., Chen, Q., Ma, Y.: Follow-your-shape: Shape-aware image editing via trajectory-guided region control. *arXiv preprint arXiv:2508.08134* (2025)
29. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research* **22**(4), 730–751 (2025)
30. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Re-paint: Inpainting using denoising diffusion probabilistic models. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 11461–11471 (2022)

31. Luo, X., Wang, J., Wu, C., Xiao, S., Jiang, X., Lian, D., Zhang, J., Liu, D., et al.: Editscore: Unlocking online rl for image editing via high-fidelity reward modeling. arXiv preprint arXiv:2509.23909 (2025)
32. Miyake, D., Iohara, A., Saito, Y., Tanaka, T.: Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2063–2072. IEEE (2025)
33. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6038–6047 (2023)
34. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)
35. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952 (2023)
36. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
37. Rout, L., Chen, Y., Ruiz, N., Caramanis, C., Shakkottai, S., Chu, W.S.: Semantic image inversion and editing using rectified stochastic differential equations. arXiv preprint arXiv:2410.10792 (2024)
38. Samuel, D., Meiri, B., Maron, H., Tewel, Y., Darshan, N., Avidan, S., Chechik, G., Ben-Ari, R.: Lightning-fast image inversion and editing for text-to-image diffusion models. arXiv preprint arXiv:2312.12540 (2023)
39. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
40. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
41. Sun, J., Wang, H., Cao, J., Huang, H., He, R.: Marmot: Object-level self-correction via multi-agent reasoning. arXiv preprint arXiv:2504.20054 (2025)
42. Sun, J., Wang, P., Wang, H., Liu, X., Huang, H., He, R.: Towards fine-grained attribution: Instance-aware preference optimization for aligning diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 43155–43164 (2026)
43. Tumanyan, N., Geyer, M., Bagon, S., Dekel, T.: 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. pp. 1921–1930 (2023)
44. Wallace, B., Gokul, A., Naik, N.: Edict: Exact diffusion inversion via coupled transformations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 22532–22541 (2023)
45. Wang, F., Yin, H., Dong, Y.J., Zhu, H., Zhao, H., Qian, H., Li, C., et al.: Belm: Bidirectional explicit linear multi-step sampler for exact inversion in diffusion models. *Advances in Neural Information Processing Systems* **37**, 46118–46159 (2024)
46. Wang, Y., Wang, Z., Chen, L.: Target-aware image editing via cycle-consistent constraints. arXiv preprint arXiv:2510.20212 (2025)
47. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* **13**(4), 600–612 (2004)

48. Wei, H., Xu, H., Zhu, B., Geng, Y., Liu, A., Yin, W., Liu, J.: Patch-wise hypergraph contrastive learning with dual normal distribution weighting for multi-domain stain transfer. In: 2025 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6. IEEE (2025)
49. Wu, C.H., De la Torre, F.: A latent space of stochastic diffusion models for zero-shot image editing and guidance. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7378–7387 (2023)
50. Xu, P., Jiang, B., Hu, X., Luo, D., He, Q., Zhang, J., Wang, C., Wu, Y., Ling, C., Wang, B.: Unveil inversion and invariance in flow transformer for versatile image editing. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 28479–28489 (2025)
51. Xu, S., Huang, Y., Pan, J., Ma, Z., Chai, J.: Inversion-free image editing with natural language. arXiv preprint arXiv:2312.04965 (2023)
52. Xue, S., Yi, M., Luo, W., Zhang, S., Sun, J., Li, Z., Ma, Z.M.: Sa-solver: Stochastic adams solver for fast sampling of diffusion models. *Advances in Neural Information Processing Systems* **36**, 77632–77674 (2023)
53. Yang, X., Chen, C., Yang, X., Liu, F., Lin, G.: Text-to-image rectified flow as plug-and-play priors. arXiv preprint arXiv:2406.03293 (2024)
54. Zhang, K., Sun, M., Sun, J., Zhang, K., Sun, Z., Tan, T.: Open-vocabulary text-driven human image generation. *International Journal of Computer Vision* **132**(10), 4379–4397 (2024)
55. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
56. Zhu, T., Zhang, S., Shao, J., Tang, Y.: Kv-edit: Training-free image editing for precise background preservation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16607–16617 (2025)