InstaEdit: Instant Image Editing via Optimized Noise Prediction

Ning Ma[1] and Yangrui Shao[1]

Dalian University of Technology, Dalian Liaoning 116024, P.R.China
{sherlockma,harrytrolor}@mail.dlut.edu.cn

Abstract. This paper proposes InstaEdit, an acceleration framework for image editing models that reduces the inference steps while preserving, or even improving, generation quality. Unlike existing approaches, the proposed method employed a noise prediction strategy that incorporates both textual instructions and visual conditions into the noise initialization process, producing latent variables that are closer to intermediate states along the generative trajectory. These optimized latent variables were then used as the starting point for the sampling process. The key component of the framework is a deep noise predictor that estimates a more suitable initial noise state for the generation procedure, thereby reducing redundant denoising steps and improving inference efficiency. InstaEdit can be seamlessly integrated into existing mainstream Flow Matching inference pipelines. We applied InstaEdit to several state-of-the-art image editing models, including Step1X-Edit, FLUX.1-Kontext, and FLUX.2-klein-base-9B. Experimental results demonstrated that InstaEdit achieved acceleration factors of 4.13×, 3.88×, and 4.22×, respectively, while maintaining stable and high-quality generative performance across multiple standard image editing benchmarks.

Keywords: Flow-Matching · Image Editing · Acceleration

1 Introduction

Image editing is now widely regarded as a fundamental capability of modern generative models, allowing users to modify images according to textual instructions while preserving the original visual content [21,42,44]. With the rapid progress of diffusion and flow matching frameworks, both the quality and controllability of instruction-guided image editing have improved considerably. Nevertheless, the inference procedure of these models remains computationally demanding, mainly because generating a final result requires a large number of sequential denoising steps [13,66]. As model scales and image resolutions continue to increase, the associated computational cost grows accordingly, resulting in noticeable inference latency and reduced deployment efficiency in practical applications.

Improving the inference efficiency of image editing models has become an important research direction. Most existing acceleration techniques are adapted from general image generation strategies [5, 22, 46, 50, 52, 75], including pruning,

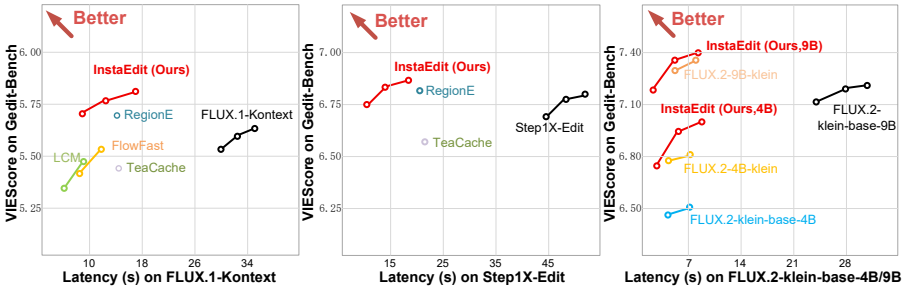

Fig. 1: Quality-latency comparison of different image edit models. Visual quality versus latency curves of the proposed InstaEdit approach and other accelerate methods in both visual quality and efficiency. Latency is evaluated on a single A100 GPU under 1024 $\times$ 1024 resolution.

quantization, and distillation to reduce sampling steps. While effective to some extent, these methods do not fully exploit the characteristics of image editing, which requires both semantic consistency and structural preservation, often leading to suboptimal quality. Some methods tailored for image editing accelerate inference by modifying or compressing the generative trajectory [4, 9, 24, 27, 40, 41, 54, 68], but typically at the cost of fidelity.

Recent studies indicated that carefully optimized initial noise could noticeably influence the quality of generated images [2, 7, 11, 17, 36, 43, 70, 77]. These findings suggest that there exists a better initialization state during the generation process. Instead, it can be interpreted as a latent state corresponding to a particular timestep along the generative trajectory [12,15,33,43,45,56,64]. When the initial noise is closer to an optimal intermediate state, model can reach the final result with fewer denoising iterations. This observation naturally raises an important question: can the process of obtaining such states be formulated as a learnable task? If a model could directly predict a suitable intermediate latent state and use it as the initial noise for inference, redundant evolution might be avoided, thereby reducing sampling steps while maintaining generation quality.

Motivated by this perspective, we revisit the acceleration problem in image editing from the viewpoint of initialization optimization and introduce InstaEdit, a noise prediction module. InstaEdit combines information from reference images and textual instructions to refine the initial noise, producing latent representations that lie closer to the desired generative trajectory. This improved starting point enables the solver to converge with fewer iterations, thereby reducing overall inference cost. Experimental evaluations show that even with substantially fewer denoising steps, InstaEdit maintains generation quality comparable to that of the original baseline models. We integrate InstaEdit with several recent state-of-the-art image editing frameworks, including Step1X-Edit [35], FLUX.1-Kontext [29], and FLUX.2-klein-base-9B [28]. The proposed approach achieves

acceleration factors of 4.13x, 3.88x, and 4.22x, respectively, while demonstrating stable performance across multiple widely used image editing benchmarks.

Our contributions are summarized as follows:

1. We propose InstaEdit, it predicts an optimized initialization noise conditioned on both the reference image and the editing instruction, enabling high-quality image editing with only a few denoising steps.
2. We design an noise prediction architecture. The proposed framework integrates a reparameterization module and a Singular Value Prediction (SVP) module to generate high-quality initialization noise for image editing.
3. Extensive experiments on multiple state-of-the-art image editing foundation models demonstrate the effectiveness and generality of InstaEdit. InstaEdit consistently achieves approximately $4\times$ inference acceleration while maintaining editing quality across several standard image editing benchmarks.

2 Related Work

Image Editing. Image editing represents a central task in generative modeling. Early approaches built upon the U-Net [48] architecture, such as ControlNet [72], established a stable and reliable framework by introducing structured control mechanisms. At the same time, instruction-based editing methods have progressed rapidly. Representative works including InstructEdit [59], MagicBrush [71], and BrushEdit [32] adopt modular designs that combine textual prompts with spatial constraints, enabling diffusion models to perform targeted image modifications. More recently, Stable Diffusion [14, 47] has provided the foundational pipeline for image editing in latent space, while Diffusion Transformer (DiT) [42] exploits the global modeling capacity of Transformer architectures to improve the quality of high-resolution image editing. FLUX [28, 29] further integrates flow matching with Transformer models, achieving a more balanced trade-off between generation speed and output fidelity. In addition, the Qwen-Image series [61] incorporates multimodal understanding with precise editing capabilities, demonstrating strong performance in tasks such as text rendering, semantic modification, and appearance preservation.

Efficient diffusion models. Research on efficient diffusion models has become an important direction for accelerating general-purpose diffusion frameworks. Existing studies explore several strategies. For example, pruning-based approaches such as LD-Pruner [6] reduce model complexity, while quantization methods including PTQ4DM [53], FPQuant [76], and SVDQuant [30] lower computational precision to improve efficiency. Distillation-based techniques, such as BK-SDM [22], LCM [37], Add [50] and Clear [50], aim to compress the sampling process, and sparse attention mechanisms—including DiTFastAttn [69], SVG [62], Sparse-vDiT [10], and VORTA [55]—optimize the attention computation within DiTs. In addition, early stopping strategies such as ES-DDPM [38] further reduce unnecessary computation during inference. To address temporal redundancy, several works such as DeepCache [39], Δ-DiT [8], Fora [51], and

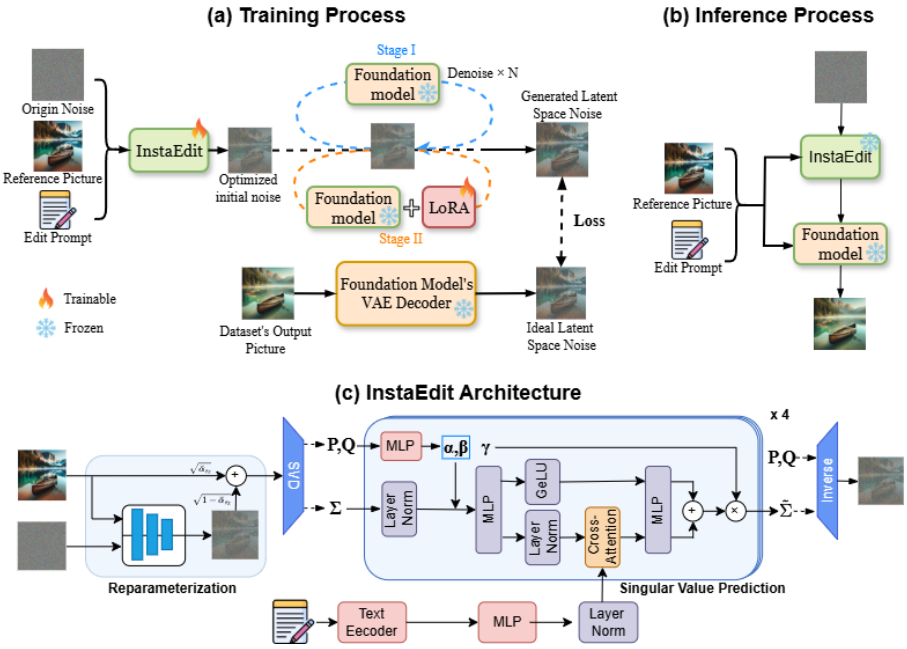

Fig. 2: Our InstaEdit method directly transforms random Gaussian noise into a more optimal noise representation, which is then fed into the image editing model to accelerate the inference process. The training procedure consists of two stages. In Stage I, only the InstaEdit module is trained. In Stage II, the InstaEdit module continues to be optimized while the Foundation model is further fine-tuned using LoRA. Subfigure (a) illustrates the overall training pipeline, while subfigure (b) presents the inference pipeline. The complete architecture of InstaEdit is detailed in subfigure (c).

TeaCache [34] reuse intermediate representations across diffusion steps. For image editing tasks, RegionE [9] exploits the distinctive trajectory characteristics of editing processes while jointly optimizing spatial and temporal redundancy. NP-Edit [27] utilizes gradient signals from vision-language models to supervise the training of editing models, thereby removing the need for paired training data. FastFlow [4] estimates prediction velocities using finite-difference approximations, improving generation efficiency without introducing substantial additional computational cost.

3 Method

This section introduces the InstaEdit method. Our core idea is not to directly compress the denoising trajectory, but to learn a better initialization for the generation process, thereby shifting the starting point closer to the target solution space and reducing the number of required sampling steps for more efficient image editing. To this end, we design a dedicated network that takes the original

noise, reference image, and text prompt as inputs, and predicts an optimized noise initialization, which is then fed into the backbone generative model for standard denoising. The overall framework is illustrated in Figure 2 and Section 3.1. Section 3.2 further details the training and inference procedures.

3.1 Model Architecture

In this section, we describe the InstaEdit architecture, which is composed of two main components: a reparameterization module and a Singular Value Prediction (SVP) module. The reparameterization module incorporated reference image information into the original noise, thereby introducing global layout constraints into the generation process. The SVP module further integrated textual information to structurally model the noise representation, enhancing stability and providing improved initial conditions for the subsequent generation stage.

Reparameterization Module. To enable reference-guided image editing, this module incorporated visual cues from the reference image into the noise initialization, thereby establishing the basic layout structure of the generation process. Following the reparameterization strategy commonly used in Variational Autoencoders (VAE) [23], the module predicted the mean and variance of a Gaussian distribution instead of directly estimating noise values. This formulation promoted distributional consistency and contributes to stable training. The predicted parameters were then used to adjust the original noise z_{ori}, guiding it toward the distribution implied by the reference image while maintaining probabilistic coherence. In practice, this procedure can be interpreted as applying a controlled perturbation to the original noise, resulting in a refined noise representation. The corresponding mathematical formulation is given as:

$$z_{\text{rep}} = \sqrt{\bar{\alpha}_{\tau_S}} x_{\text{ref}} + \sqrt{1 - \bar{\alpha}_{\tau_S}} f_{\text{ref}}(x_{\text{ref}}, z_{\text{ori}}, \tau_S), \tag{1}$$

where x_{ref} denotes the reference image; while $f_{\text{ref}}(x_{\text{ref}}, z_{\text{ori}}, \tau_S)$ denote the noise map predicted by the network based on the reference image x_{ref} and the scheduled starting timestep τ_S, respectively.

From an architectural perspective, this module was built upon the VQ-GAN [57] encoder structure and contains two down-sampling blocks, each equipped with a self-attention mechanism [58]. Such a design enabled the model to capture long-range dependencies within the image, allowing both global structure and semantic information to be effectively incorporated during noise generation. Through the resampling process, controlled perturbations were introduced into the original noise distribution.

Singular Value Prediction Module. Since the latent space is high-dimensional and structural as well as semantic information are highly entangled, directly predicting the target noise within the latent space is inherently difficult. Prior studies [1, 3, 49, 77] indicated that decomposing noise into multiple components facilitated more precise processing and reconstruction, which in turn improves both recovery accuracy and stability. Following this idea, we first applied Singular Value Decomposition (SVD) to the noise obtained from the previous stage,

decomposing it into three components: U , Σ, and $V^\top$. From the perspective of spectral stability, the singular vector subspaces (U and $V^\top$)) tend to remain relatively stable, whereas most variations are reflected in the singular value matrix Σ [31, 63]. Based on this observation, our model focuses on learning and predicting the $\tilde{\Sigma}$ corresponding to the target noise, while keeping the singular vectors fixed. This design significantly simplifies the learning problem. In practice, U and $V^\top$ provide stable spatial constraints that guide the update of the Σ within the existing structural subspace. Meanwhile, the frozen text encoder of the Foundation model was employed to extract textual features. The text prompt c, processed using Classifier-Free Guidance (CFG), was incorporated into the noise representation. Through a cross-modal interaction mechanism, the model conditionally adjusted the Σ. Afterward, the inverse SVD transformation was applied to reconstruct the target noise. By optimizing the singular value spectrum under a fixed structural subspace, the method efficiently reconstructs the target noise while maintaining semantic consistency.

We denote $\rho(\cdot)$ as the SVD function, $\rho(\cdot)^{-1}$ denotes its inverse transformation; $F(\cdot,\cdot)$ denotes cross-modal interaction mechanism; $\mathcal{E}$ denotes the frozen text encoder; and σ denotes AdaGroupNorm, which helps stabilize training. The formulation of framework can be written as follows:

$$\mathbf{e} = \sigma(\mathbf{z}_{\text{ori}}, \mathcal{E}(\mathbf{c})), \quad P, \Sigma, Q^\top = \rho(\mathbf{z}_{\text{rep}}), \tag{2}$$

$$\tilde{\Sigma} = F(\Sigma, e), \quad \mathbf{z}'_{\text{svp}} = \rho(P, \tilde{\Sigma}, Q^\top)^{-1}, \tag{3}$$

where e denotes the normalized text embedding and $\mathbf{z}'_{\text{svp}}$ denotes the predicted target noise.

3.2 Model Training and Inference

Figure 2 clearly illustrates the model's training and inference processes.

Training Process. Given the dataset $\mathcal{D} := \{x_{\text{ref}i}, c_i, x_{\text{tar}i}\}_{i=1}^{|\mathcal{D}|}$, the reference image x_{ref}, prompt c, and original noise z_{ori} were fed into the InstaEdit Φ to obtain the optimized noise z_f. Subsequently, the t denoising steps on z_f were performed using the foundation model Γ to get the predicted latent state z_{pre}. Meanwhile, the target reference image x_{tar} was encoded through the Variational Autoencoder (VAE) of the foundation model to acquire the target latent state z_{tar}. During training, each data sample is reused with different numbers of denoising steps, which improves the model's adaptability to various inference steps. The training objective is to optimize the model such that the z_f is as close as possible to the z_{tar} decoded from the target reference image. The general formula for the noise initializer learning task is expressed as follows:

$$\Phi^* = \arg\min_{\phi} \mathbb{E}_{\mathcal{D}}[\ell(\Gamma(\Phi(x_{\text{ref}_i}, c_i, z_{\text{ori}}), t), z_{\text{tar}_i})]. \tag{4}$$

In the first stage, all parameters of the Foundation model were frozen, and only the InstaEdit module was trained. This stage allowed InstaEdit to learn

Fig. 3: Examples of edited images by InstaEdit and baseline on FLUX.1-Kontext.

structured optimization of the initial noise under a fixed generative backbone. In the second stage, Low-Rank Adaptation (LoRA) [18] was introduced to perform lightweight fine-tuning of the Foundation model, further enhancing the synergy between InstaEdit and the Foundation model. This design enabled improved adaptation while maintaining parameter efficiency.

Loss Function. To train InstaEdit, the L2 loss $\mathcal{L}_2$, LPIPS [73] loss $\mathcal{L}_{\text{LPIPS}}$, and GAN [25] loss $\mathcal{L}_{\text{GAN}}$ were adopted, following common practices in recent studies [60, 70, 77]. In addition, we observed that generation performance tended to remain stable when the predicted noise approximately follows a normal distribution with a mean of 0 and a variance of 0.1. In practical inference scenarios, however, the model output may occasionally deviate from this desired distribution. To address this issue, we introduced an additional regularization term $\mathcal{L}_{\text{reg}}$ that constrains the statistical properties of the generated noise. Specifically, let the mean and standard deviation of the model output be: $\mu_\theta(\mathbf{z}_t)$, $\sigma_\theta(\mathbf{z}_t)$. Given

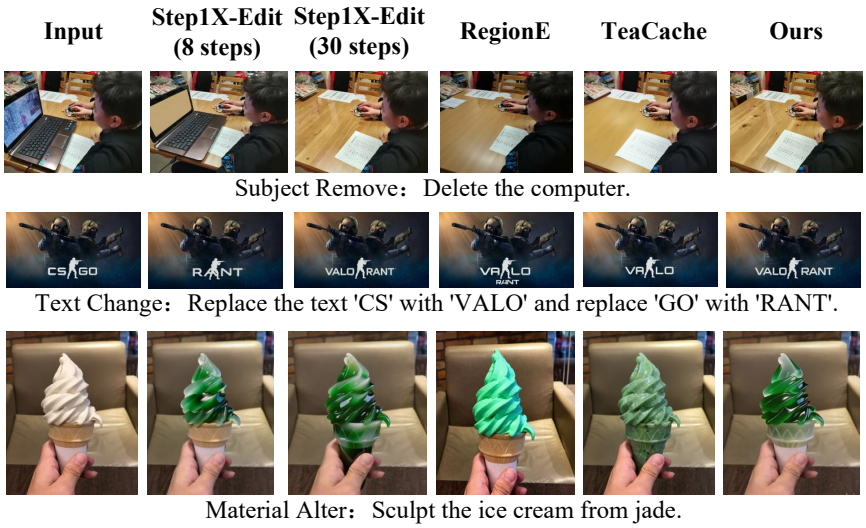

Fig. 4: Examples of edited images by InstaEdit and baseline on Step1X-Edit.

the target statistics: μ_{tgt}, σ_{tgt}. The regularization loss is defined as follows:

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{reg}} \Big[\big\| \mu_\theta(\mathbf{z}_t) - \mu_{\text{tgt}} \big\|_2^2 + \big\| \sigma_\theta(\mathbf{z}_t) - \sigma_{\text{tgt}} \big\|_2^2 \Big], \tag{5}$$

where λ_{reg} is a tunable weight coefficient that controls the relative contribution of the regularization term with respect to the other loss functions.

For training convenience and improved computational efficiency, all losses are computed in the latent space. Furthermore, the LPIPS loss is fine-tuned in the latent space to better adapt to our model architecture and training objectives.

Inference Process. Our model effectively incorporates reference images x_{ref} and editing prompts c into noise z_{ori}, converting it into high-quality noise z_{f}, thereby providing more optimal initial conditions for image editing tasks and enables efficient acceleration. InstaEdit can be directly integrated at the front end of the Foundation model without requiring substantial modifications to the original codebase. Furthermore, we will release the model weights for both training stages along with reference implementations, facilitating rapid integration and reproducibility across different Foundation models and application scenarios.

4 Empirical Analysis

4.1 Experimental Setup

This section presents an empirical evaluation of the effectiveness, generalization capability, and efficiency of InstaEdit. Extensive experiments were conducted

Fig. 5: Examples of edited images by InstaEdit and baseline on FLUX.2-klein-4B/FLUX.2-klein-9B.

across a range of image editing models to comprehensively assess its performance. Due to space limitations, additional experimental results are provided in the supplementary materials.

Foundation Model. This study evaluated InstaEdit on three open-source state-of-the-art (SOTA) models: Step1X-Edit-v1p1 [35], FLUX.1-Kontext [29], and FLUX.2-klein-4B/9B [28]. For Step1X-Edit, the CFG scale was set to 6; for FLUX.1-Kontext, the scale was set to 2.5; and for FLUX.2-klein-base-4B/9B, the scale was set to 1. All models were evaluated using 30 sampling steps.

Baseline. For FLUX.1-Kontext, RegionE [9], TeaCache [34], and FlowFast [4] were selected as baseline methods for comparison. To assess the effectiveness of distillation-based strategies, we selected LCM [37] as a representative distilled variant. For Step1X-Edit, RegionE and TeaCache were adopted as the baseline methods for evaluation. For FLUX.2-klein-4B/9B-Base, since no dedicated acceleration methods have been specifically designed for this model, we used its official distilled variants, FLUX.2-4B/9B-klein, as the comparison baselines.

Training Data. We aggregated several publicly available image editing datasets, including ImgEdit [67], AnyEdit [20], UltraEdit [74], SEED-Data-Edit [16], and HQ-Edit [19]. Only single-turn editing samples were retained. For each generated image, we computed the GPT Score [26] and Fake Score [65], which measure instruction adherence and visual realism, respectively. Based on these two metrics, we filtered samples that demonstrated strong editing performance, high

semantic consistency, and stable visual quality. The resulting dataset contains approximately one million samples spanning 10 editing categories, providing diverse and high-quality data for model training and evaluation. We will provide detailed information about the datasets in the supplementary materials.

Benchmark. For evaluation, we employed three datasets: the English subset of GEdit-Benchmark [35], ImgEdit [67], and Kontext-bench [29]. GEdit-Benchmark contains 606 image-prompt pairs covering 11 editing tasks. ImgEdit provides three evaluation splits, including a base test set, a high-difficulty single-round test set, and a dedicated multi-round test set. Kontext-bench consists of 1026 image-prompt pairs spanning five editing tasks.

Metrics. Following prior work, we adopted the VIEScore [26] metric based on GPT-4.0 to evaluate editing performance. This metric assesses each edited image from two perspectives: (1) Semantic Consistency (SC), which measures the degree to which the generated result follows the editing instructions, and (2) Perceived Quality (PQ), which evaluates visual realism and the absence of artifacts. After computing the VIEScore, the overall score was obtained by taking the geometric mean of the SC and PQ scores for each image and then averaging the results across all evaluation benchmarks. For efficiency evaluation, we revealed both the actual runtime latency and the relative speedup compared with standard pre-trained models. Due to space limitations, we will include additional qualitative analyses of more metrics in the supplementary material.

Implementation Details. All models were trained on 8 NVIDIA A100 GPUs. During evaluation, experiments are conducted on a NVIDIA A100 GPU. To ensure fairness and reproducibility, all latency measurements are reported under a single-GPU setting, with each run processing one image at a time.

4.2 Main Results

Performance on GEdit-Bench. Table 1 outlines the quantitative evaluation results on GEdit-Bench. Under the few-step inference setting, InstaEdit demonstrated consistent performance improvements across multiple mainstream Foundation models. Compared with the original inference pipeline, the proposed method substantially reduced the number of sampling steps while achieving comparable—or even superior—results in terms of SC Score, PQ Score, and the overall metric. For example, when applied to strong baselines such as FLUX.2-klein-9B and Step1X-Edit, InstaEdit maintained or even improved the overall score under a 4-step inference setting. These results indicated that the proposed noise optimization strategy effectively preserved both generation fidelity and semantic consistency, even in low-step regimes. Moreover, compared with existing acceleration methods, InstaEdit achieves comparable or even better performance while using fewer sampling steps and shorter inference time, demonstrating a more favorable performance–efficiency trade-off.

Performance on ImgEdit-Bench. Table 2 evaluates the effectiveness of InstaEdit on ImgEdit-Bench. The proposed approach achieved a more favorable trade-off between efficiency and generation quality. Across both the FLUX and Step1X-Edit model families, InstaEdit consistently improved SC Score and PQ

Table 1: Quantitative evaluation on GEdit-Bench. Our method performs on par or better than baselines under the few-step setting.

Method	Step	Time(s)	Speed↑	SC Score↑	PQ Score↑	overall↑
FLUX.1-Kontext	30	34.91	1×	6.30	6.65	5.65
Step1X-Edit	30	52.64	1×	7.31	7.37	6.79
FLUX.2-klein-base-4B	30	14.22	1×	7.22	7.18	6.64
FLUX.2-klein-base-9B	30	31.09	1×	7.56	7.41	7.24
FLUX.1-Kontext	8	9.65	3.61×	5.94	5.86	5.21
+ Distillation	8	9.57	3.64×	6.11	6.07	5.40
+ RegionE	-	14.54	2.40×	**6.34**	6.57	5.65
+ TeaCache	-	14.75	2.37×	6.32	6.41	5.38
+ FlowFast	10	**8.77**	**3.98×**	6.12	6.06	5.45
+ InstaEdit (Ours)	4	8.98	3.88×	6.33	**6.61**	**5.67**
Step1X-Edit	8	13.27	3.87×	6.91	6.83	6.41
+ RegionE	-	20.48	2.57×	**7.35**	7.29	6.82
+ TeaCache	-	21.14	2.49×	7.21	7.08	6.67
+ InstaEdit (Ours)	4	**12.75**	**4.13×**	7.34	**7.32**	**6.84**
FLUX.2-klein-4B	8	**4.04**	**3.52×**	7.29	7.23	6.77
FLUX.2-klein-base-4B	8	4.07	3.49×	6.82	6.74	6.31
+ InstaEdit (Ours)	4	5.53	2.57×	**7.41**	**7.40**	**6.95**
FLUX.2-klein-9B	8	7.38	4.21×	7.73	7.50	7.36
FLUX.2-klein-base-9B	8	7.44	4.18×	7.19	7.07	6.62
+ InstaEdit (Ours)	4	**7.37**	**4.22×**	**7.78**	**7.52**	**7.40**

Score while significantly reducing the number of sampling steps, achieving state-of-the-art or tied-best performance on the overall metric among all compared methods. For example, on FLUX.2-klein-9B, InstaEdit achieved an overall score of 7.48 with only four inference steps, surpassing the corresponding eight-step baseline result. These results demonstrated the strong capability of InstaEdit to preserve semantic consistency and structural fidelity even under low-step inference settings.

Efficiency Analysis. InstaEdit obtained substantial inference acceleration while maintaining high-quality editing performance, offering a strong balance between efficiency and effectiveness. Specifically, it delivered speedups of 4.13×, 3.90×, and 3.98× on Step1X-Edit, FLUX.1-Kontext, and FLUX.2-klein-base-9B, respectively. In terms of absolute latency, the inference time was reduced from 52.67s to 8.94s, 34.89s to 8.94s, and 30.99s to 7.77s, demonstrating consistent and practically meaningful improvements in efficiency. On the efficiency-quality trade-off curve, InstaEdit exhibited a more favorable balance compared with competing approaches.

InstaEdit introduced a additional inference overhead of approximately 2-4 seconds on A100 GPUs. However, this overhead was acceptable, as InstaEdit significantly reduces the number of subsequent sampling steps in the overall

Table 2: Quantitative evaluation on ImgEdit-Bench. Our method performs on par or better than baselines under the few-step setting.

Method	Step	Time(s)	Speed↑	SC Score↑	PQ Score↑	overall↑
FLUX.1-Kontext	30	34.89	1x	6.94	6.73	6.47
Step1X-Edit	30	52.67	1x	7.26	7.30	6.72
FLUX.2-klein-base-4B	30	14.25	1x	7.29	7.25	6.71
FLUX.2-klein-base-9B	30	30.99	1x	7.67	7.46	7.30
FLUX.1-Kontext	8	9.65	3.61×	6.37	5.98	5.78
+ LCM	8	9.57	3.64×	6.52	6.23	5.94
+ RegionE	-	14.54	2.40×	**7.06**	6.74	6.51
+ TeaCache	-	14.75	2.37×	7.02	6.62	6.18
+ FlowFast	10	**8.77**	**3.98×**	6.71	6.31	6.13
+ InstaEdit (Ours)	4	8.94	3.90×	7.05	**6.78**	**6.57**
Step1X-Edit	8	13.27	3.95×	6.88	6.78	6.31
+ RegionE	-	21.14	2.48×	**7.29**	7.32	**6.75**
+ TeaCache	-	20.48	2.55×	7.02	7.09	6.57
+ InstaEdit (Ours)	4	**12.72**	**4.13×**	7.28	**7.36**	6.73
FLUX.2-klein-4B	8	4.06	3.51x	7.34	7.27	6.94
FLUX.2-klein-base-4B	8	**4.04**	**3.53x**	6.87	6.79	6.39
+ InstaEdit (Ours)	4	5.51	2.58x	**7.41**	**7.32**	**6.99**
FLUX.2-klein-9B	8	**7.38**	**4.21x**	7.81	7.55	7.39
FLUX.2-klein-base-9B	8	7.44	4.16x	7.38	7.24	6.81
+ InstaEdit (Ours)	4	7.48	4.21x	**7.83**	**7.56**	**7.47**

inference pipeline. Consequently, the model achieved comparable—or even superior—editing performance with similar or shorter total runtime. In other words, a small localized computational cost lead to a global improvement in efficiency and performance. Moreover, the additional parameters introduced by InstaEdit accounted for less than 5% of the original model parameters, resulting in minimal overhead in terms of model capacity. This highlighted the strong parameter efficiency and deployment practicality of the proposed method.

Visualization of Generated Results. Figures 3 to 5 illustrate the generated results. The proposed model generated images with high visual fidelity, with some results even surpassing those produced by the original baseline models. Beyond improving speed, our approach also demonstrated strong capability in preserving the shape and distinctive characteristics of the primary subject in the reference image. These observations further emphasized the effectiveness and advantages of the proposed architecture.

4.3 Ablation Studies

Table 3 summarizes an ablation study of the InstaEdit architecture. The results indicated that both core components contributed significantly to the over-

Table 3: Ablation studies of the proposed components on GEdit-Bench.

Method	Step	Time(s)	SC Score↑	PQ Score↑	overall↑
FLUX.1-Kontext	30	34.91	6.30	6.65	5.65
FLUX.1-Kontext	8	9.65	5.94	5.86	5.21
w/o Rep	4	7.81	6.22	6.07	5.35
w/o SVP	4	6.96	5.99	6.32	5.27
w/o L_{reg}	4	8.99	6.18	6.43	5.48
InstaEdit (Ours)	4	**8.98**	**6.33**	**6.61**	**5.67**
Stage I	4	8.97	6.24	6.49	5.51
Stage II	4	8.98	6.33	6.61	5.67

all model performance. The reparameterization module established the global layout structure of the image, while the SVP module enhanced semantic editing capability while maintaining the stability of the noise structure. These two components complement each other and jointly improved the overall performance of the model. Additionally, more detailed ablation experiments are provided in the appendix, further validating the contribution of each component to model training and performance improvement.

Moreover, the staged training strategy also improves the synergy between InstaEdit and the Foundation model. In Stage I, InstaEdit enhances performance by optimizing the noise initialization; however, since the Foundation model remains frozen, the overall improvement is limited. In Stage II, by fine-tuning the Foundation model, InstaEdit achieves higher generation quality and demonstrates significant performance gains.

4.4 Other Experiment

Comparison with Dynamic Methods. Table 4 outlines a comparative analysis of several dynamic acceleration methods, including InstaEdit, LCM, and FlowFast, under different sampling step configurations. The results indicated that InstaEdit consistently maintained stable and competitive generation quality across various step settings. Compared with the other methods, the proposed approach effectively alleviated performance degradation while reducing the number of sampling steps, achieving a more favorable trade-off between inference efficiency and generation quality.

Orthogonality with Distillation Method. Table 5 evaluates InstaEdit on the FLUX.2-klein-4B model, which has already incorporated distillation for acceleration, to examine its compatibility with distilled models and its potential for additional acceleration. The experimental results demonstrated that InstaEdit was complementary to existing distillation strategies. Specifically, when applied on top of the distilled model, InstaEdit was able to achieve performance comparable to the original distilled model while using fewer sampling steps. However, since the distilled model has already achieved high inference efficiency, the ad-

Table 4: Comparison with dynamic acceleration methods.

Method	Step	Time(s)	Speed↑	SC Score↑	PQ Score↑	overall↑
FLUX.1-Kontext	30	34.91	1×	6.30	6.65	5.65
FLUX.1-Kontext	8	9.65	3.61×	5.94	5.86	5.21
+ LCM	8	**9.57**	**3.64×**	6.11	6.07	5.60
+ FlowFast	20	15.54	2.25×	6.28	6.53	5.67
+ InstaEdit (Ours)	8	13.04	2.67×	**6.52**	**6.78**	**5.75**
FLUX.1-Kontext	4	4.98	7.01×	5.75	5.59	5.01
+ LCM	4	**4.96**	**7.03×**	5.95	5.89	5.23
+ FlowFast	10	8.77	3.98×	6.12	6.06	5.45
+ InstaEdit (Ours)	4	8.98	3.88×	**6.33**	**6.61**	**5.67**

Table 5: Orthogonality studies on FLUX.2-klein-9B.

Method	Step	Time(s)	Speed↑	SC Score↑	PQ Score↑	overall↑
FLUX.2-klein-9B	8	7.38	1×	**7.73**	**7.50**	**7.36**
+ InstaEdit (Ours)	3	6.34	1.16×	7.71	**7.50**	7.35

ditional reduction in overall inference time remained relatively limited. Future work may explore optimizations specifically designed for distilled models to further exploit the potential benefits of combining these two approaches.

Due to space constraints, we will include the evaluation results on KontextBench and additional generated examples in the supplementary material. We will also provide a more detailed comparison with existing methods.

5 Conclusion

This paper proposes a noise prediction approach that provides an efficient acceleration solution for image editing. By integrating textual and visual information into the initialization noise, the proposed method estimated optimal intermediate latent states along the forward trajectory, enabling the model to bypass redundant inference steps adaptively. As a result, the approach substantially improved generation efficiency while maintaining strong editing and generation performance. Extensive experiments demonstrated the effectiveness of InstaEdit. With negligible loss in visual quality, InstaEdit achieved end-to-end acceleration of 4.13×, 3.90× and 2.58x/4.22× on Step1X-Edit, FLUX.1-Kontext, and FLUX.2-klein-base-4B/9B, respectively. These results highlighted the simplicity and general applicability of InstaEdit. Future work will extend the initialization prediction strategy proposed in this paper to other generative tasks.

References

1. Abdi, H.: Singular value decomposition (svd) and generalized singular value decomposition. Encyclopedia of measurement and statistics **907**(912), 44 (2007)
2. Ahn, D., Kang, J., Lee, S., Min, J., Kim, M., Jang, W., Cho, H., Paul, S., Kim, S., Cha, E., et al.: A noise is worth diffusion guidance. arXiv preprint arXiv:2412.03895 (2024)
3. Akritas, A.G., Malaschonok, G.I.: Applications of singular-value decomposition (svd). Mathematics and computers in simulation **67**(1-2), 15–31 (2004)
4. Bajpai, D.J., Bhardwaj, D., Roy, S., Duseja, T., Agarwal, H., Sandansing, A., Hanawal, M.K.: Fastflow: Accelerating the generative flow matching models with bandit inference. arXiv preprint arXiv:2602.11105 (2026)
5. Castells, T., Song, H.K., Kim, B.K., Choi, S.: Ld-pruner: Efficient pruning of latent diffusion models using task-agnostic insights. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 821–830 (2024)
6. Castells, T., Song, H.K., Kim, B.K., Choi, S.: Ld-pruner: Efficient pruning of latent diffusion models using task-agnostic insights. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 821–830 (2024)
7. Chefer, H., Alaluf, Y., Vinker, Y., Wolf, L., Cohen-Or, D.: Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. ACM transactions on Graphics (TOG) **42**(4), 1–10 (2023)
8. Chen, P., Shen, M., Ye, P., Cao, J., Tu, C., Bouganis, C.S., Zhao, Y., Chen, T.: Delta-dit: A training-free acceleration method tailored for diffusion transformers. arXiv preprint arXiv:2406.01125 (2024)
9. Chen, P., Zeng, X., Zhao, M., Shen, M., Ye, P., Xiang, B., Wang, Z., Cheng, W., Yu, G., Chen, T.: Regione: Adaptive region-aware generation for efficient image editing. arXiv preprint arXiv:2510.25590 (2025)
10. Chen, P., Zeng, X., Zhao, M., Ye, P., Shen, M., Cheng, W., Yu, G., Chen, T.: Sparse-vdit: Unleashing the power of sparse attention to accelerate video diffusion transformers. arXiv preprint arXiv:2506.03065 (2025)
11. Chen, S.X., Vaxman, Y., Ben Baruch, E., Asulin, D., Moreshet, A., Lien, K.C., Sra, M., Sen, P.: Tino-edit: Timestep and noise optimization for robust diffusion-based image editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6337–6346 (2024)
12. Chen, X., Feng, Y., Chen, M., Wang, Y., Zhang, S., Liu, Y., Shen, Y., Zhao, H.: Zero-shot image editing with reference imitation. Advances in Neural Information Processing Systems **37**, 84010–84032 (2024)
13. Davtyan, A., Dadi, L.T., Cevher, V., Favaro, P.: Faster inference of flow-based generative models via improved data-noise coupling. In: The Thirteenth International Conference on Learning Representations (2025)
14. 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)
15. Fu, F., Zhang, L., Huang, M., Mao, Z.: Feededit: Text-based image editing with dynamic feedback regulation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 2661–2670 (2025)
16. Ge, Y., Zhao, S., Li, C., Ge, Y., Shan, Y.: Seed-data-edit technical report: A hybrid dataset for instructional image editing. arXiv preprint arXiv:2405.04007 (2024)

17. Guo, X., Liu, J., Cui, M., Li, J., Yang, H., Huang, D.: Initno: Boosting text-to-image diffusion models via initial noise optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9380–9389 (2024)
18. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. Iclr **1**(2), 3 (2022)
19. Hui, M., Yang, S., Zhao, B., Shi, Y., Wang, H., Wang, P., Zhou, Y., Xie, C.: Hq-edit: A high-quality dataset for instruction-based image editing. arXiv preprint arXiv:2404.09990 (2024)
20. Jiang, H., Fang, J., Zhang, N., Ma, G., Wan, M., Wang, X., He, X., Chua, T.s.: Anyedit: Edit any knowledge encoded in language models. arXiv preprint arXiv:2502.05628 (2025)
21. Kang, M., Zhu, J.Y., Zhang, R., Park, J., Shechtman, E., Paris, S., Park, T.: Scaling up gans for text-to-image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10124–10134 (2023)
22. Kim, B.K., Song, H.K., Castells, T., Choi, S.: Bk-sdm: Architecturally compressed stable diffusion for efficient text-to-image generation. In: Workshop on Efficient Systems for Foundation Models@ ICML2023 (2023)
23. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
24. Koo, G., Yoon, S., Yoo, C.D.: Wavelet-guided acceleration of text inversion in diffusion-based image editing. In: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4380–4384. IEEE (2024)
25. Krichen, M.: Generative adversarial networks. In: 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT). pp. 1–7. IEEE (2023)
26. Ku, M., Jiang, D., Wei, C., Yue, X., Chen, W.: Viescore: Towards explainable metrics for conditional image synthesis evaluation. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 12268–12290 (2024)
27. Kumari, N., Wang, S.Y., Zhao, N., Nitzan, Y., Li, Y., Singh, K.K., Zhang, R., Shechtman, E., Zhu, J.Y., Huang, X.: Learning an image editing model without image editing pairs. arXiv preprint arXiv:2510.14978 (2025)
28. Labs, B.F.: FLUX.2: Frontier Visual Intelligence. `https://bfl.ai/blog/flux-2` (2025)
29. Labs, B.F., Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., et al.: Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. arXiv preprint arXiv:2506.15742 (2025)
30. Li, M., Lin, Y., Zhang, Z., Cai, T., Li, X., Guo, J., Xie, E., Meng, C., Zhu, J.Y., Han, S.: Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models. arXiv preprint arXiv:2411.05007 (2024)
31. Li, R.C.: Matrix perturbation theory. Handbook of linear algebra pp. 15–1 (2006)
32. Li, Y., Bian, Y., Ju, X., Zhang, Z., Zhuang, J., Shan, Y., Zou, Y., Xu, Q.: Brushedit: All-in-one image inpainting and editing. arXiv preprint arXiv:2412.10316 (2024)
33. Lin, H., Chen, Y., Wang, J., An, W., Wang, M., Tian, F., Liu, Y., Dai, G., Wang, J., Wang, Q.: Schedule your edit: A simple yet effective diffusion noise schedule for image editing. Advances in Neural Information Processing Systems **37**, 115712–115756 (2024)

34. Liu, F., Zhang, S., Wang, X., Wei, Y., Qiu, H., Zhao, Y., Zhang, Y., Ye, Q., Wan, F.: Timestep embedding tells: It's time to cache for video diffusion model. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 7353–7363 (2025)
35. Liu, S., Han, Y., Xing, P., Yin, F., Wang, R., Cheng, W., Liao, J., Wang, Y., Fu, H., Han, C., et al.: Step1x-edit: A practical framework for general image editing. arXiv preprint arXiv:2504.17761 (2025)
36. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11461–11471 (2022)
37. Luo, S., Tan, Y., Huang, L., Li, J., Zhao, H.: Latent consistency models: Synthesizing high-resolution images with few-step inference. arXiv preprint arXiv:2310.04378 (2023)
38. Lyu, Z., Xu, X., Yang, C., Lin, D., Dai, B.: Accelerating diffusion models via early stop of the diffusion process. arXiv preprint arXiv:2205.12524 (2022)
39. Ma, X., Fang, G., Wang, X.: Deepcache: Accelerating diffusion models for free. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15762–15772 (2024)
40. Nguyen, T.T., Nguyen, Q., Nguyen, K., Tran, A., Pham, C.: Swiftedit: Lightning fast text-guided image editing via one-step diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21492–21501 (2025)
41. Pan, Z., Gherardi, R., Xie, X., Huang, S.: Effective real image editing with accelerated iterative diffusion inversion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15912–15921 (2023)
42. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)
43. Qi, Z., Bai, L., Xiong, H., Xie, Z.: Not all noises are created equally: Diffusion noise selection and optimization. arXiv preprint arXiv:2407.14041 (2024)
44. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents
45. Ren, Y., Jiang, Z., Zhang, T., Forchhammer, S., Süsstrunk, S.: Fds: Frequency-aware denoising score for text-guided latent diffusion image editing. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 2651–2660 (2025)
46. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
47. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
48. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
49. Sadek, R.A.: Svd based image processing applications: state of the art, contributions and research challenges. arXiv preprint arXiv:1211.7102 (2012)
50. Sauer, A., Lorenz, D., Blattmann, A., Rombach, R.: Adversarial diffusion distillation. In: European Conference on Computer Vision. pp. 87–103. Springer (2024)
51. Selvaraju, P., Ding, T., Chen, T., Zharkov, I., Liang, L.: Fora: Fast-forward caching in diffusion transformer acceleration. arXiv preprint arXiv:2407.01425 (2024)

52. Shang, Y., Yuan, Z., Xie, B., Wu, B., Yan, Y.: Post-training quantization on diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1972–1981 (2023)
53. Shang, Y., Yuan, Z., Xie, B., Wu, B., Yan, Y.: Post-training quantization on diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1972–1981 (2023)
54. Starodubcev, N., Khoroshikh, M., Babenko, A., Baranchuk, D.: Invertible consistency distillation for text-guided image editing in around 7 steps. Advances in Neural Information Processing Systems **37**, 12496–12527 (2024)
55. Sun, W., Tu, R.C., Ding, Y., Jin, Z., Liao, J., Liu, S., Tao, D.: Vorta: Efficient video diffusion via routing sparse attention. arXiv preprint arXiv:2505.18809 (2025)
56. Tian, F., Li, Y., Yan, Y., Guan, S., Ge, Y., Yang, X.: Postedit: Posterior sampling for efficient zero-shot image editing. arXiv preprint arXiv:2410.04844 (2024)
57. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. Advances in neural information processing systems **30** (2017)
58. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
59. Wang, Q., Zhang, B., Birsak, M., Wonka, P.: Instructedit: Improving automatic masks for diffusion-based image editing with user instructions. arXiv preprint arXiv:2305.18047 (2023)
60. Wang, X., Xie, L., Dong, C., Shan, Y.: Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1905–1914 (2021)
61. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.m., Bai, S., Xu, X., Chen, Y., et al.: Qwen-image technical report. arXiv preprint arXiv:2508.02324 (2025)
62. Xi, H., Yang, S., Zhao, Y., Xu, C., Li, M., Li, X., Lin, Y., Cai, H., Zhang, J., Li, D., et al.: Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. arXiv preprint arXiv:2502.01776 (2025)
63. Xie, Z., Tang, Q.Y., Sun, M., Li, P.: On the overlooked structure of stochastic gradients. Advances in Neural Information Processing Systems **36**, 66257–66276 (2023)
64. Xu, K., Zhang, L., Shi, J.: Good seed makes a good crop: Discovering secret seeds in text-to-image diffusion models. in 2025 ieee/cvf winter conference on applications of computer vision (wacv) (2025)
65. Xu, Z., Zhang, X., Li, R., Tang, Z., Huang, Q., Zhang, J.: Fakeshield: Explainable image forgery detection and localization via multi-modal large language models. arXiv preprint arXiv:2410.02761 (2024)
66. Yan, H., Liu, X., Pan, J., Liew, J.H., Liu, Q., Feng, J.: Perflow: Piecewise rectified flow as universal plug-and-play accelerator. Advances in Neural Information Processing Systems **37**, 78630–78652 (2024)
67. Ye, Y., He, X., Li, Z., Lin, B., Yuan, S., Yan, Z., Hou, B., Yuan, L.: Imgedit: A unified image editing dataset and benchmark. arXiv preprint arXiv:2505.20275 (2025)
68. Yu, Z., Li, H., Fu, F., Miao, X., Cui, B.: Accelerating text-to-image editing via cache-enabled sparse diffusion inference. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 16605–16613 (2024)
69. Yuan, Z., Zhang, H., Pu, L., Ning, X., Zhang, L., Zhao, T., Yan, S., Dai, G., Wang, Y.: Ditfastattn: Attention compression for diffusion transformer models. Advances in Neural Information Processing Systems **37**, 1196–1219 (2024)

70. Yue, Z., Liao, K., Loy, C.C.: Arbitrary-steps image super-resolution via diffusion inversion. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 23153–23163 (2025)
71. Zhang, K., Mo, L., Chen, W., Sun, H., Su, Y.: Magicbrush: A manually annotated dataset for instruction-guided image editing. Advances in Neural Information Processing Systems **36**, 31428–31449 (2023)
72. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3836–3847 (2023)
73. 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)
74. Zhao, H., Ma, X.S., Chen, L., Si, S., Wu, R., An, K., Yu, P., Zhang, M., Li, Q., Chang, B.: Ultraedit: Instruction-based fine-grained image editing at scale. Advances in Neural Information Processing Systems **37**, 3058–3093 (2024)
75. Zhao, M., Chen, P., Yu, C., Wen, Y., Tan, X., Chen, T.: Pioneering 4-bit fp quantization for diffusion models: Mixup-sign quantization and timestep-aware fine-tuning. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 18134–18143 (2025)
76. Zhao, M., Chen, P., Yu, C., Wen, Y., Tan, X., Chen, T.: Pioneering 4-bit fp quantization for diffusion models: Mixup-sign quantization and timestep-aware fine-tuning. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 18134–18143 (2025)
77. Zhou, Z., Shao, S., Bai, L., Zhang, S., Xu, Z., Han, B., Xie, Z.: Golden noise for diffusion models: A learning framework. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17688–17697 (2025)