This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

InverFill: One-Step Inversion for Enhanced Few-Step Diffusion Inpainting

Duc Vu[1⋆] Kien Nguyen[1⋆] Trong-Tung Nguyen[1⋆] Ngan Nguyen[1⋆]
Phong Nguyen[1] Khoi Nguyen[1] Cuong Pham[1,2] Anh Tran[1]
[1] Qualcomm AI Research† [2] Posts & Telecommunications Inst. of Tech., Vietnam
{ducvu, kienn, tunnguy, nganngguy, phongnh, khoi, pcuong, anhtra}@qti.qualcomm.com cuongpv@ptit.edu.vn

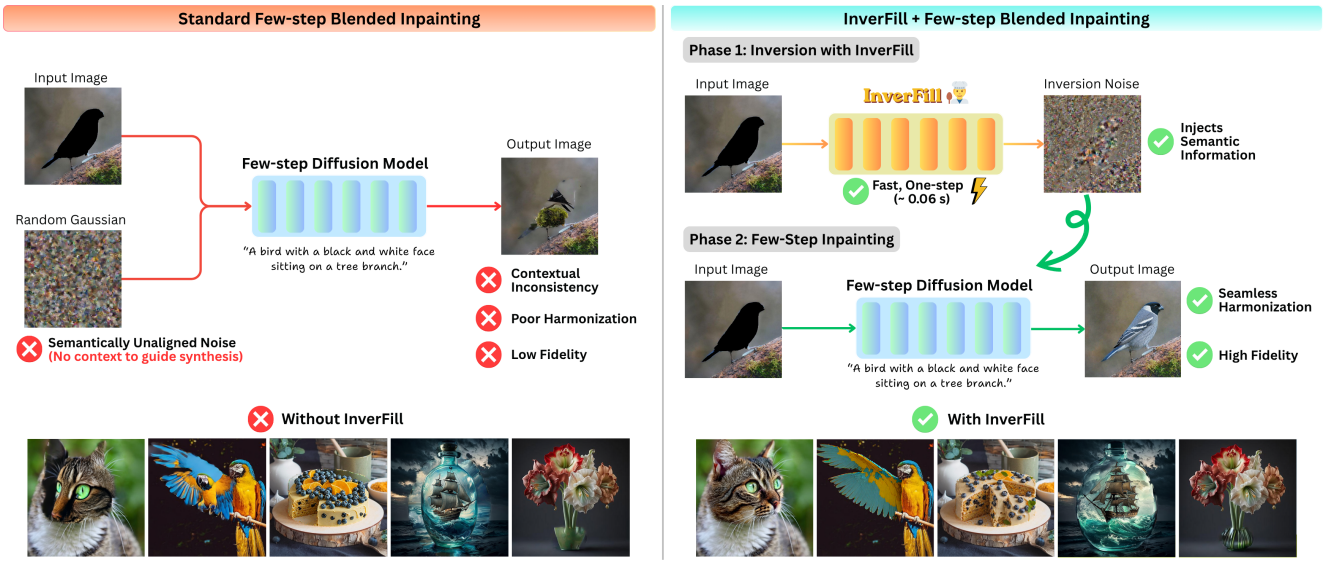


Figure 1. **InverFill** enhances few-step inpainting by generating semantically aligned inverted noise latents, while adding as little as **0.06 seconds** of overhead on a single NVIDIA A100 40GB GPU.

Abstract

Recent diffusion-based models achieve photorealism in image inpainting but require many sampling steps, limiting practical use. Few-step text-to-image models offer faster generation, but naively applying them to inpainting yields poor harmonization and artifacts between the background and inpainted region. We trace this cause to random Gaussian noise initialization, which under low function evaluations causes semantic misalignment and reduced fidelity. To overcome this, we propose InverFill, a one-step inversion method tailored for inpainting that injects semantic information from the input masked image into the initial noise, enabling high-fidelity few-step inpainting. Instead of training inpainting models, InverFill leverages few-step text-to-image models in a blended sampling pipeline with semantically aligned noise as input, significantly improving vanilla blended sampling and even matching specialized inpainting models at low NFEs. Moreover, InverFill does not require real-image supervision and only adds minimal inference overhead. Extensive experiments show that InverFill consistently boosts baseline few-step models, improving image quality and text coherence without costly retraining or heavy iterative optimization.

1. Introduction

Recent generative models enable photorealistic and detail-rich visual synthesis across many tasks [2, 30, 34, 38, 46]. Among them, text-guided image inpainting has become a key direction, aiming to fill masked regions with content that is semantically aligned with the prompt and visually consistent with the background. Progress in this area is largely driven by large-scale text-to-image diffusion models [3, 17, 38]. To adapt pretrained models for inpainting, recent methods rely on blended sampling or fine-tuning with spatially aware architectures, exploiting strong pretrained priors for seamless results. Early approaches fine-tune the full diffusion U-Net with mask conditioning [33, 38, 50], while adapter-based methods like

⋆ Equal Contribution
† Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

BrushNet [13] add lightweight trainable branches to frozen backbones. Training-free methods [11, 19, 25] instead use guided sampling or attention manipulation. Despite their effectiveness, most techniques require many sampling steps, resulting in high latency and limiting real-time deployment. This underscores the need for faster inpainting solutions.

While many few-step text-to-image diffusion models exist [4, 24, 41, 42], adapting them for image inpainting is nontrivial. A natural solution is blended sampling [1], where predictions are iteratively merged with the unmasked regions. This strategy works well for multi-step diffusion models, where gradual denoising allows the synthesized content to blend smoothly with the preserved context. In the few-step regime, however, each denoising step induces much larger updates, leading to semantic misalignment between the initial noise and the masked content, and ultimately causing poor harmonization with the surrounding background. To the best of our knowledge, TurboFill [49] is the only successful few-step, text-guided specialized inpainting model. It reduces inference steps using a 3-step adversarial scheme that trains an inpainting adapter on top of a distilled few-step text-to-image model [53]. However, this design is complex, requires real-image supervision, and is computationally heavy. Moreover, prior inpainting methods follow the standard diffusion process [10, 44], which always starts from pure Gaussian noise. This gives the model no initial clue about the semantics or structure of the unmasked regions, often causing a semantic mismatch between the inpainted content and its surrounding context. Multi-step models can gradually correct this mismatch, but few-step or one-step models have no such allowance, leaving little room to recover from the initial randomness. As a result, inpainting under low NFEs tends to produce blurry, poorly integrated regions and degraded overall fidelity.

To this end, we introduce **InverFill**, an efficient one-step inversion network that significantly improves performance of few-step inpainting with minimal overhead. As shown in Fig. 1, InverFill maps the masked image into an inverted noise latent, replacing random Gaussian initialization with a semantically informed noise for few-step inpainting. Although diffusion inversion has been explored for editing and inpainting [5, 19, 26, 27], we are the first to design a one-step inversion customized for inpainting. While SwiftEdit [31] proposes a one-step inversion framework for image editing, a naive adaptation to inpainting fails for two reasons. First, training on masked images causes substantial leakage from the visible regions into the inverted noise latent. Second, its reconstruction objective does not constrain the inverted latent to follow the required Gaussian distribution. To overcome this, we introduce Re-Blending to prevent information leakage and a Gaussian regularization loss to ensure the inverted noise latent aligns with the expected noise distribution. Our training pipeline is image-free, requiring no curated image–mask–text triplets and no multi-stage procedures. With these designs, InverFill enhances few-step inpainting and enables few-step text-to-image models to perform on par with specialized inpainting systems, without any finetuning while introducing negligible latency. Our contributions are summarized as follows:

- We propose **InverFill**, an efficient one-step inversion network for few-step image inpainting, which generates semantically informed initial noise to improve inpainting quality while introducing minimal overhead.
- We introduce the **Re-Blending** operation to mitigate information leakage during training while preserving key semantics in the inverted noise latent for inpainting.
- We introduce a **Gaussian regularization** loss to align the inverted noise latent with the expected Gaussian distribution, enhancing stability and quality.
- Our method features a **highly simplified, image-free training** pipeline that eliminates the need for image-mask-text triplets and complex multi-stage training.
- We demonstrate that InverFill significantly boosts the performance of existing few-step inpainting models and enables few-step text-to-image models to perform high-quality inpainting without any task-specific fine-tuning.

2. Related Works

2.1. Fast Text-to-image Diffusion Models

Traditional multi-step diffusion models [3, 10, 34, 38] are known for slow sampling, often requiring dozens to hundreds of neural function evaluations (NFEs) per image. Recent diffusion distillation methods [6, 24, 29, 32, 39, 45] significantly accelerate generation by aligning the student model's prediction trajectory with that of a pre-trained multi-step teacher, enabling few-step (4-8 step) inference. Progressive Distillation [39] repeatedly distills from the teacher while halving the number of steps at each stage, preserving high sample quality while reducing from thousands of steps. Consistency Model [24, 28, 45] enforces self-consistency in predictions via either distillation-based or distillation-free objectives. ADD [42] and LADD [41] employs a combination of adversarial training and score distillation for turning pretrained multi-step diffusion models into few-step diffusion model. SANA-Sprint [4] accelerates sampling with a training-free transformation into TrigFlow [22], followed by few-step training with dense time embeddings, QK-normalization, and max-time weighting.

2.2. Image Inpainting Approaches

Image inpainting fills missing regions so they blend naturally with the surrounding context. Early methods [5, 23, 35] use GANs or unconditional diffusion models trained on specific datasets [7, 15, 21]. For instance, RePaint [23] uses an unconditional DDPM [10] as a generative prior and

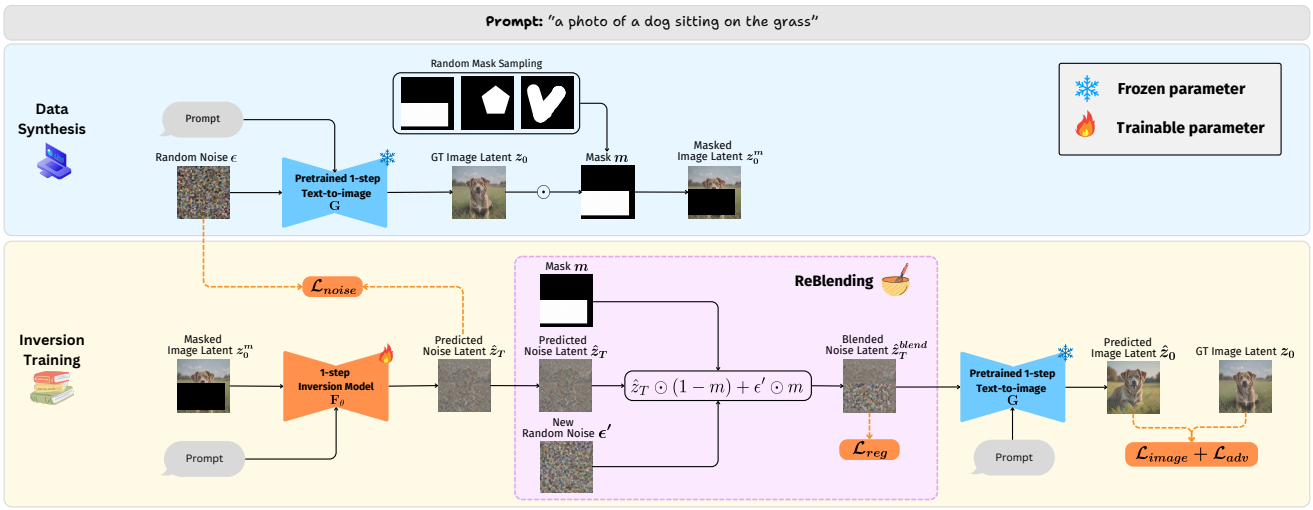


Figure 2. **Inversion Network Training:** We train an inversion network, $\mathbf{F}_\theta$, to invert a masked image to a noise latent $\hat{z}_T$ such that, after blending with random Gaussian noise to form $\hat{z}_T^{\text{blend}}$, the latent enables high-fidelity, well-harmonized reconstruction of the original image.

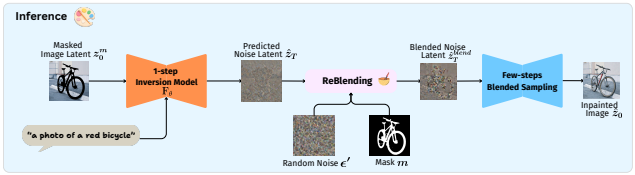


Figure 3. **Inpainting Pipeline:** The inversion network extracts the latent $\hat{z}_T$ from a masked image, which is blended with random noise to form $\hat{z}_T^{\text{blend}}$ and then fed into a few-step inpainting pipeline to generate the final image. *(Zoom in for details)*

blends available pixels into the sampling process. Text-to-image diffusion models provide strong image–text priors for text-guided inpainting, which demands both realistic content completion and semantic alignment with the prompt. BrushNet [13] fine-tunes both a pretrained text-to-image model and an additional conditional branch for inpainting, and then relies on multi-step sampling to produce coherent results. Meanwhile, Blended Latent Diffusion [1] guides the multi-step sampling process using a blending operation, gradually aligning the inpainting content with the surrounding background from the source image. Such methods require many NFEs to achieve high-quality results. As fast few-step generative models emerge [4, 24, 41], reducing the number of sampling steps becomes increasingly necessary, motivating the study of few-step inpainting. A straightforward idea is to apply similar blending strategies on few-step text-to-image models. However, extending inpainting to few-step diffusion models [4, 24, 41] remains challenging, as blended sampling alone is insufficient to produce coherent results, leading to poor visual quality, as shown in Fig. 1. TurboFill [49] addresses this by training an inpainting adapter on a few-step text-to-image generation model with a complex 3-step adversarial training scheme, which requires extensive real-image supervision. Moreover, TurboFill exclusively explores on UNet–based architectures [34, 38], leaving its generalization to other models questionable. Hence, few-step inpainting remains under-explored.

2.3. Diffusion-based Inversion

While diffusion models generate images by progressively denoising a noisy latent, diffusion inversion methods [9, 14, 27, 40, 44] perform the reverse: recovering an inverted latent that faithfully reconstructs the original image when re-denoised. Such inversion is essential for reconstruction, latent exploration, and downstream editing. DDIM Inversion [44] introduced a deterministic reverse process by linearizing noise prediction across adjacent steps, an approximation effective for models with many sampling iterations [8, 17, 34, 38]. This enables reversed sampling for faithful reconstruction and editing. Null-text Inversion [27] refines null-text embeddings via costly iterative optimization, whereas Direct Inversion [14] eliminates this optimization by decoupling reconstruction and editing pathways.

However, the linear approximation used in prior inversion methods breaks down for few-step diffusion models [24, 42, 45], resulting in poor inversion quality. Recent works [9, 40] therefore develop inversion techniques tailored to few-step models [17, 34]. Renoise [9] refines noise latents using fixed-point iteration combined with step-wise averaging, while GNRI [40] formulates inversion as a scalar root-finding problem solved with 1-2 Newton–Raphson iterations per step. These methods significantly accelerate and stabilize inversion compared to multi-step approaches. Recently, SwiftEdit [31] pushes this further with a one-step inversion network trained for one-step diffusion models [6, 20, 29, 53, 54]. This network directly maps source

images into its noise latent in a single forward pass, enabling fast image reconstruction and editing with minimal overhead. Inspired by this, we incorporate a similar inversion network into our inpainting framework, enhanced with refinements and dedicated training objectives to enable efficient, high-quality few-step inpainting.

3. Preliminaries

3.1. Text-to-Image Diffusion Models.

Text-to-image diffusion models synthesize images by aligning textual inputs with corresponding visual features. State-of-the-art methods primarily use latent diffusion [34, 38], where a Variational Auto-Encoder (VAE) [16] encoder $\mathcal{E}$ maps an image I to a latent z. The denoising process comprises a fixed forward noising step and a learned reverse step. In the forward process, a clean latent $z_0 = \mathcal{E}(I)$ is gradually corrupted into Gaussian noise over T timesteps via a Markov chain $q(z_t|z_{t-1})$ with a variance schedule β_t:

$$q(z_t|z_{t-1}) = \mathcal{N}(z_t; \sqrt{1-\beta_t}z_{t-1}, \beta_t\mathbf{I}). \quad (1)$$

$$z_t = \sqrt{1-\beta_t}z_{t-1} + \sqrt{\beta_t}\epsilon, \quad \text{where } \epsilon \sim \mathcal{N}(0, \mathbf{I}). \quad (2)$$

Given an input noise z_T sampled from Eq. (1) and a text prompt c, the training objective of the denoising network ϵ_θ at timestep t is defined as:

$$\min_\theta \mathbb{E}_{z_0,c,\epsilon\sim\mathcal{U}(1,T),t\sim\mathcal{N}(0,I)} \|\epsilon - \epsilon_\theta(z_t, t, c)\|_2^2 \quad (3)$$

During inference, ϵ_θ iteratively estimates and removes the noise from the noisy image across T timesteps. In practice, large T are required to gradually refine the image, ensuring high-quality generation. In contrast, few-step models apply large, discrete updates at each step, which limits the opportunity for smooth adjustments. Any intermediate modification, such as blending, can easily disrupt the denoising trajectory, leading to artifacts or failed reconstructions.

3.2. Image Inpainting

Problem Definition. Given a masked image $I_m \in \mathbb{R}^{H\times W\times C}$ with missing content defined by a binary mask $M \in \{0,1\}^{H\times W\times C}$, where 0 denotes unmasked regions and 1 denotes masked regions, image inpainting aims to generate $I_{inpaint}$ within the masked region to form a composited image $I = I_m \odot (1-M) + I_{inpaint} \odot M$ such that the inpainted regions are semantically aligned with a text prompt c and visually consistent with the unmasked context, accurately reflecting what and where to inpaint.

Blended Sampling Strategy. This inpainting approach, exemplified by Blended Latent Diffusion (BLD) [1], gradually blends known information from unmasked regions with generated content in the masked areas. Given a masked image I_m and a corresponding binary mask M (resized to m in the latent space), the initial masked latent representation is computed as $z_0^m = \mathcal{E}(I_m)$. During the reverse diffusion process, at each timestep t, BLD adds noise to the known regions of the original latent z_0^m following Eq. (1), yielding z_t^m. As described in Eq. (4), BLD then blends z_t^m with the predicted denoised latent $\hat{z}_t$ using the mask m. The resulting blended latent serves as the input for the subsequent denoising step at $t-1$, ensuring a seamless transition between the unmasked context and the newly generated content.

$$z_t^{blend} = z_t^m \odot (1-m) + \hat{z}_t \odot m \quad (4)$$



Figure 4. **Failure of BLD in few-step models** (SDXL-Turbo, 4 steps) is illustrated in *Column 3* and corrected by our method in *Column 4. (Zoom in for details)*

4. Method

In Sec. 4.1, we analyze the failure of blended sampling in few-step models and outline the motivations behind InverFill. The key component of our system is a one-step inversion network tailored for inpainting. We will present an overview on this network (Sec. 4.2), followed by our proposed components in training pipeline (Secs. 4.3 to 4.7). Finally, we present our inpainting pipeline in Sec. 4.8. Fig. 2 illustrates the training pipeline of our inversion network, while Fig. 3 shows the full inpainting inference pipeline.

4.1. Motivation

Blended Sampling Strategy for Few-Step Model. While BLD is effective for multi-step diffusion models, applying it directly to few-step models significantly reduces inpainting quality. As shown in Fig. 4, blending under few-step inference introduces semantic and stylistic inconsistencies between the generated and unmasked regions, yielding visible artifacts. This limitation originates from the initialized random Gaussian noise in the reverse process: multi-step models progressively refine this noise and adapt to the context of the unmasked regions within I_m, whereas few-step models make large ODE updates and lack sufficient refinement steps. When initialized from semantically distant noise, few coarse updates cannot correct the mismatch. Thus, effec-

tive few-step blending requires initializing z_T semantically aligned with the unmasked regions of the image.

Inversion for Image Inpainting. A promising direction for mitigating semantic misalignment is diffusion inversion, which maps the unmasked image into the final noise latent z_T. However, existing inversion methods are iterative and introduce considerable overhead, contradicting the efficiency requirements of few-step sampling. A one-step inversion is critical for fast inference, as demonstrated by SwiftEdit [31], which provides efficient and semantically coherent initialization. Nonetheless, directly applying SwiftEdit to inpainting is unsuitable for two reasons: (1) it is not explicitly designed for processing masked inputs, which causes information leakage during training, and (2) its training objectives do not enforce the inverted noise to follow the required Gaussian prior, resulting in distributional mismatch and degraded reconstructions.

To overcome these limitations, we introduce **InverFill**, a one-step inversion network designed for inpainting. InverFill (1) operates directly on masked images to produce semantically aligned initial noise latents, and (2) explicitly regularizes the inverted noise to match the Gaussian prior. Addressing both issues enables InverFill to achieve high-fidelity, coherent inpainting within the few-step regime.

4.2. Masked Image Inversion Network

Problem Definition. Given a pretrained one-step text-to-image generator $\mathbf{G}$, we aim to develop a one-step inversion network $\mathbf{F}_\theta$ that is tailored for the inpainting purpose. Specifically, given a ground-truth image I_{gt} and a masked image I_m produced from I_{gt} using a binary mask M, i.e., $I_m = I_{gt} \odot (1 - M)$, their image latents are $z_0 = \mathcal{E}(I_{gt})$ and $z_0^m = \mathcal{E}(I_m)$, where $\mathcal{E}$ is the VAE encoder. We train $\mathbf{F}_\theta$ to map z_0^m and text prompt c to an inverted noise latent. The network is optimized so that passing this latent through $\mathbf{G}$ produces a predicted image latent $\hat{z}_0$ resembling the original latent z_0. The predicted noise latent should yield a reconstruction where (1) the background **faithfully preserves** the masked input, and (2) the generated region **harmonizes** with the background while remaining consistent with the text prompt and the unmasked content of I_m.

Inversion Network Architecture. Following [31], $\mathbf{F}_\theta$ shares the architecture of the one-step generator $\mathbf{G}$ and inherits its pretrained weights during as initialization.

Masked Image Training. To adapt our inversion network to masked image inputs, we leverage the one-step generator $\mathbf{G}$ to synthesize training image–mask–prompt triplets on the fly. Given a text prompt c and random Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, $\mathbf{G}$ produces a ground-truth image latent $z_0 = \mathbf{G}(\epsilon, c)$ and its corresponding image $I_{gt} = \mathcal{D}(z_0)$, where $\mathcal{D}$ is the VAE decoder. To ensure robustness and prevent overfitting to specific masks, we randomly sample a mask M of diverse shapes and brush types and apply it to I_{gt} to generate the masked image I_m. The subsequent masked image latent $z_0^m = \mathcal{E}(I_m)$ serves as input to our one-step inversion network $\mathbf{F}_\theta$, which predicts the inverted noise latent $\hat{z}_T$. In the following sections, we introduce our objective functions and describe how we optimize and integrate $\hat{z}_T$ to achieve a high-quality reconstruction of z_0.

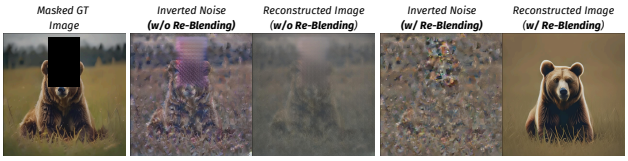


Figure 5. Effects of the proposed **Re-Blending operation** during training, without $\mathcal{L}_{\text{reg}}$. *(Zoom in for details)*

4.3. Reconstruction Objectives

Similar to SwiftEdit [31], we apply reconstruction losses in both the noise latent ($\mathcal{L}_{\text{noise}}$) and image latent ($\mathcal{L}_{\text{image}}$) spaces. Since our inversion network operates on a masked image I_m, applying $\mathcal{L}_{\text{noise}}$ over the entire predicted latent $\hat{z}_T$ is suboptimal: the regions of z_0^m corresponding to the masked areas of I_m contain no meaningful information, and penalizing these regions can hinder training. Therefore, we restrict $\mathcal{L}_{\text{noise}}$ to the unmasked regions. Our reconstruction objectives are formulated as follows:

$$\mathcal{L}_{\text{noise}} = \|(1 - m) \odot \hat{z}_T - (1 - m) \odot \epsilon\|_2^2, \tag{5}$$

$$\mathcal{L}_{\text{image}} = \|\hat{z}_0 - z_0\|_2^2 \tag{6}$$

$$\mathcal{L}_{\text{recons}} = \lambda_{\text{noise}} * \mathcal{L}_{\text{noise}} + \lambda_{\text{image}} * \mathcal{L}_{\text{image}} \tag{7}$$

4.4. Re-Blending Operation

Our inversion network maps the unmasked image content to a noise latent $\hat{z}_T$. In SwiftEdit [31], $\mathcal{L}_{\text{noise}}$ ensures that the predicted noise latent $\hat{z}_T$ preserves details of the complete input image I in the noise latent space. However, for inpainting tasks, our inversion network $\mathbf{F}_\theta$ only receives the incomplete masked image I_m to predict $\hat{z}_T$. Consequently, our masked loss $\mathcal{L}_{\text{noise}}$ in Eq. (5) causes training bias towards the unmasked regions. This bias causes image-space structural patterns from I_m to leak into $\hat{z}_T$, while regions corresponding to the mask exhibit low variance and artifacts. As a result, $\hat{z}_T$ deviates significantly from the Gaussian distribution expected by the diffusion model. During training, this distributional mismatch leads $\mathbf{G}$ to collapse when computing $\hat{z}_0 = \mathbf{G}(\hat{z}_T, c)$, producing the incoherent, artifact-filled outputs illustrated in Fig. 5.

To address this, we introduce a **Re-Blending** operation. During training and inference, the masked regions of the predicted noise latent $\hat{z}_T$ are replaced with random Gaussian noise $\epsilon' \sim \mathcal{N}(0, I)$, partially restoring the latent to the expected distribution and recovering key semantic features,

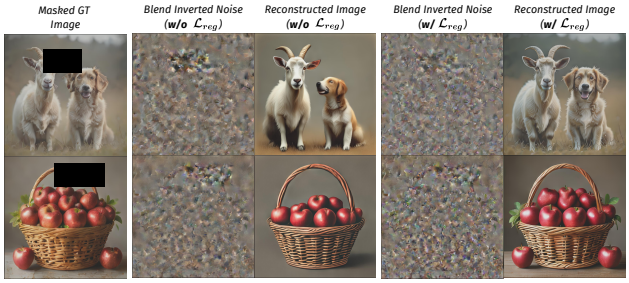


Figure 6. **Ablation Study on $\mathcal{L}_{\text{reg}}$.** Without the regularization loss, the model fails to preserve the background during reconstruction and produces blurred, low-detail outputs. With the loss, the background is well preserved and fine image details are restored.

as shown in Fig. 5. Following Eq. (8), generator $\mathbf{G}$ inputs the corrected latent $\hat{z}_T^{blend}$ to produce the final output $\hat{z}_0$.

$$\hat{z}_T^{blend} = \hat{z}_T \odot (1-m) + \epsilon' \odot m, \quad \hat{z}_0 = \mathbf{G}(\hat{z}_T^{blend}, c) \quad (8)$$

where m is latent-space mask downsampled from M.

4.5. Gaussian Regularization

As shown in Fig. 5, Re-Blending mitigates information leakage and partially restores the Gaussian structure of $\hat{z}_T^{\text{blend}}$, but the predicted latent $\hat{z}_0$ recovers only key semantics. The output fails to preserve background, and the generated content within masked regions remains blurry and low-detail, as shown in Fig. 6. Despite Re-Blending's partial correction, $\hat{z}_T^{\text{blend}}$ still deviates from the standard Gaussian expected by the generator. This occurs because $\mathcal{L}_{\text{noise}}$ in Eq. (5) focuses solely on the unmasked regions. While the image-space loss $\mathcal{L}_{\text{image}}$ in Eq. (6) indirectly encourages the injected noise ϵ' to harmonize with $\hat{z}_T$ to better reconstruct image latent, it cannot fully enforce Gaussian consistency due to the lack of direct supervision with ground-truth Gaussian noise within the masked regions. As a result, the added ϵ' remains visually distinct from the background inverted noise in $\hat{z}_T$ (Column 2 in Fig. 6), indicating that $\hat{z}_T^{\text{blend}}$ is still far from the target Gaussian distribution.

To address this issue, inspired by [12], we introduce an additional Gaussian regularization term. This term explicitly encourages Gaussian distribution on the blended latent $\hat{z}_T^{\text{blend}}$ by matching its statistical moments with the theoretical moments of a standard Gaussian. Let μ_n be the n-th theoretical moment of a standard Gaussian. The moment-matching loss for the n-th moment is defined as:

$$\mathcal{L}_n = \left\| \left| \frac{1}{D} \sum_{k=1}^{D} \left(\hat{z}_T^{blend}\right)^n \right|^{\frac{1}{n}} - \mu_n^{\frac{1}{n}} \right\|, \quad (9)$$

where $D = c \times h \times w$ is the total number of pixels of $\hat{z}_T^{blend}$. Our final regularization loss, $\mathcal{L}_{\text{reg}}$, is the sum of the losses for the first and second moments, corresponding to the mean and variance of Gaussian distribution:

$$\mathcal{L}_{\text{reg}} = \sum_{n \in \{1,2\}} \mathcal{L}_n \quad (10)$$

As shown in Fig. 6, Gaussian Regularization Loss during training helps $\hat{z}_T^{\text{blend}}$ better align with the Gaussian prior, enabling faithful reconstruction of the original image while preserving the background, as confirmed in Tab. 2.

4.6. Improving Quality with Adversarial Loss

Previous works [4, 53, 54] show that adversarial losses during training improve visual quality. Following LADD [41], we use the frozen teacher model to define a latent feature space for adversarial supervision, with multiple discriminator heads on intermediate layers for stable, efficient distillation. In our training, we treat the original image latent z_0 as real and the predicted image latent $\hat{z}_0$ as fake to train the inversion model and discriminator as follows:

$$\mathcal{L}_{\text{adv}}^{G}(\theta) = -\mathbb{E}_{\hat{z}_0,t}\left[\sum_k D_{\psi,k}\left(G_{\text{pre}}\left(\hat{z}_t, t, c\right)\right)\right] \quad (11)$$

$$\mathcal{L}_{\text{adv}}^{D}(\psi) = \mathbb{E}_{z_0,t}\left[\sum_k \text{ReLU}\left(1 - D_{\psi,k}\left(G_{\text{pre}}\left(z_t, t, c\right)\right)\right)\right] + \mathbb{E}_{\hat{z}_0,t}\left[\sum_k \text{ReLU}\left(1 + D_{\psi,k}\left(G_{\text{pre}}\left(\hat{z}_t, t, c\right)\right)\right)\right] \quad (12)$$

where z_t, $\hat{z}_t$ are noisy versions of original image latent z_0 and predicted image latent $\hat{z}_0$ at timesteps t. G_{pre} denotes a frozen multi-step teacher model. $D_{\psi,k}$ denotes discriminator heads at the k-th intermediate layers of G_{pre}.

4.7. Final Objectives

Our final training objective for $\mathbf{F}_\theta$ is defined as follows:

$$\mathcal{L}_{\text{final}} = \lambda_{\text{recons}} * \mathcal{L}_{\text{recons}} + \lambda_{\text{reg}} * \mathcal{L}_{\text{reg}} + \lambda_{\text{adv}} * \mathcal{L}_{\text{adv}} \quad (13)$$

4.8. Inpainting Pipeline

Fig. 3 illustrates our inpainting pipeline, which closely follows the blended sampling strategy described in Sec. 3.2. However, instead of initializing with random Gaussian noise, we employ our trained inversion model $\mathbf{F}_\theta$ to predict the inverted noise latent $\hat{z}_T$ and obtain the blended latent $\hat{z}_T^{blend}$ using Eq. (8). This blended latent $\hat{z}_T^{blend}$ serves as the Gaussian noise input to the inpainting process.

5. Experiments

5.1. Training Details

We train InverFill on Sana-Sprint 0.6B [4] and SDXL-Turbo [42], which represent two common diffusion architectures: DiT and UNet. All training is performed on four

Table 1. **Quantitative comparison of InverFill against few-step and multi-step diffusion inpainting baselines** on BrushBench and MagicBrush. NFEs denotes the number of function evaluations. ↑ indicates that higher is better, ↓ indicates that lower is better

Type	Method	NFEs	BrushBench				MagicBrush				Runtime↓ (seconds)
			IR$_{\times 10}$ ↑	HPS$_{\times 10^2}$ ↑	AS↑	CLIP↑	IR$_{\times 10}$ ↑	HPS$_{\times 10^2}$ ↑	AS↑	CLIP↑	
Few-step	SANA-Sprint 0.6B	2	11.02	26.21	6.05	27.12	2.55	25.07	5.32	25.67	0.37
	SANA-Sprint 0.6B + InverFill	2	**11.65**	**27.93**	**6.15**	**27.17**	**3.04**	**25.37**	**5.42**	**25.71**	0.43 (+0.06)
	SANA-Sprint 0.6B	4	10.82	26.34	6.00	27.11	2.56	25.12	5.37	25.63	0.45
	SANA-Sprint 0.6B + InverFill	4	**11.76**	**27.83**	**6.18**	**27.19**	**3.14**	**25.47**	**5.43**	**25.74**	0.51 (+0.06)
	SDXL Turbo	4	11.42	28.20	6.06	27.26	3.51	25.76	5.46	25.79	0.66
	SDXL Turbo + InverFill	4	**12.38**	**28.44**	**6.08**	**27.67**	**3.75**	**25.84**	**5.48**	**26.08**	0.70 (+0.04)
	SDXL Turbo + BrushNet	4	12.56	28.26	6.00	27.51	**4.20**	24.92	5.20	25.62	0.70
	SDXL Turbo + BrushNet + InverFill	4	**12.63**	**28.43**	**6.03**	**27.62**	4.15	**25.10**	**5.23**	**25.68**	0.74 (+0.04)
Multi-step	SANA 0.6B	20	12.12	27.04	6.17	27.49	3.68	24.11	5.48	25.93	1.18
	HD-Painter	30	12.82	28.17	6.30	27.43	3.59	24.60	5.65	25.87	23.45
	SDXL-Inpainting	30	13.16	28.92	6.37	27.15	3.91	24.13	5.51	25.50	3.35
	SDXL + BrushNet	30	13.26	28.28	6.26	27.54	3.94	24.28	5.46	25.60	4.31

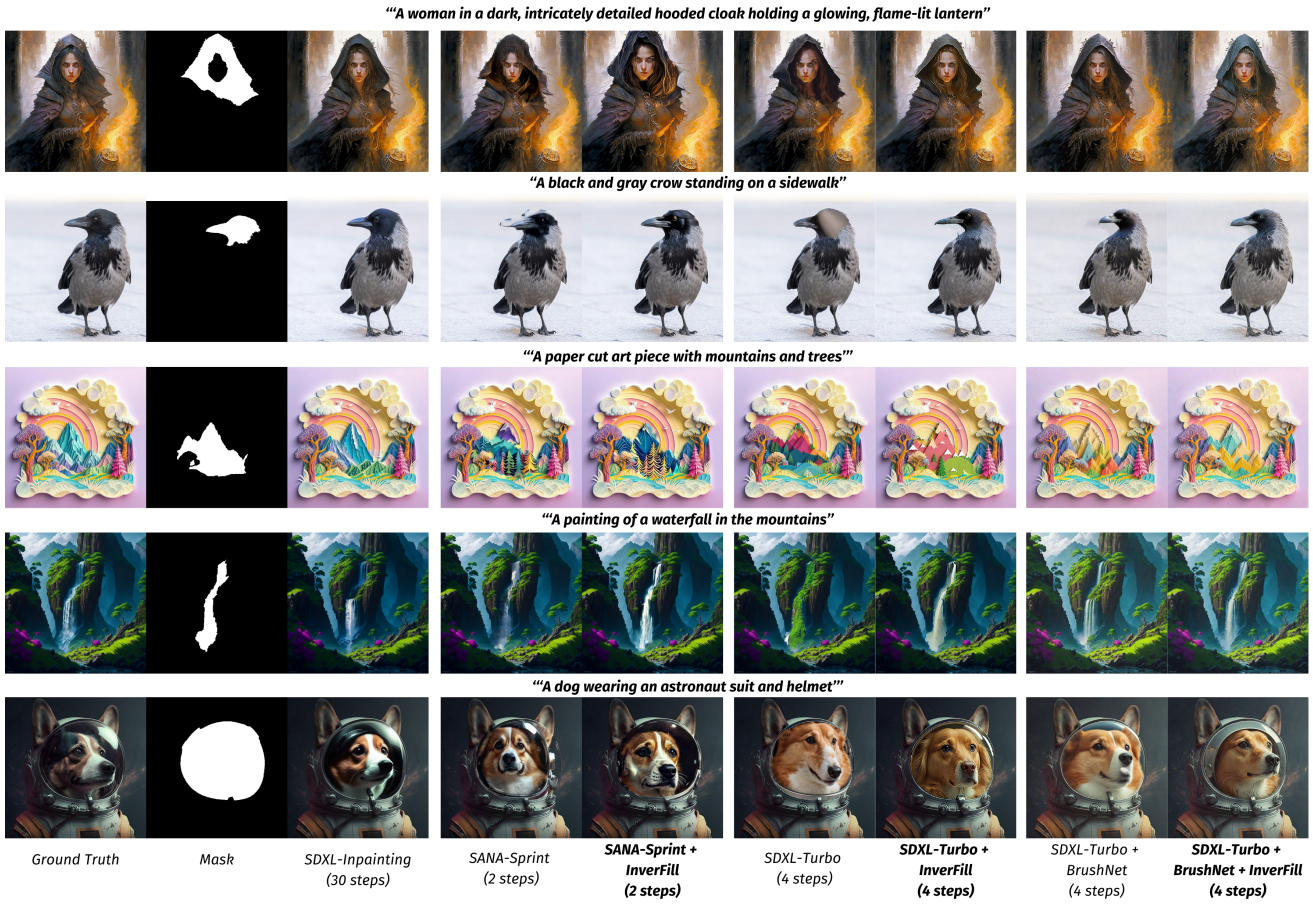


Figure 7. Our method achieves qualitative results comparable to multi-step SDXL-Inpainting and is on par with BrushNet (4 steps), as shown in Columns **7** and **8**. Notably, this performance is obtained using only text prompts during training, whereas competing methods rely on full text–image–mask supervision. Moreover, integrating our approach with BrushNet further enhances semantic coherence.

NVIDIA A100 40GB GPUs for 8-10 hours. During training, we randomly sample text prompts from BrushData [13] and MSCOCO [18]. We use a total batch size of 32 and a learning rate of 1×10^{-5} with AdamW optimizer.

Table 2. Effects of $\mathcal{L}_{\text{reg}}$ on SANA-Sprint 0.6B [4] with 2 NFEs on BrushBench [13]. All models were evaluated at 5000 iterations.

Method	IR$_{\times 10}$↑	HPS$_{\times 10^2}$↑	AS↑	CLIP↑
w/o $\mathcal{L}_{reg}$	11.11	26.69	6.08	27.13
w/ $\mathcal{L}_{reg}$	**11.40**	**27.22**	**6.12**	**27.15**

5.2. Evaluation Setup

Dataset. We perform evaluation on inpainting BrushBench [13], with 600 images and annotated masks, and image editing MagicBrush [55] benchmark. For inpainting, we adapt MagicBrush's 535-image test set using its captions and masks. Each image includes multiple segmentation and random masks, providing a diverse and challenging evaluation for inpainting performance. All experiments and evaluations were performed using 1024^2 resolution.

Evaluation Metrics. We evaluate our results from two criteria: image generation quality and text alignment.

- *Image Generation Quality.* We use three human-aligned metrics: ImageReward (IR) [51], HPS v2 (HPS) [47], and Aesthetic Score (AS) [43]. IR and HPS are reward models trained on large-scale human preference data, while AS is a linear model trained to predict perceptual quality.
- *Text Alignment.* We measure text–image alignment using CLIP Similarity (CLIP) [36], which quantifies how well the inpainted images match their prompts.

Baselines. We evaluate InverFill on state-of-the-art few-step text-to-image diffusion models, SANA-Sprint 0.6B [4] and SDXL-Turbo [42], using the blended sampling strategy in Sec. 3.2 for inpainting. We report results using 2- and 4-step NFE settings for SANA-Sprint and 4-step for SDXL-Turbo. Following [49], we integrate SDXL-Turbo [34] with BrushNet [13] to evaluate InverFill using few-step specialized inpainting models that do not rely on the blended sampling strategy. For reference, we report results from multi-step models, including Sana 0.6B [48], HD-Painter [25], SDXL-Inpainting [34] and SDXL with BrushNet [13].

5.3. Quantitative Results

As shown in Tab. 1, InverFill consistently improves performance across few-step diffusion settings. When integrated with SANA-Sprint and SDXL-Turbo under blended sampling, InverFill boosts all metrics on BrushBench and MagicBrush. For example, SANA-Sprint (2 NFEs) + InverFill raises IR from 11.02 to **11.65** on BrushBench and 2.55 to **3.04** on MagicBrush. InverFill also strengthens specialized inpainting model. In BrushNet + InverFill (4 NFEs), IR improves from 12.56 to **12.63** and HPS from 28.26 to **28.43**. Regarding text alignment, InverFill achieves substantial gains in CLIP scores. Notably, InverFill-equipped few-step models match or surpass multi-step methods while remaining efficient; SDXL-Turbo + InverFill (4 NFEs) outperforms HD-Painter (30 NFEs) on key metrics. Despite these gains, InverFill introduces extremely minimal overhead, only **0.06s** on SANA-Sprint and **0.04s** on SDXL.

Table 3. Quantitative comparison of InverFill against few-step and multi-step diffusion inpainting baselines on **BrushBench with enhanced prompts**. ↑ indicates that higher is better.

	Method	NFEs	IR$_{\times 10}$↑	HPS$_{\times 10^2}$↑	AS↑	CLIP↑
Few-step	SANA-Sprint 0.6B	2	7.53	25.81	6.09	28.10
	SANA-Sprint 0.6B + InverFill	2	**8.55**	**26.49**	**6.19**	**28.26**
	SANA-Sprint 0.6B	4	7.32	25.81	6.11	28.15
	SANA-Sprint 0.6B + InverFill	4	**8.54**	**26.50**	**6.21**	**28.29**
	SDXL Turbo	4	8.09	26.85	6.12	28.25
	SDXL Turbo + InverFill	4	**9.00**	**27.11**	**6.12**	**28.70**
	SDXL Turbo + BrushNet	4	9.35	27.22	6.06	28.87
	SDXL Turbo + BrushNet + InverFill	4	**9.53**	**27.37**	**6.10**	**28.91**
Multi-step	SANA 0.6B	20	9.34	26.77	6.24	28.63
	HD-Painter	30	9.60	27.83	6.35	28.63
	SDXL-Inpainting	30	9.90	27.43	6.38	28.38
	SDXL + BrushNet	30	10.41	28.11	6.34	28.76

5.4. Qualitative Results

Fig. 7 shows that integrating InverFill improves coherence and background harmonization. Without using real images, InverFill achieves quality comparable to BrushNet (4-step SDXL-Turbo), which relies on an inpainting dataset of real images [13]. Moreover, combining InverFill with the BrushNet + SDXL-Turbo pipeline further boosts semantic quality, indicating that InverFill can also strengthen specialized few-step inpainting systems.

5.5. Enhanced Caption for BrushBench

Motivation. A limitation of BrushBench [13] is its reliance on simple, short prompts, which limits evaluation of text understanding and compositional generation. Modern models, SDXL [34] with dual text encoders and SANA-Sprint [4] with Gemma-2 [37], are built for more context-heavy prompts. Therefore, we expand BrushBench captions using Qwen3 [52] with detailed foreground and background descriptions, enabling more comprehensive evaluation of text alignment and visual coherence in inpainting.

Quantitative Results. Tab. 3 shows that InverFill remains effective under detailed, complex prompts, improving all baselines and demonstrating robustness in text-rich settings. For SANA-Sprint, CLIP gains exceed those with simple prompts in Tab. 1, indicating stronger visual–text alignment and better use of large encoders like Gemma-2.

6. Conclusion

In this work, we introduce **InverFill**, a lightning-fast one-step inversion network explicitly designed for image inpainting that enhances existing few-step inpainting methods. Extensive experiments show that InverFill produces high-quality inpainting results while adding as few as **0.06 seconds** of overhead.

References

[1] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM transactions on graphics (TOG)*, 42 (4):1–11, 2023. 2, 3, 4

[2] Manuel Brack, Felix Friedrich, Katharia Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian Kersting, and Apolinário Passos. Ledits++: Limitless image editing using text-to-image models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8861–8870, 2024. 1

[3] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. 1, 2

[4] Junsong Chen, Shuchen Xue, Yuyang Zhao, Jincheng Yu, Sayak Paul, Junyu Chen, Han Cai, Enze Xie, and Song Han. Sana-sprint: One-step diffusion with continuous-time consistency distillation. *CoRR*, 2025. 2, 3, 6, 8

[5] Ciprian Corneanu, Raghudeep Gadde, and Aleix M Martinez. Latentpaint: Image inpainting in latent space with diffusion models. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 4334–4343, 2024. 2

[6] Trung Dao, Thuan Hoang Nguyen, Thanh Le, Duc Vu, Khoi Nguyen, Cuong Pham, and Anh Tran. Swiftbrush v2: Make your one-step diffusion model better than its teacher. In *European Conference on Computer Vision*, pages 176–192. Springer, 2024. 2, 3

[7] Trung Tuan Dao, Duc Hong Vu, Cuong Pham, and Anh Tran. Efhq: Multi-purpose extremepose-face-hq dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22605–22615, 2024. 2

[8] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 2021. 3

[9] Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. Renoise: Real image inversion through iterative noising. In *European Conference on Computer Vision*, pages 395–413. Springer, 2024. 3

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[11] Teng-Fang Hsiao, Bo-Kai Ruan, Sung-Lin Tsai, Yi-Lun Wu, and Hong-Han Shuai. Freecond: Free lunch in the input conditions of text-guided inpainting. *arXiv preprint arXiv:2412.00427*, 2024. 2

[12] Jisung Hwang, Jaihoon Kim, and Minhyuk Sung. Moment- and power-spectrum-based gaussianity regularization for text-to-image models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 6

[13] Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In *European Conference on Computer Vision*, pages 150–168. Springer, 2024. 2, 3, 7, 8

[14] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Pnp inversion: Boosting diffusion-based editing with 3 lines of code. In *The Twelfth International Conference on Learning Representations*, 2024. 3

[15] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2

[16] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014. 4

[17] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024. 1, 3

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 7

[19] Haipeng Liu, Yang Wang, and Meng Wang. One stone with two birds: A null-text-null frequency-aware diffusion models for text-guided image inpainting. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 2

[20] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *International Conference on Learning Representations*, 2024. 3

[21] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 2

[22] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[23] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022. 2

[24] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference, 2023. 2, 3

[25] Hayk Manukyan, Andranik Sargsyan, Barsegh Atanyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Hd-painter: high-resolution and prompt-faithful text-guided image inpainting with diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2023. 2, 8

[26] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2063–2072. IEEE, 2025. 2

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

the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 6038–6047, 2023. 2, 3

[28] Anh Nguyen, Viet Van Nguyen, Duc Vu, Trung Tuan Dao, Chi Tran, Toan Tran, and Anh Tuan Tran. Improved training technique for shortcut models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 2

[29] Thuan Hoang Nguyen and Anh Tran. Swiftbrush: One-step text-to-image diffusion model with variational score distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7807–7816, 2024. 2, 3

[30] Trong-Tung Nguyen, Duc-Anh Nguyen, Anh Tran, and Cuong Pham. Flexedit: Flexible and controllable diffusion-based object-centric image editing. *arXiv preprint arXiv:2403.18605*, 2024. 1

[31] Trong-Tung Nguyen, Quang Nguyen, Khoi Nguyen, Anh Tran, and Cuong Pham. Swiftedit: Lightning fast text-guided image editing via one-step diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21492–21501, 2025. 2, 3, 5

[32] Viet Nguyen, Anh Nguyen, Trung Dao, Khoi Nguyen, Cuong Pham, Toan Tran, and Anh Tran. Supercharged one-step text-to-image diffusion models with negative prompts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18004–18013, 2025. 2

[33] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 1

[34] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 2, 3, 4, 8

[35] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10619–10629, 2022. 2

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 8

[37] Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, et al. Gemma 2: Improving open language models at a practical size. *CoRR*, 2024. 8

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2, 3, 4

[39] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. 2

[40] Dvir Samuel, Barak Meiri, Haggai Maron, Yoad Tewel, Nir Darshan, Shai Avidan, Gal Chechik, and Rami Ben-Ari. Lightning-fast image inversion and editing for text-to-image diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025. 3

[41] Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. 2, 3, 6

[42] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXXVI*, page 87–103, Berlin, Heidelberg, 2024. Springer-Verlag. 2, 3, 6, 8

[43] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022. 8

[44] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 2, 3

[45] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*. JMLR.org, 2023. 2, 3

[46] Yufei Wang, Wenhan Yang, Xinyuan Chen, Yaohui Wang, Lanqing Guo, Lap-Pui Chau, Ziwei Liu, Yu Qiao, Alex C Kot, and Bihan Wen. Sinsr: diffusion-based image super-resolution in a single step. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 25796–25805, 2024. 1

[47] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *CoRR*, 2023. 8

[48] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024. 8

[49] Liangbin Xie, Daniil Pakhomov, Zhonghao Wang, Zongze Wu, Ziyan Chen, Yuqian Zhou, Haitian Zheng, Zhifei Zhang, Zhe Lin, Jiantao Zhou, et al. Turbofill: Adapting few-step text-to-image model for fast image inpainting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7613–7622, 2025. 2, 3, 8

[50] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22428–22437, 2023. 1

[51] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023. 8

[52] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 8

[53] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024. 2, 3, 6

[54] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024. 3, 6

[55] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023. 8