ReNoise: Real Image Inversion Through Iterative Noising

Daniel Garibi[1], Or Patashnik[1], Andrey Voynov[2], Hadar Averbuch-Elor[1], and Daniel Cohen-Or[1]

[1] Tel Aviv University
[2] Google Research

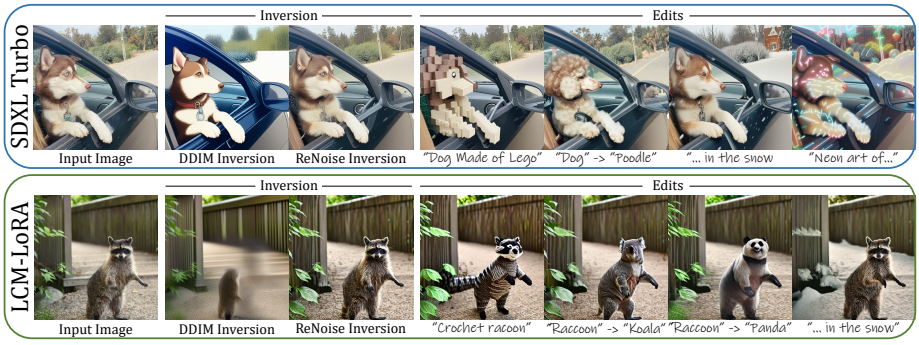

Fig. 1: Our ReNoise inversion technique can be applied to various diffusion models, including recent few-step ones. This figure illustrates the performance of our method with SDXL Turbo and LCM models, showing its effectiveness compared to DDIM inversion. Additionally, we demonstrate that the quality of our inversions allows prompt-driven editing. As illustrated on the right, our approach also allows for prompt-driven image edits.

Abstract. Recent advancements in text-guided diffusion models have unlocked powerful image manipulation capabilities. However, applying these methods to real images necessitates the inversion of the images into the domain of the pretrained diffusion model. Achieving faithful inversion remains a challenge, particularly for more recent models trained to generate images with a small number of denoising steps. In this work, we introduce an inversion method with a high quality-to-operation ratio, enhancing reconstruction accuracy without increasing the number of operations. Building on reversing the diffusion sampling process, our method employs an iterative renoising mechanism at each inversion sampling step. This mechanism refines the approximation of a predicted point along the forward diffusion trajectory, by iteratively applying the pretrained diffusion model, and averaging these predictions. We evaluate the performance of our ReNoise technique using various sampling algorithms and models, including recent accelerated diffusion models. Through comprehensive evaluations and comparisons, we show its effectiveness in terms of both accuracy and speed. Furthermore, we confirm that our method preserves editability by demonstrating text-driven image editing on real images.

Keywords: Diffusion Models · Image Editing · Inversion

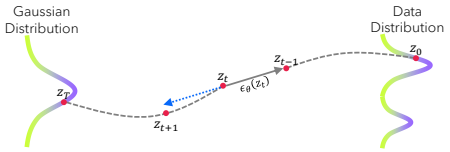

Fig. 2: The diffusion process samples a Gaussian noise and iteratively denoises it until reaching the data distribution. At each point along the denoising trajectory, the model predicts a direction, $\epsilon_\theta(z_t)$, to step to the next point along the trajectory. To invert a given image from the distribution, the direction from z_t to z_{t+1} is approximated with the inverse of the direction from z_t to z_{t-1} denoted by a dotted blue line.

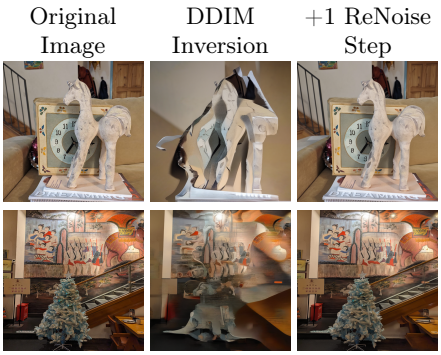

Fig. 3: Comparing reconstruction results of plain DDIM Inv. with SDXL to DDIM Inv. with one renoising iteration.

1 Introduction

Large-scale text-to-image diffusion models have revolutionized the field of image synthesis [19,37,39,40]. In particular, many works have shown that these models can be employed for various types of image manipulation [4,7–9,15,17,23,29,33, 35,47]. To edit *real* images, many of these techniques often require the inversion of the image into the domain of the diffusion model. That is, given a real image z_0, one has to find a Gaussian noise z_T, such that denoising z_T with the pretrained diffusion model reconstructs the given real image z_0. The importance of this task for real image manipulation has prompted many efforts aimed at achieving accurate reconstruction [16,21,30,31].

The diffusion process consists of a series of denoising steps $\{\epsilon_\theta(z_t, t)\}_{t=T}^{1}$, which form a trajectory from the Gaussian noise to the model distribution (see Figure 2). Each denoising step is computed by a trained network, typically implemented as a UNet, which predicts z_{t-1} from z_t [19]. The output of the model at each step forms a *direction* from z_t to z_{t-1} [45]. These steps are not invertible, in the sense that the model was not trained to predict z_t from z_{t-1}. Thus, the problem of inverting a given image is a challenge, and particularly for real images, as they are not necessarily in the model distribution (see Figure 3).

In this paper, we present an inversion method with a high quality-to-operation ratio, which achieves superior reconstruction accuracy for the same number of UNet operations. We build upon the commonly used approach of reversing the diffusion sampling process, which is based on the linearity assumption that the direction from z_t to z_{t+1} can be approximated by the negation of the direction from z_t to z_{t-1} [12,43] (see Figure 2). To enhance this approximation, we employ the fixed-point iteration methodology [10]. Specifically, given z_t, we begin by using the common approximation to get an initial estimate for z_{t+1}, denoted by $z_{t+1}^{(0)}$. Then, we iteratively *renoise* z_t, following the direction implied by $z_{t+1}^{(k)}$ to obtain $z_{t+1}^{(k+1)}$. After repeating this renoising process several times, we apply an averaging on $z_{t+1}^{(k)}$ to form a more accurate direction from z_t to z_{t+1}.

We show that this approach enables longer strides along the inversion trajectory while improving image reconstruction. Therefore, our method can also be effective with diffusion models trained to generate images using a small number of denoising steps [27, 42]. Furthermore, despite the need to repeatedly renoise in each step of the inversion process, the longer strides lead to a more favorable tradeoff of UNet operations for reconstruction quality.

Through extensive experiments, we demonstrate the effectiveness of our method in both image reconstruction and inversion speed. We validate the versatility of our approach across different samplers and models, including recent time-distilled diffusion models (e.g., SDXL-Turbo [42]). Importantly, we demonstrate that the editability of the inversion achieved by our method allows a wide range of text-driven image manipulations (see Figure 1).

2 Related Work

Image Editing via Diffusion Models Recent advancements in diffusion models [12, 19] have resulted in unprecedented diversity and fidelity in visual content creation guided by free-form text prompts [36, 37, 39, 40]. Text-to-image models do not directly support text-guided image editing. Therefore, harnessing the power of these models for image editing is a significant research area and many methods have utilized these models for different types of image editing [4,7–9,11,14,15,17,18,23,29,33,35,47,48]. A common approach among these methods requires *inversion* [21, 31, 43, 49] to edit real images, i.e., obtaining a latent code z_T such that denoising it with the pretrained diffusion model returns the original image. Specifically, in this approach two backward processes are done simultaneously using z_T. One of the processes reconstructs the image using the original prompt, while the second one injects features from the first process (e.g., attention maps) to preserve some properties of the original image while manipulating other aspects of it.

Inversion in Diffusion Models Initial efforts in image inversion for real image editing focused on GANs [2,3,5,6,13,34,38,46,52–54]. The advancements in diffusion models, and in diffusion-based image editing in particular have recently prompted works studying the inversion of a diffusion-based denoising process. This inversion depends on the sampler algorithm used during inference, which can be deterministic [43] or non-deterministic [19, 22]. Inversion methods can be accordingly categorized into two: methods that are suitable for deterministic sampling, and methods suitable for non-deterministic sampling.

Methods that approach the deterministic inversion commonly rely on the DDIM sampling method [43], and build upon DDIM inversion [12, 43]. Mokady et al. [31] observed that the use of classifier-free guidance during inference magnifies the accumulated error of DDIM inversion and therefore leads to poor reconstruction. Following this observation, several works [16, 30, 31] focused on solving this issue by replacing the null text token with a different embedding, which is found after an optimization process or by a closed solution. However, excluding [31] which requires a lengthy optimization, these methods are limited by the reconstruction accuracy of DDIM inversion, which can be poor, especially when a small number of denoising steps is done. In our work, we present a method

that improves the reconstruction quality of DDIM inversion and therefore can be integrated with methods that build on it.

Another line of work [21, 50] tackles the inversion of DDPM sampler [19]. In these works [21,50], instead of inverting the image into an initial noise z_T, a series of noises $\{z_T, \epsilon_T, ..., \epsilon_1\}$ is obtained. The definition of this noises series ensures that generating an image with it returns the original input image. However, these methods require a large number of inversion and denoising steps to allow image editing. Applying these methods with an insufficient number of steps leads to too much information encoded in $\{\epsilon_T, ..., \epsilon_1\}$ which limits the ability to edit the generated image. As shall be shown, The editability issue of these methods is particularly evident in few-steps models [26, 27, 42].

Most relevant to our work, two recent inversion methods [28, 32] also use the fixed-point iteration technique. Specifically, they improve the reconstruction accuracy of DDIM inversion [43] with Stable Diffusion [39] without introducing a significant computational overhead. In our work, we focus on the problem of real image inversion for recently introduced few-step diffusion models, where the difficulties encountered by previous methods are amplified. Furthermore, we show that our inversion method successfully works with various models and different samplers.

Few Steps Models Recently, new methods [26, 27, 41, 42, 44] that fine-tune text-to-image diffusion models enabled a significant reduction of the number of steps needed for high-quality image generation. While standard diffusion models typically require 50 denoising steps to generate high-quality images, recent accelerated models achieve high-quality synthesis with 1-4 steps only. These new methods pave the way for interactive editing workflows. However, as we show in this paper, using current methods for the inversion of an image with a small number of steps degrades the reconstruction quality in terms of accuracy [12, 43] or editability [21, 50].

3 Method

3.1 ReNoise Inversion

Reversing the Sampler Samplers play a critical role in the diffusion-based image synthesis process. They define the noising and denoising diffusion processes and influence the processes' trajectories and quality of the generated images. While different samplers share the same pre-trained UNet model (denoted by ϵ_θ) as their backbone, their sampling approaches diverge, leading to nuanced differences in output. The goal of the denoising sampler is to predict the latent code at the previous noise level, z_{t-1}, based on the current noisy data z_t, the pretrained UNet model, and a sampled noise, ϵ_t. Various first-order denoising sampling algorithms adhere to the form:

$$z_{t-1} = \phi_t z_t + \psi_t \epsilon_\theta(z_t, t, c) + \rho_t \epsilon_t, \tag{1}$$

where c represents a text embedding condition, and ϕ_t, ψ_t, and ρ_t denote sampler parameters. At each step, these parameters control the extent to which the

previous noise is removed(ϕ_t), the significance assigned to the predicted noise from the UNet(ψ_t), and the given weight to the additional noise introduced(ρ_t).

A given image z_0 can be inverted by reformulating Equation 1 and applying it iteratively:

$$z_t = \frac{z_{t-1} - \psi_t \epsilon_\theta(z_t, t, c) - \rho_t \epsilon_t}{\phi_t}, \tag{2}$$

where for non-deterministic samplers, a series of random noises $\{\epsilon_t\}_{t=1}^{T}$ is sampled and used during both inversion and image generation processes. However, directly computing z_t from Equation 2 is infeasible since it relies on $\epsilon_\theta(z_t, t, c)$, which, in turn, depends on z_t, creating a circular dependency. To solve this implicit function, Dhariwal et al. [12] propose using the approximation $\epsilon_\theta(z_t, t, c) \approx \epsilon_\theta(z_{t-1}, t, c)$:

$$z_t^{(1)} = \frac{z_{t-1} - \psi_t \epsilon_\theta(z_{t-1}, t, c) - \rho_t \epsilon_t}{\phi_t}. \tag{3}$$

This method has several limitations. First, the assumption underlying the approximation used in [12] is that the number of inversion steps is large enough, implying a trajectory close to linear. This assumption restricts the applicability of this inversion method in interactive image editing with recent few-step diffusion models [26, 27, 42, 44], as the inversion process would take significantly longer than inference. Second, this method struggles to produce accurate reconstructions in certain cases, such as highly detailed images or images with large smooth regions, see Figure 3. Moreover, we observe that this inversion method is sensitive to the prompt c and may yield poor results for certain prompts.

ReNoise In a successful inversion trajectory, the direction from z_{t-1} to z_t aligns with the direction from z_t to z_{t-1} in the denoising trajectory. To achieve this, we aim to improve the approximation of $\epsilon_\theta(z_t, t, c)$ in Eq. 2 compared to the one used in [12]. Building on the fixed-point iteration technique [10], our approach better estimates the instance of z_t that is inputted to the UNet, rather than relying on z_{t-1}.

Intuitively, we utilize the observation that $z_t^{(1)}$ (from Eq. 3) offers a more precise estimate of z_t compared to z_{t-1}. Therefore, employing $z_t^{(1)}$ as the input to the UNet is likely to yield a more accurate direction, thus contributing to reducing the overall error in the inversion step. We illustrate this observation in Figure 5. Iterating this process generates a series of estimations for z_t, denoted by $\{z_t^{(k)}\}_{k=1}^{\mathcal{K}+1}$. While the fixed-point iteration technique [10] does not guarantee convergence of this series in the general case, in Section 4, we empirically show that convergence holds in our setting. However, as the convergence is not monotonic, we refine our prediction of z_t by averaging several $\{z_t^{(k)}\}$, thus considering more than a single estimation of z_t. See Figure 6 for an intuitive illustration.

In more detail, our method iteratively computes estimations of z_t during each inversion step t by renoising the noisy latent z_{t-1} multiple times, each with a different noise prediction (see Figure 4). Beginning with $z_t^{(1)}$, in the k-th renoising iteration, the input to the UNet is the result of the previous iteration, $z_t^{(k)}$. Then, $z_t^{(k+1)}$ is calculated using the inverted sampler while maintaining z_{t-1} as the starting point of the step. After $\mathcal{K}$ renoising iterations, we obtain a set of

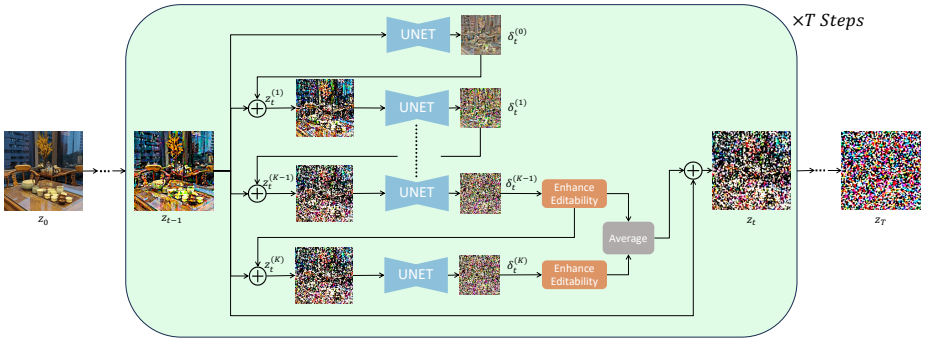

Fig. 4: Method overview. Given an image z_0, we iteratively compute $z_1, ..., z_T$, where each z_t is calculated from z_{t-1}. At each time step, we apply the UNet (ϵ_θ) $\mathcal{K}+1$ times, each using a better approximation of z_t as the input. The initial approximation is z_{t-1}. The next one, $z_t^{(1)}$, is the result of the reversed sampler step (i.e., DDIM). The reversed step begins at z_{t-1} and follows the direction of $\epsilon_\theta(z_{t-1}, t)$. At the k renoising iteration, $z_t^{(k)}$ is the input to the UNet, and we obtain a better z_t approximation. For the lasts iterations, we optimize $\epsilon_\theta(z_t^{(k)}, t)$ to increase editability. As the final denoising direction, we use the average of the UNet predictions of the last few iterations.

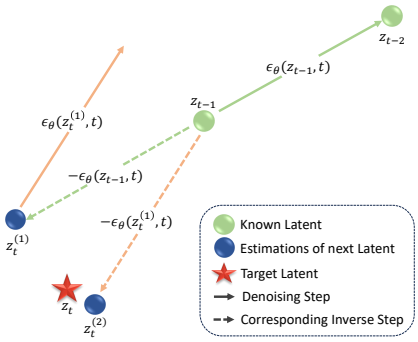

Fig. 5: Geometric intuition for ReNoise. At each inversion step, we estimate z_t (marked with a red star) based on z_{t-1}. The straightforward approach is to use the negated direction of the denoising step from z_{t-1}, assuming the trajectory is approximately linear. However, this assumption is inaccurate, especially in few-step models, where the size of the steps is large. We use the linearity assumption as an initial estimation and keep improving this estimation. We recalculate the denoising step from the previous estimation (which is closer to the target z_t) and then proceed with its negated direction from z_{t-1} (see the orange vectors).

estimations $\{z_t^{(k)}\}_{k=1}^{\mathcal{K}+1}$. The next point on the inversion trajectory, z_t, is then defined as their weighted average, where w_k is the weight assigned to $z_t^{(k)}$. For a detailed description of our method, refer to Algorithm 1.

3.2 Reconstruction-Editability Tradeoff

Enhance Editability The goal of inversion is to facilitate the editing of real images using a pretrained image generation model. While the the renoising approach attains highly accurate reconstruction results, we observe that the resulting z_T lacks editability. This phenomenon can be attributed to the reconstruction-editability tradeoff in image generative models [46]. To address this limitation, we incorporate a technique to enhance the editability of our method.

It has been shown [33] that the noise maps predicted during the inversion process often diverge from the statistical properties of uncorrelated Gaussian white noise, thereby affecting editability. To tackle this challenge, we follow pix2pix-

zero [33] and regularize the predicted noise at each step, $\epsilon_\theta(z_t, t, c)$, using the following loss terms.

First, we encourage $\epsilon_\theta(z_t, t, c)$ to follow the same distribution as $\epsilon_\theta(z'_t, t, c)$, where z'_t represents the input image z_0 with added random noise corresponding to the noise level at timestep t. We do so by dividing $\epsilon_\theta(z_t, t, c)$ and $\epsilon_\theta(z'_t, t, c)$ into small patches (e.g., 4×4), and computing the KL-divergence between corresponding patches. We denote this loss term by $\mathcal{L}_{\text{patch-KL}}$. Second, we utilize $\mathcal{L}_{\text{pair}}$ proposed in pix2pix-zero [33], which penalizes correlations between pairs of pixels. We leverage these losses to enhance the editability of our method, and denote the combination of them as $\mathcal{L}_{\text{edit}}$. For any renoising iteration k where $w_k > 0$, we regularize the UNet's prediction $\epsilon_\theta(z_t^{(k)}, t, c)$ using $\mathcal{L}_{\text{edit}}$ before computing $z_t^{(k+1)}$. See line 9 in Algorithm 1.

Algorithm 1: ReNoise Inversion

1 **Input:** An image z_0, number of renoising steps $\mathcal{K}$, number of inversion steps T, a series of renoising weights $\{w_k\}_{k=1}^{\mathcal{K}+1}$.
2 **Output:** A noisy latent z_T and set of noises $\{\epsilon_t\}_{t=1}^{T}$.
3 **for** $t = 0, 1, \ldots, T$ **do**
4 sample $\epsilon_t \sim \mathcal{N}(0, I)$
5 $z_t^{(0)} \leftarrow z_{t-1}$
6 $z_t^{(\text{avg})} \leftarrow 0$
7 **for** $k = 0, \ldots, \mathcal{K}$ **do**
8 $\delta_t^k \leftarrow \epsilon_\theta(z_t^{(k)}, t)$
9 $\delta_t^k \leftarrow$ `Enhance-edit`(δ_t^k, w_{k+1})
10 $z_t^{(k+1)} \leftarrow$ `Inverse-Step`(z_{t-1}, δ_t^k)
11 **end**
`// Average ReNoised predictions`
12 $z_t^{(\text{avg})} \leftarrow \sum_{k=1}^{\mathcal{K}+1} w_k \cdot z_t^{(k)}$
13 $\epsilon_t \leftarrow$ `Noise-Corr`$(z_t^{(\text{avg})}, t, \epsilon_t, z_{t-1})$
14 **end**
15 **return** $(z_T, \{\epsilon_t\}_{t=1}^{T})$
16 **Function** `Inverse-Step`$(z_{t-1},\ \delta_t,\ t)$**:**
17 **return** $\frac{1}{\phi_t} z_{t-1} - \frac{\psi_t}{\phi_t}\delta_t - \frac{\rho_t}{\phi_t}\epsilon_t$
`// Enhance-editability`
18 **Function** `Enhance-edit`$(\delta_t^k,\ w_{k+1})$**:**
19 **if** $w_{k+1} > 0$ **then**
20 $\delta_t^k \leftarrow \delta_t^k - \nabla_{\delta_t^k} \mathcal{L}_{\text{edit}}(\delta_t^k)$
21 **end**
22 **return** δ_t^k
`// Noise Correction`
23 **Function** `Noise-Corr`$(z_t,\ t,\ \epsilon_t,\ z_{t-1})$**:**
24 $\delta_t \leftarrow \epsilon_\theta(z_t, t)$
25 $\epsilon_t \leftarrow \epsilon_t - \nabla_{\epsilon_t} \frac{1}{\rho_t}(z_{t-1} - \phi_t z_t - \psi_t \delta_t)$
26 **return** ϵ_t
27

Noise Correction in Non-deterministic Samplers Non-deterministic samplers, in which $\rho_t > 0$, introduce noise (ϵ_t) at each denoising step. Previous methods [21, 50] suggested using ϵ_t to bridge the gap between the inversion and denoising trajectories in DDPM inversion. Specifically, given a pair of points z_{t-1}, z_t on the inversion trajectory, we denote by $\hat{z}_{t-1}$ the point obtained by denoising z_t. Ideally, z_{t-1} and $\hat{z}_{t-1}$ should be identical. We define:

$$\epsilon_t = \frac{1}{\rho_t}(z_{t-1} - \phi_t z_t - \psi_t \epsilon_\theta(z_t, t, c)). \tag{4}$$

Integrating this definition into Eq. 1 yields $\hat{z}_{t-1} = z_{t-1}$. However, we found that replacing ϵ_t with the above definition affects editability. Instead, we suggest a more tender approach, optimizing ϵ_t based on Eq. 4 as our guiding objective:

$$\epsilon_t = \epsilon_t - \nabla_{\epsilon_t} \frac{1}{\rho_t}(z_{t-1} - \phi_t z_t - \psi_t \epsilon_\theta(z_t, t, c)). \tag{5}$$

This optimization improves the reconstruction fidelity while preserving the distribution of the noisy-latents.

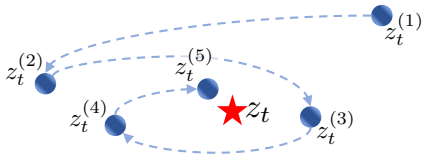

Fig. 6: Schematic illustration of the ReNoise convergence process to the true inversion of z_{t-1}. While estimates may converge non-monotonically to the unknown target z_t, we found that averaging them improves true value estimation. Typically, the initial iteration exhibits an exponential decrease in the norm between consecutive elements.

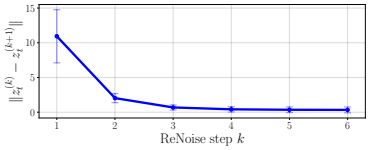

Fig. 7: Average distance between consequent estimations $z_t^{(k)}$, and $z_t^{(k+1)}$. Vertical bars indicate the standard deviation. The averages are computed over 32 images and 10 different timesteps.

4 Convergence Discussion

In this section, we first express the inversion process as a backward Euler process and our renoising iterations as fixed-point iterations. While these iterations do not converge in the general case, in the supplementary materials we present a toy example where they yield accurate inversions. Then, we analyze the convergence of the renoising iterations in our real-image inversion scenario and empirically verify our method's convergence.

Inversion Process as Backward Euler The denoising process of diffusion models can be mathematically described as solving an ordinary differential equation (ODE). A common method for solving such equations is the Euler method, which takes small steps to approximate the solution. For ODE in the form of $y'(t) = f(t, y(t))$, Euler solution is defined as:

$$y_{n+1} = y_n + h \cdot f(t_n, y_n),$$

where h is the step size. The inversion process can be described as solving ODE using the backward Euler method (or implicit Euler method) [1]. This method is similar to forward Euler, with the difference that y_{n+1} appears on both sides of the equation:

$$y_{n+1} = y_n + h \cdot f(t_{n+1}, y_{n+1}).$$

For equations lacking an algebraic solution, several techniques estimate y_{n+1} iteratively. As we described in Section 3.1, the inversion process lacks a closed-form solution, as shown in Equation 2. To address this, the ReNoise method leverages fixed-point iterations, which we refer to as *reonising iterations,* to progressively refine the estimate of y_{n+1}:

$$y_{n+1}^{(0)} = y_n, \quad y_{n+1}^{(k+1)} = y_n + h \cdot f(t_{n+1}, y_{n+1}^{(k)}).$$

In our ReNoise method, we average these renoising iterations to mitigate convergence errors, leading to improvement in the reconstruction quality.

ReNoise Convergence During the inversion process, we aim to find the next noise level inversion, denoted by $\hat{z}_t$, such that applying the denoising step to $\hat{z}_t$ recovers the previous state, z_{t-1}. Given the noise estimation $\epsilon_\theta(z_t, t)$ and a given z_{t-1}, the ReNoise mapping defined in Section 3.1 can be written as $\mathcal{G} : z_t \rightarrow \text{InverseStep}(z_{t-1}, \epsilon_\theta(z_t, t))$. For example, in the case of using DDIM sampler the mapping is $\mathcal{G}(z_t) = \frac{1}{\phi_t}(z_{t-1} - \psi_t \epsilon_\theta(z_t, t))$. The point $\hat{z}_t$, which is

mapped to z_{t-1} after the denoising step, is a stationary point of this mapping. Given $z_t^{(1)}$, the first approximation of the next noise level z_t, our goal is to show that the sequence $z_t^{(k)} = \mathcal{G}^{k-1}(z_t^{(1)}), k \to \infty$ converges. As the mapping $\mathcal{G}$ is continuous, the limit point would be its stationary point. The definition of $\mathcal{G}$ gives:

$$\|z_t^{(k+1)} - z_t^{(k)}\| = \|\mathcal{G}(z_t^{(k)}) - \mathcal{G}(z_t^{(k-1)})\|,$$

where the norm is always assumed as the l_2-norm. For the ease of the notations, we define $\Delta^{(k)} = z_t^{(k)} - z_t^{(k-1)}$. For convergence proof, it is sufficient to show that the sum of norms of these differences converges, which will imply that $z_t^{(k)}$ is the Cauchy sequence. Below we check that in practice $\|\Delta^{(k)}\|$ decreases exponentially as $k \to \infty$ and thus has finite sum. In the assumption that $\mathcal{G}$ is $\mathcal{C}^2$-smooth, the Taylor series conducts:

$$\begin{aligned}
\|\Delta^{(k+1)}\| = \|\mathcal{G}(z_t^{(k)}) - \mathcal{G}(z_t^{(k-1)})\| = \\
\|\mathcal{G}(z_t^{(k-1)}) + \frac{\partial \mathcal{G}}{\partial z}|_{z_t^{(k-1)}} \cdot \Delta^{(k)} + O(\|\Delta^{(k)}\|^2) - \mathcal{G}(z_t^{(k-1)})\| = \\
\|\frac{\partial \mathcal{G}}{\partial z}|_{z_t^{(k-1)}} \cdot \Delta^{(k)} + O(\|\Delta^{(k)}\|^2)\| \leq \|\frac{\partial \mathcal{G}}{\partial z}|_{z_t^{(k-1)}}\| \cdot \|\Delta^{(k)}\| + O(\|\Delta^{(k)}\|^2) = \\
\frac{\psi_t}{\phi_t} \cdot \|\frac{\partial \epsilon_\theta}{\partial z}|_{z_t^{(k-1)}}\| \cdot \|\Delta^{(k)}\| + O(\|\Delta^{(k)}\|^2)
\end{aligned}$$

Thus, in a sufficiently small neighborhood, the convergence dynamics is defined by the scaled Jacobian norm $\frac{\psi_t}{\phi_t} \cdot \|\frac{\partial \epsilon_\theta}{\partial z}|_{z_t^{(k-1)}}\|$. In the supplementary materials, we show this scaled norm estimation for the SDXL diffusion model for various steps and ReNoise iterations indices (k). Remarkably, the ReNoise indices minimally impact the scale factor, consistently remaining below 1. This confirms in practice the convergence of the proposed algorithm. Notably, the highest scaled norm values occur at smaller t (excluding the first step) and during the initial renoising iteration. This validates the strategy of not applying ReNoise in early steps, where convergence tends to be slower compared to other noise levels. Additionally, the scaled norm value for the initial t approaches 0, which induces almost immediate convergence.

Figure 7 illustrates the exponential decrease in distances between consecutive elements $z_t^{(k)}$ and $z_t^{(k+1)}$, which confirms the algorithm's convergence towards the stationary point of the operator $\mathcal{G}$. The proposed averaging strategy is aligned with the conclusions described above, and also converges to the desired stationary point. In The supplementary materials, we present a validation for this claim.

5 Experiments

In this section, we conduct extensive experiments to validate the effectiveness of our method. We evaluate both the reconstruction quality of our inversion and its editability. To demonstrate the versatility of our approach, we apply it to four models, SD [39], SDXL [36], SDXL Turbo [42], and LCM-LoRA [27], with SDXL Turbo and LCM-LoRA being few-step models. Additionally, we use various sampling algorithms including both deterministic and non-deterministic ones. Implementation details for each model are provided in the supplementary materials. Following previous works [5, 31], we quantitatively evaluate our

Table 1: Image reconstruction results with a fixed number of 100 UNet operations. Each row showcases the results obtained using different combinations of inversion steps, denoising steps, and renoising iterations, totaling 100 operations. As observed, allocating some of the operations to renoising iterations improves the reconstruction quality while maintaining the same execution time.

Image Reconstruction Results					
Inv Steps	Inf Steps	ReNoise Steps	L2 ↓	PSNR ↑	LPIPS ↓
50	50	0	0.00364	26.023	0.06273
75	25	0	0.00382	25.466	0.06605
80	20	0	0.00408	25.045	0.07099
90	10	0	0.01023	20.249	0.10305
25	25	2	<u>0.00182</u>	<u>29.569</u>	<u>0.03637</u>
20	20	3	**0.00167**	**29.884**	**0.03633**
10	10	8	0.00230	28.156	0.04678

Table 2: Quantitative ablation study on SDXL Turbo. We demonstrate the impact of each component of our inversion method on reconstruction results. The results improve with additional renoising iterations and significant enhancements occur through averaging final estimations. Additionally, we observe a reconstruction-editability trade-off, with edit losses causing degradation that is effectively mitigated by Noise Correction.

Ablation - Image Reconstruction			
	L2 ↓	PSNR ↑	LPIPS↓
Euler Inversion	0.0700	11.784	0.20337
+ 1 ReNoise	0.0552	12.796	0.20254
+ 4 ReNoise	0.0249	16.521	0.14821
+ 9 ReNoise	<u>0.0126</u>	19.702	0.10850
+ Averaging ReNoise	**0.0087**	<u>21.491</u>	<u>0.08832</u>
+ Edit Losses	0.0276	18.432	0.12616
+ Noise Correction	0.0196	**22.077**	**0.08469**

method with three metrics: L_2, LPIPS [51], and PSNR. Unless stated otherwise, for both inversion and generation we use the prompt obtained from BLIP2 [24].

5.1 Reconstruction and Speed

We begin by evaluating the reconstruction-speed tradeoff. The main computational cost of both the inversion and denoising processes is the forward pass through the UNet. In each renoising iteration, we perform one forward pass, which makes it computationally equal to a standard inversion step (as done in DDIM Inversion for example). In the following experiments, we compare the results of a sampler reversing with our method, where we match the number of UNet passes between the methods. For example, 8 steps of sampler reversing are compared against 4 steps with one renoising iteration at each step.

Qualitative Results In Figure 8 we show qualitative results of image reconstruction on SDXL Turbo [42]. Here, we utilize DDIM as the sampler, and apply four denoising steps for all configurations. Each row exhibits results obtained using a different amount of UNet operations. In our method, we apply four inversion steps, and a varying number of renoising iterations. As can be seen, the addition of renoising iterations gradually improves the reconstruction results. Conversely, employing more inversion steps proves insufficient for capturing all details in the image, as evident by the background of the car, or even detrimental to the reconstruction, as observed in the Uluro example.

Quantitative Results For the quantitative evaluation, we use the MS-COCO 2017 [25] validation dataset. Specifically, we retain images with a resolution greater than 420×420, resulting in a dataset containing 3,865 images.

We begin by evaluating both the sampler reversing approach and our ReNoise method, while varying the number of UNet operations during the inversion process and keeping the number of denoising steps fixed. This experiment is conducted using various models (SDXL, SDXL Turbo, LCM) and samplers. For all

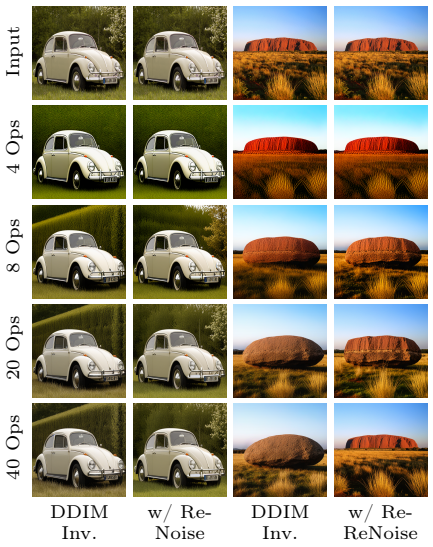

Fig. 8: Qualitative comparison between DDIM Inversion and our ReNoise method on SDXL Turbo (4 denoising steps). The first row displays the original images. The following rows display reconstructions from both methods, using the same number of UNet operations. DDIM employs smaller strides, while ReNoise utilizes more renoising steps.

Fig. 9: Ablation study on SDXL Turbo. The first row presents the input image. In each subsequent row, we show the reconstruction results using an additional component of our inversion method. The images in the bottom row represent the results obtained by our full method. NC stands for Noise Correction.

models, we utilize the DDIM [43] sampler. In addition, we employ the Ancestral-Euler scheduler for SDXL Turbo, and the default LCM sampler for LCM-LoRA. We set the number of denoising steps to 50 for SDXL, and to 4 for SDXL Turbo and LCM-LoRA. Quantitative results, using PSNR as the metric, are presented in Figure 11. We evaluate our method using different configurations. The x-axis refers to the number of UNet operations in the inversion process. Other metrics results are provided in the supplementary materials.

As depicted in the graphs, incorporating additional renoising iterations proves to be more beneficial for image reconstruction compared to adding more inversion steps. Note that the performance of the Ancestral-Euler and LCM samplers noticeably degrades when the number of inversion steps exceeds the number of denoising steps. Unlike DDIM, these samplers have $\Phi_t \approx 1$, resulting in an increase in the latent vector's norm beyond what can be effectively denoised in fewer steps. In this experiment, we maintain the same number of UNet operations for both ReNoise and the sampler reversing approach. However, in ReNoise, the number of inversion steps remains fixed, and the additional operations are utilized for renoising iterations, refining each point on the inversion trajectory. Consequently, our method facilitates improved reconstruction when using these noise samplers.

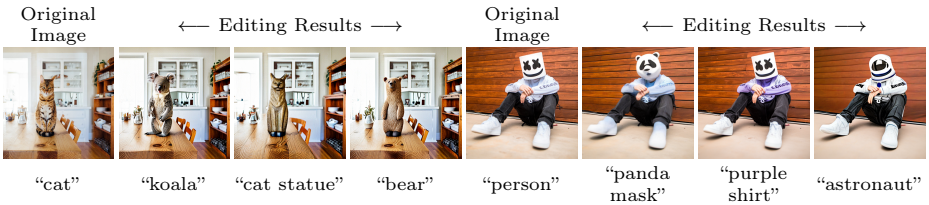

Fig. 10: LCM Editing Results. We showcase two examples of real images, each followed by three edits. The text below each edited image indicates the specific word or phrase replaced or added to the original prompt for that specific edit.

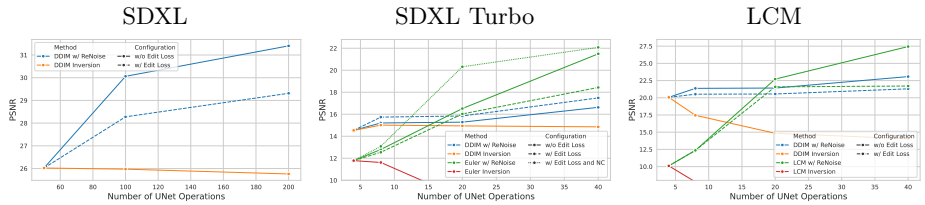

Fig. 11: Image reconstruction results comparing sampler reversing inversion techniques across different samplers (e.g., vanilla DDIM inversion) with our ReNoise method using the same sampler. The number of denoising steps remains constant. However, the number of UNet passes varies, with the sampler reversing approach increasing the number of inversion steps, while our method increases the number of renoising iterations. We present various configuration options for our method, including options with or without edit enhancement loss and Noise Correction (NC).

We continue by evaluating both the sampler reversing approach and our method while maintaining a fixed total number of UNet operations for the inversion and denoising processes combined. The results for SDXL with DDIM are presented in Table 1. The table displays various combinations of inversion, denoising, and renoising steps, totaling 100 UNet operations. Despite employing longer strides along the inversion and denoising trajectories, our ReNoise method yields improved reconstruction accuracy, as evident in the table. Furthermore, a reduced number of denoising steps facilitates faster image editing, especially since it commonly involves reusing the same inversion for multiple edits.

5.2 Reconstruction and Editability

In Figure 10, we illustrate editing results generated by our method with LCM LoRA [27]. These results were obtained by inverting the image using a source prompt and denoising it with a target prompt. Each row exhibits an image followed by three edits accomplished by modifying the original prompt. These edits entail either replacing the object word or adding descriptive adjectives to it. As can be seen, the edited images retain the details present in the original image. For instance, when replacing the cat with a koala, the details in the background are adequately preserved.

5.3 Ablation Studies

Figure 9 qualitatively demonstrates the effects of each component in our method, highlighting their contribution to the final outcome. Here, we use SDXL Turbo model [42], with the Ancestral-Euler sampler, which is non-deterministic. As our baseline, we simply reverse the sampler process. The reconstruction, while

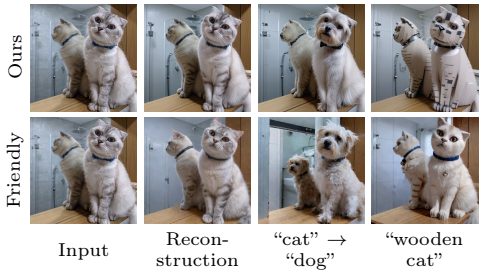

Fig. 12: Comparison with edit-friendly with SDXL Turbo. We inverted the image with the prompt "a cat is sitting in front of a mirror" and applied edits.

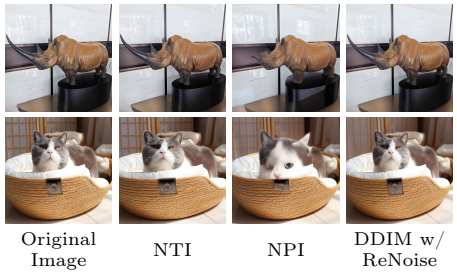

Fig. 13: Image reconstruction comparisons with SD. While NTI and our method achieve comparable results, ours demonstrates significant speed improvement.

semantically capturing the main object, fails to reproduce the image's unique details. For example, in the bird image, the reconstruction contains a bird standing on a branch, but the branch is in a different pose and the bird is completely different. Using 9 ReNoise iterations significantly improves the reconstruction, recovering finer details like the bird's original pose and branch texture. However, some subtle details, such as the bird's colors or the color in Brad Pitt's image, remain incomplete. Averaging the final iterations effectively incorporates information from multiple predictions, leading to a more robust reconstruction that captures finer details. Regularize the UNet's noise prediction with $\mathcal{L}_{\text{edit}}$ can introduce minor artifacts to the reconstruction, evident in the smoother appearance of the hair of the two people on the left, or in the cake example. Finally, we present our full method by adding the noise correction technique.

Table 2 quantitatively showcases the effect each component has on reconstruction results. As can be seen, the best results were obtained by our full method or by averaging the last estimations of z_t. Our final method also offers the distinct advantage of getting an editable latent representation.

In the Appendix, we present an ablation study to justify our editability enhancement and noise correction components.

5.4 Comparisons

Inversion for Non-deterministic Samplers. In Figure 12 we show a qualitative comparison with "an edit-friendly DDPM" [21] where we utilize SDXL Turbo [42]. Specifically, we assess the performance of the edit-friendly method alongside our ReNoise method in terms of both reconstruction and editing.

We observe that in non-deterministic samplers like DDPM, the parameter ρ_0 in Equation 1 equals zero. This means that in the final denoising step, the random noise addition is skipped to obtain a clean image. In long diffusion processes (e.g., 50-100 steps), the final denoising step often has minimal impact as the majority of image details have already been determined. Conversely, shorter diffusion processes rely on the final denoising step to determine fine details of the image. Due to focusing solely on noise correction to preserve the original image in the inversion process, edit-friendly struggles to reconstruct fine details of the image, such as the shower behind the cat. However, our ReNoise method finds an inversion trajectory that faithfully reconstructs the image and does not rely

solely on noise corrections. This allows us to better reconstruct fine details such as the shower. Additionally, encoding a significant amount of information within only a few external noise vectors, ϵ_t, limits editability in certain scenarios, See more examples in the appendix

Null-prompt Inversion Methods In Figure 13, we present a qualitative comparison between our method and null-text based inversion methods. For this comparison, we utilize Stable Diffusion [39] since these methods rely on a CFG [20] mechanism, which is not employed in SDXL Turbo [42]. Specifically, we compare DDIM Inversion [43] with one renoising iteration to Null-Text Inversion (NTI) [31] and Negative-Prompt Inversion (NPI) [30]. Both NTI and NPI enhance the inversion process by replacing the null-text token embedding when applying CFG. Our method achieves results comparable to NTI, while NPI highlights the limitations of plain DDIM inversion. This is because NPI sets the original prompt as the negative prompt, essentially resulting in an inversion process identical to plain DDIM inversion. Regarding running time, our ReNoise inversion process takes 13 seconds, significantly faster than NTI's 3 minutes. For comparison, plain DDIM inversion and NPI each take 9 seconds.

6 Conclusion

In this work, we have introduced ReNoise, a universal approach that enhances various inversion algorithms of diffusion models. ReNoise gently guides the inversion curve of a real image towards the source noise from which a denoising process reconstructs the image. ReNoise can be considered as a meta-algorithm that warps the trajectory of any iterative diffusion inversion process. Our experiments demonstrate that averaging the last few renoising iterations significantly enhances reconstruction quality. For a fixed amount of computation, ReNoise shows remarkably higher reconstruction quality and editability. The method is theoretically supported and our experiments reconfirm its effectiveness on a variety of diffusion models and sampling algorithms. Moreover, the method is numerically stable, and always converges to some inversion trajectory that eases hyperparameters adjustment.

Beyond the net introduction of an effective inversion, the paper presents a twofold important contribution: an effective inversion for few-steps diffusion models, which facilitates effective editing on these models.

A limitation of ReNoise is the model-specific hyperparameter tuning required for Edit Enhancement and Noise Correction. While these hyperparameters remain stable for a given model, they may vary across models, and tuning them is necessary to achieve high reconstruction quality while maintaining editability.

While ReNoise demonstrates the potential for editing few-step diffusion models, more extensive testing with advanced editing methods is needed. It is worth noting that no such editing has been demonstrated for the few-step diffusion models. We believe and hope that our ReNoise method will pave the way for fast and effective editing methods based on the few-steps models. We also believe that ReNoise can be adapted to the challenging problem of inverting video-diffusion models.

References

1. Numerical Differential Equation Methods, chap. 2, pp. 45–121. John Wiley and Sons, Ltd (2003). https://doi.org/https://doi.org/10.1002/0470868279.ch2, https://onlinelibrary.wiley.com/doi/abs/10.1002/0470868279.ch2
2. Abdal, R., Qin, Y., Wonka, P.: Image2stylegan: How to embed images into the stylegan latent space? In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE (Oct 2019). https://doi.org/10.1109/iccv.2019.00453, http://dx.doi.org/10.1109/ICCV.2019.00453
3. Abdal, R., Qin, Y., Wonka, P.: Image2stylegan++: How to edit the embedded images? In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (Jun 2020). https://doi.org/10.1109/cvpr42600.2020.00832, http://dx.doi.org/10.1109/cvpr42600.2020.00832
4. Alaluf, Y., Garibi, D., Patashnik, O., Averbuch-Elor, H., Cohen-Or, D.: Cross-image attention for zero-shot appearance transfer (2023)
5. Alaluf, Y., Patashnik, O., Cohen-Or, D.: Restyle: A residual-based stylegan encoder via iterative refinement. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE (Oct 2021). https://doi.org/10.1109/iccv48922.2021.00664, http://dx.doi.org/10.1109/ICCV48922.2021.00664
6. Alaluf, Y., Tov, O., Mokady, R., Gal, R., Bermano, A.H.: Hyperstyle: Stylegan inversion with hypernetworks for real image editing (2021)
7. Avrahami, O., Fried, O., Lischinski, D.: Blended latent diffusion. arXiv preprint arXiv:2206.02779 (2022)
8. Avrahami, O., Lischinski, D., Fried, O.: Blended diffusion for text-driven editing of natural images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18208–18218 (2022)
9. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions (2023)
10. Burden, R., Faires, J., Burden, A.: Numerical Analysis. Cengage Learning (2015)
11. Couairon, G., Verbeek, J., Schwenk, H., Cord, M.: Diffedit: Diffusion-based semantic image editing with mask guidance. ArXiv **abs/2210.11427** (2022)
12. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis (2021)
13. Dinh, T.M., Tran, A.T., Nguyen, R., Hua, B.S.: Hyperinverter: Improving stylegan inversion via hypernetwork. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
14. Epstein, D., Jabri, A., Poole, B., Efros, A.A., Holynski, A.: Diffusion self-guidance for controllable image generation (2023)
15. Ge, S., Park, T., Zhu, J.Y., Huang, J.B.: Expressive text-to-image generation with rich text. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2023)
16. Han, L., Wen, S., Chen, Q., Zhang, Z., Song, K., Ren, M., Gao, R., Stathopoulos, A., He, X., Chen, Y., Liu, D., Zhangli, Q., Jiang, J., Xia, Z., Srivastava, A., Metaxas, D.: Improving tuning-free real image editing with proximal guidance (2023)
17. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control (2022)
18. Hertz, A., Voynov, A., Fruchter, S., Cohen-Or, D.: Style aligned image generation via shared attention (2023)
19. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models (2020)
20. Ho, J., Salimans, T.: Classifier-free diffusion guidance (2022)

21. Huberman-Spiegelglas, I., Kulikov, V., Michaeli, T.: An edit friendly ddpm noise space: Inversion and manipulations (2023)
22. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. Advances in Neural Information Processing Systems **35**, 26565–26577 (2022)
23. Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models. In: Conference on Computer Vision and Pattern Recognition 2023 (2023)
24. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597 (2023)
25. Lin, T., Maire, M., Belongie, S.J., Bourdev, L.D., Girshick, R.B., Hays, J., Perona, P., Ramanan, D., Doll'a r, P., Zitnick, C.L.: Microsoft COCO: common objects in context. CoRR **abs/1405.0312** (2014)
26. Luo, S., Tan, Y., Huang, L., Li, J., Zhao, H.: Latent consistency models: Synthesizing high-resolution images with few-step inference (2023)
27. Luo, S., Tan, Y., Patil, S., Gu, D., von Platen, P., Passos, A., Huang, L., Li, J., Zhao, H.: Lcm-lora: A universal stable-diffusion acceleration module. arXiv preprint arXiv:2311.05556 (2023)
28. Meiri, B., Samuel, D., Darshan, N., Chechik, G., Avidan, S., Ben-Ari, R.: Fixed-point inversion for text-to-image diffusion models (2023)
29. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Guided image synthesis and editing with stochastic differential equations (2022)
30. Miyake, D., Iohara, A., Saito, Y., Tanaka, T.: Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models (2023)
31. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models (2022)
32. 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 (ICCV). pp. 15912–15921 (October 2023)
33. Parmar, G., Kumar Singh, K., Zhang, R., Li, Y., Lu, J., Zhu, J.Y.: Zero-shot image-to-image translation. In: Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings. SIGGRAPH '23, ACM (Jul 2023). https://doi.org/10.1145/3588432.3591513, http://dx.doi.org/10.1145/3588432.3591513
34. Parmar, G., Li, Y., Lu, J., Zhang, R., Zhu, J.Y., Singh, K.K.: Spatially-adaptive multilayer selection for gan inversion and editing. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (Jun 2022). https://doi.org/10.1109/cvpr52688.2022.01111, http://dx.doi.org/10.1109/CVPR52688.2022.01111
35. Patashnik, O., Garibi, D., Azuri, I., Averbuch-Elor, H., Cohen-Or, D.: Localizing object-level shape variations with text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
36. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis (2023)
37. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents (2022)
38. Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., Cohen-Or, D.: Encoding in style: a stylegan encoder for image-to-image translation. In:

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2021)
39. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models (2022)
40. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M.: Photorealistic text-to-image diffusion models with deep language understanding (2022)
41. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. In: International Conference on Learning Representations (2021)
42. Sauer, A., Lorenz, D., Blattmann, A., Rombach, R.: Adversarial diffusion distillation (2023)
43. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models (2022)
44. Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency models (2023)
45. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: International Conference on Learning Representations (2020)
46. Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., Cohen-Or, D.: Designing an encoder for stylegan image manipulation. ACM Trans. Graph. **40**(4) (jul 2021). `https://doi.org/10.1145/3450626.3459838`
47. Tumanyan, N., Geyer, M., Bagon, S., Dekel, T.: Plug-and-play diffusion features for text-driven image-to-image translation pp. 1921–1930 (June 2023)
48. Voynov, A., Hertz, A., Arar, M., Fruchter, S., Cohen-Or, D.: Anylens: A generative diffusion model with any rendering lens (2023)
49. Wallace, B., Gokul, A., Naik, N.: Edict: Exact diffusion inversion via coupled transformations. arXiv preprint arXiv:2211.12446 (2022)
50. Wu, C.H., De la Torre, F.: Unifying diffusion models' latent space, with applications to cyclediffusion and guidance. arXiv preprint arXiv:2210.05559 (2022)
51. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
52. Zhu, J., Shen, Y., Zhao, D., Zhou, B.: In-domain gan inversion for real image editing. In: Proceedings of European Conference on Computer Vision (ECCV) (2020)
53. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: Proceedings of European Conference on Computer Vision (ECCV) (2016)
54. Zhu, P., Abdal, R., Qin, Y., Femiani, J., Wonka, P.: Improved stylegan embedding: Where are the good latents? (2020)