Regularized Newton Raphson Inversion for Text-to-Image Diffusion Models

Dvir Samuel[1,3], **Barak Meiri**[1,2], **Nir Darshan**[1], **Shai Avidan**[2], **Gal Chechik**[3,4], **Rami Ben-Ari**[1]
[1]OriginAI, Tel-Aviv, Israel
[2]Tel-Aviv University, Tel-Aviv, Israel
[3]Bar-Ilan University, Ramat-Gan, Israel
[4]NVIDIA Research, Tel-Aviv, Israel

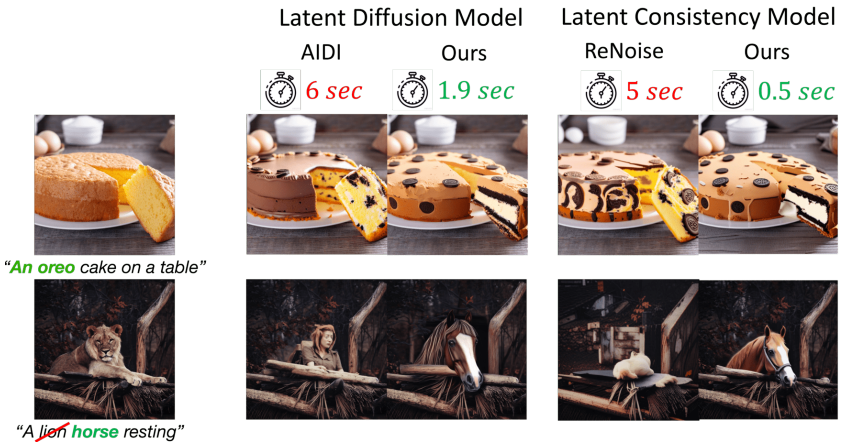

Figure 1: Image editing using our approach for inversion demonstrates significant speed-up and improved quality compared to previous state-of-the-art methods. Results are shown for both Latent Diffusion models and fast Latent consistency models.

Abstract

Diffusion inversion is the problem of taking an image and a text prompt that describes it and finding a noise latent that would generate the image. Most current inversion techniques operate by approximately solving an implicit equation and may converge slowly or yield poor reconstructed images. Here, we formulate the problem as finding the roots of an implicit equation and design a method to solve it efficiently. Our solution is based on Newton-Raphson (NR), a well-known technique in numerical analysis. A naive application of NR may be computationally infeasible and tends to converge to incorrect solutions. We describe an efficient regularized formulation that converges quickly to a solution that provides high-quality reconstructions. We also identify a source of inconsistency stemming from prompt conditioning during the inversion process, which significantly degrades the inversion quality. To address this, we introduce a prompt-aware adjustment of the encoding, effectively correcting this issue. Our solution, **Regularized Newton-Raphson Inversion**, inverts an image within 0.5 sec for latent consistency models, opening the door for interactive image editing. We further demonstrate improved results in image interpolation and generation of rare objects.

1 Introduction

Text-to-image diffusion models [1, 25–27] can generate diverse and high-fidelity images based on user-provided text prompts. These models are further used in several important tasks that require

*Correspondence to: Dvir Samuel <dvirsamuel@gmail.com>.

inversion, namely, discovering an initial noise (seed) that, when subjected to a backward (denoising) diffusion process along with the prompt, generates the input image. Inversion is used in various tasks including image editing [11], personalizaton [8, 9], seed noise interpolation [28], for semantic augmentation and generating rare concepts [29].

As inversion became a critical building block in various tasks, several inversion methods have been suggested. Denoising Diffusion Implicit Models (DDIM) [32] introduced a deterministic and fast sampling technique for image generation with diffusion models. However, DDIM *inversion* transforms an image back into a latent noise representation by approximating the inversion equation. Although this approximation makes it very fast, it also introduces an approximation error (as explained in section 3), causing noticeable distortion artifacts in inverted images. This is particularly noticeable in consistency models [19, 30], with large gaps between the diffusion time-steps, and where inference is achieved with only 2 – 4 DDIM steps. Several attempts have been made to address the inconsistencies in DDIM Inversion [22, 23, 33], which lead to poor reconstruction. AIDI [23] solved the DDIM-inversion implicit equation using fixed-point iterations [23], a numerical scheme with linear convergence rate, while [12] addresses the minimization of the residual error in that equation by gradient descent. Despite these methods showing improvements over previous approaches, their reconstruction and editing performance remain poor and often require extensive amounts of time.

In this paper, we frame the diffusion inversion problem as finding specific roots of an implicit function and propose a solution based on the Newton-Raphson (NR) numerical scheme [16]. Widely used in numerical analysis for its robustness and speed, NR (theoretical) quadratic convergence rate [2, 3, 6], enables very fast inversion. However, we find that a naive application of standard NR often fails to find the correct root, leading to significant distortions in the reconstructed images. We show that this issue can be resolved by regularizing the objective with a prior, derived from the training process of diffusion models. We name our approach RNRI, for ***R**egularized **N**ewton **R**aphson **I**nversion*. RNRI converges to a consistent inversion of an image in a small number of steps at each diffusion stage. In practice, 1-2 iterations are sufficient for convergence that yields significantly more accurate results than other inversion methods. RNRI requires no model training or finetuning, no prompt optimization, or any additional parameters. It can be combined with all pre-trained diffusion models, and we demonstrate its benefits to inversion of latent diffusion models (LDM) [26] and of latent consistency models (LCM) [30]. Figure 1 demonstrates the quality and speed of RNRI for editing, compared to a SoTA inversion method. Using latent diffusion models that require 50 DDIM steps, our approach can edit real images within 1.9 seconds. For latent consistency models requiring just 4 DDIM steps, the process converges in only 0.5 seconds. To the best of our knowledge, this speed allows users to edit images *on-the-fly*, using text-to-image models, for the first time.

In practice, high-quality editing using LDM or LCM requires applying classifier-free guidance with a large guidance scale. Similar to [13, 22, 24], we also identify that inversion with high guidance scale significantly degrades the results. When images are encoded into the latent space their encoding often miss-aligns with the prompt. We demonstrate that applying a brief adjustment to the encoding substantially enhances the inversion quality of all iterative methods.

We evaluate RNRI extensively. First, we directly assess the quality of inversions found with RNRI by measuring reconstruction errors, showing comparable results to [22, 33] but with $\times 4$ to $\times 12$ speedup gain. We then demonstrate the benefit of RNRI in two downstream tasks (1) In *Image editing*, RNRI smoothly changes fine details in the image in a consistent and coherent way, whereas previous methods struggle to do so. With RNRI inversion, we achieve the fastest editing times, where in the case of latent consistency models a single edit takes less than 0.5 seconds. This capability significantly enhances real-time editing possibilities. (2) In *Rare concept generation* with [29], and seed interpolation [28], that require diffusion inversion. In both of these tasks, RNRI yields more accurate seeds, resulting in superior generated images, both qualitatively and quantitatively, using the methods in [28, 29].

2 Related work

Text-to-image diffusion models [1, 25–27] translate random samples (seeds) from a high-dimensional space, guided by a user-supplied text prompt, into corresponding images. DDIM [32] is a widely used deterministic scheduler, that demonstrates the inversion of an image to its latent noise seed. When applied to inversion of text-guided diffusion models, DDIM inversion suffers from low reconstruction

accuracy that is reflected in further tasks, particularly when the classifier-free guidance constant is large [22]. This happens because it relies on a linear approximation, causing a propagation of errors that result in inaccurate image reconstruction and the loss of content. Recent studies [13, 22, 23, 33] address this limitation. Null-text inversion [22] optimizes the embedding vector of an empty string. This ensures that the diffusion process calculated using DDIM inversion, aligns with the reverse diffusion process. [21] replace the null-text embedding with a prompt embedding instead. This enhances convergence time and reconstruction quality but results in inferior image editing quality. In both [22] and [21], the optimized embedding must be stored, resulting in nearly 3 million additional parameters for each image (using 50 denoising steps of StableDiffusion [26]).

EDICT [33] introduced invertible neural network layers, specifically Affine Coupling Layers, to compute both backward and forward diffusion paths. However effective, it comes at the cost of prolonging inversion time. BDIA [34] introduced a novel integration approximation designed for EDICT, enhancing its computational efficiency while maintaining accurate diffusion inversion. Nevertheless, it still requires significantly more time (10 times longer than DDIM Inversion). AIDI [23] uses an accelerated fixed-point iteration technique at each inversion step to address the implicit function posed by DDIM equations. ExactDPM [13] utilizes gradient-based methods to achieve an effective inversion. ReNoise [10] further extends the work of [23] for Latent consistency models.

Alternative approaches proposed in [4, 15] suggest inverting with a DDPM scheduler instead of DDIM. They begin by constructing auxiliary images $x_1, ..., x_T$ and then extracting noise maps $z_1, ..., z_T$ to achieve error-free image reconstruction. Despite their effectiveness, the stochastic nature of these methods presents several challenges. Firstly, it necessitates storing $T + 1$ latents for every inverted image, resulting in an additional $16K \times T$ parameters. Secondly, the method is solely applicable to reconstruction and editing tasks. Lastly, DDPM inversion typically yields lower-quality results compared to DDIM inversion [14]. Since our focus is on DDIM deterministic inversion, these papers fall outside the scope of this paper.

3 Preliminaries

We first establish the fundamentals of Denoising Diffusion Implicit Models (DDIMs). In this model, a *backward pass* (denoising) is the process that generates an image from a seed noise. A *forward pass* is the process of adding noise gradually to an image until it becomes pure Gaussian noise. *Inversion* is similar to the forward pass but the goal is to end with a specific Gaussian noise that would generate the image if denoised.

Forward Pass in Diffusion Models. Latent diffusion and consistency models learn to generate images through a systematic process of iteratively adding Gaussian noise to a latent data sample until the data distribution is mostly noise. The data distribution is subsequently gradually restored through a reverse diffusion process initiated with a random sample (noise seed) from a Gaussian distribution. In more detail, the process of mapping a (latent) image to noise is a Markov chain that starts with z_0, and gradually adds noise to obtain latent variables $z_1, z_2, \ldots, z_T$, following $q(z_1, z_2, \ldots, z_T|z_0) = \Pi_{t=1}^{T} q(z_t|z_{t-1})$, where $\forall t : z_t \in \mathbb{R}^d$ with d denoting the dimension of the space. Each step in this process is a Gaussian transition

$$q(z_t|z_{t-1}) \sim \mathcal{N}(z_t, \sqrt{1-\beta_t} z_{t-1}, \beta_t I), \tag{1}$$

parameterized by a schedule $\beta_0, \beta_1, \ldots, \beta_T \in (0, 1)$.

Denoising Diffusion Implicit Models (DDIM). Sampling from diffusion models can be viewed alternatively as solving the corresponding diffusion Ordinary Differential Equations (ODEs) [18]. DDIM [32] scheduler, a popular deterministic scheduler, proposed denoising a latent noise vector in the following way:

$$z_{t-1} = \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} z_t - \sqrt{\alpha_{t-1}} \cdot \Delta\psi(\alpha_t) \cdot \epsilon_\theta(z_t, t, p), \tag{2}$$

where $\psi(\alpha) = \sqrt{\frac{1}{\alpha} - 1}$, and $\Delta\psi(\alpha_t) = \psi(\alpha_t) - \psi(\alpha_{t-1})$.

$\epsilon_\theta(z_t, t, p)$ is the output of a network that was trained to predict the noise to be removed.

DDIM inversion. We now focus on inversion in the latent representation. Given an image representation z_0 and its corresponding text prompt p, we seek a noise seed z_T that, when denoised, reconstructs

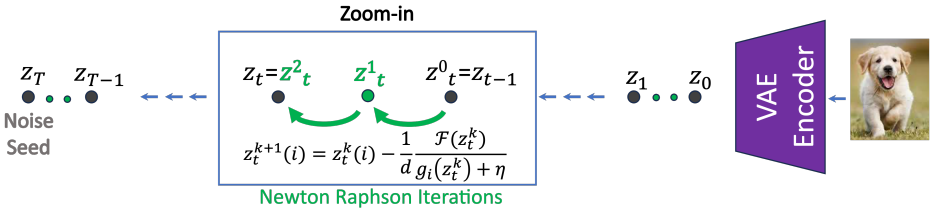

Figure 2: **Newton-Raphson Inversion** iterates over an implicit function (Eq. 8), at every time step in the inversion path. It starts with $z_t^0 = z_{t-1}$ and quickly converges (within 2 iterations) to z_t. Each box denotes one inversion step; black circles correspond to intermediate latents in the denoising process; green circles correspond to intermediate Newton-Raphson iterations.

the latent z_0. Several approaches were proposed for this task, and we focus on DDIM inversion. In this technique, Eq. (2) is rewritten as:

$$\begin{aligned} z_t &= f(z_t) \\ f(z_t) &:= \sqrt{\frac{\alpha_t}{\alpha_{t-1}}} z_{t-1} + \sqrt{\alpha_t} \cdot \Delta\psi(\alpha_t) \cdot \epsilon_\theta(z_t, t, p). \end{aligned} \tag{3}$$

DDIM inversion approximates this implicit equation in z_t by replacing z_t with z_{t-1}

$$\approx \sqrt{\frac{\alpha_t}{\alpha_{t-1}}} z_{t-1} + \sqrt{\alpha_t} \cdot \Delta\psi(\alpha_t) \cdot \epsilon_\theta(z_{t-1}, t, p). \tag{4}$$

The quality of the approximation depends on the difference $z_t - z_{t-1}$ (a smaller difference would yield a small error) and on the sensitivity of ϵ_θ to that z_t. See [7, 32] for details.

By applying Eq. (4) repeatedly for every denoising step t, one can invert an image latent z_0 to a latent z_T in the seed space. DDIM inversion is fast, but the approximation of Eq.(4) inherently introduces errors at each time step. As these errors accumulate, they cause the whole diffusion process to become inconsistent in the forward and the backward processes, leading to poor image reconstruction and editing [22, 23, 33]. This is particularly noticeable in consistency models with a small number of DDIM steps (typically 2-4 steps), where there's a significant gap between z_t and z_{t-1}, see Figure 5(b).

Iterative inversion optimization methods: Several papers proposed to improve the approximation using iterative methods [13, 23]. AIDI [23] proposed to directly solve Eq.(4) using fixed-point iterations [5], a widely-used method in numerical analysis for solving implicit functions. They solve $z = f(z)$ using fixed-point iterations. In a related way, [13] solves a more precise inversion equation, obtained by employing higher-order terms, using gradient descent.

4 Our method: Regularized Newton Raphson Inversion

DDIM inversion above often yields poor reconstruction accuracy. Proposed improvements described above have a linear convergence rate and may take many seconds to compute. In this paper we describe a faster and more robust alternative based on the well-known *Newton-Raphson* method (NR) [16]. Newton-Raphson is a method for iteratively finding the roots of a system of equations.

A naive Newton-Raphson approach. Consider first a naive way to apply NR to the inversion problem. We can define the following vector residual function,

$$r(z_t) := z_t - f(z_t) \tag{5}$$

where $r : \mathbb{R}^d \to \mathbb{R}^d$, and then seek its *zero-crossing* roots. That is, find those roots z_t for which $r(z_t) = \mathbf{0}$, meaning that the two vectors z_t, $f(z_t)$ are identical. For this problem, NR operates in the following way [6]. Given an initial guess z_0, It iterates $z_t^{k+1} = z_t^k - \left(J(z_t^k)\right)^{-1} \cdot r(z_t^k)$, where $(J(z_t^k))^{-1}$ presents the inverse of a Jacobian matrix $J \in \mathbb{R}^{d\times d}$ (all derivatives are w.r.t to z). In our case, applying this naive scheme is impractical, because the dimension of z is high ($d \approx 16K$ in StableDiffusion [26]), making it too expensive to compute the Jacobian in terms of time- and memory cost, and inverting it becomes practically infeasible.

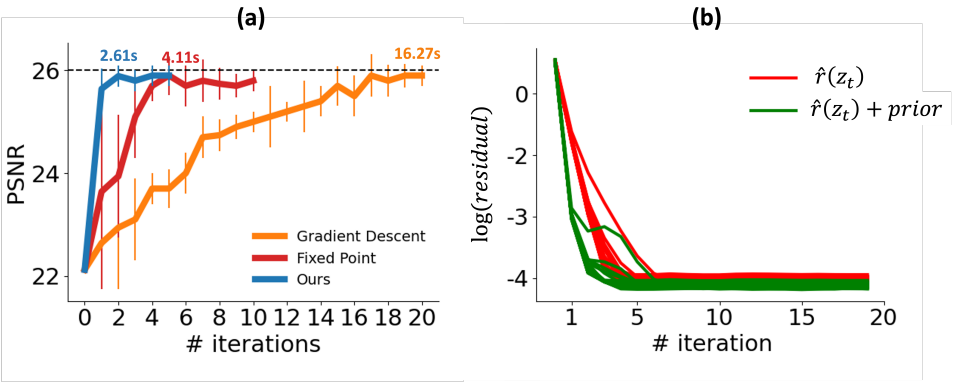

Figure 3: **(a) Convergence rate.** Comparison of iterative methods in an image inversion-reconstruction task over the COCO validation set. The mean PSNR of reconstructed images is plotted against the number of iterations. The dashed line represents the upper bound on reconstruction quality determined by the VAE in Stable Diffusion. Mean convergence time (in seconds) is denoted for each method. Our RNRI achieves a PSNR close to the upper limit and converges within only 1-2 iterations. **(b) Prior effect on convergence.** Incorporating our prior not only aids in finding the correct solution but also accelerates convergence.

To address this computational limitation, we propose to apply NR to a multi-variable, scalar function instead. Specifically, apply a norm over $r(z_t)$, $\hat{r} : \mathbb{R}^d \rightarrow \mathbb{R}_0^+$:

$$\hat{r}(z_t) \quad := \quad ||z_t - f(z_t)|| \tag{6}$$

and seek for solutions that satisfy $\hat{r}(z_t) = 0$. Here, $||\cdot||$ denotes a norm; we used L_1 to simply sum over all absolute values of $\hat{r}$. This reduction transforms the Jacobian matrix into a vector, making it easy and fast for computation (see derivation in Appendix A). Under some conditions, and particularly when there is only a single root in the ϵ-neighborhood of the initial guess, the Newton-Raphson method can be proved to have a quadratic convergence rate [16]. In fact, $\hat{r}$ introduces an underdetermined equation that may have multiple roots, and in practice, we find that solving Eq. 6 for $r(z_t) = 0$ often converges to a solution that is out-of-distribution for the diffusion model. This inversion solution leads to bad reconstruction. We address this issue in the next section.

4.1 Regularized Newton Raphson

To address the undetermined nature of our equation we suggest adding a regularization term, that can be viewed also as a soft constraint, to the Newton-Raphson objective. More precisely, since each step in the diffusion process follows a Gaussian distribution $q(z_t|z_{t-1})$ (Sec 3, Eq. 1), we add a prior over the values of z_t, by adding the negative log-likelihood as a regularizing penalty term to the objective. The objective is thus:

$$\mathcal{F}(z_t) := ||z_t - f(z_t)||_1 - \lambda \sum_{j=1}^{d} \log q_j(z_t|z_{t-1}) \tag{7}$$

Here, $\lambda > 0$ is a hyperparameter weighting factor for the regularization, and q_j is the j'th component of $q(z_t|z_{t-1})$ (refer to Appendix D for more details). We now seek a solution that satisfies $\mathcal{F}(z_t) = 0$. Note that this equation is strictly satisfied *iff* the residual $\hat{r}(z_t) = 0$ and the probabilities $q_j(z_t|z_{t-1}) = 1$, for all j.

The Newton Raphson iteration scheme for finding roots of our scalar function in Eq. (7) is given by (see derivation in Appendix A):

$$z_t^0 = z_{t-1}$$
$$z_t^{k+1}(i) = z_t^k(i) - \frac{1}{d} \frac{\mathcal{F}(z_t^k)}{g_i(z_t^k) + \eta}. \tag{8}$$

Here $g_i := \frac{\partial \mathcal{F}(z_t)}{\partial z_t(i)}$ is the partial derivative of $\mathcal{F}$ with respect to the variable (descriptor) $z_t(i)$. η is a small constant added for numerical stability and $i \in [0, 1, ..., d]$ indicates z's components. g can be

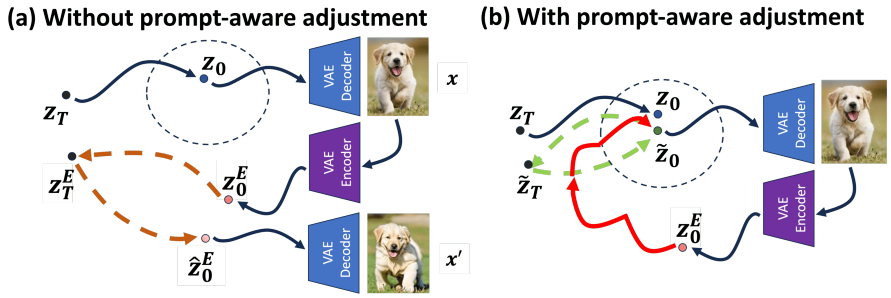

Figure 4: **Prompt-aware adjustment for more consistent inversion.** **(a)** A seed z_T is used for generating a latent image z_0, then decoded into an image x. If x is encoded into the latent space $z_0^E = E(x)$, its representation z_0^E may not be aligned with the prompt used for inversion and generation (brown dashed line). As a result, the reconstructed $\hat{z}_0^E$ differs significantly from z_0^E, yielding an inconsistent image x'. **(b)**: Inconsistency can be fixed using a short noising and denoising iterations (red curves), which yields $\tilde{z}_0$. Then, applying the complete backward-forward process (green curves) can generate a consistent $\tilde{z}_0$.

computed efficiently using automatic differentiation engines. We initialize the process with z from the previous diffusion timestep. Figure 2 illustrates this process. Note that while the solution of Eq. 7 matches the minimizer of $\mathcal{F}$, we employ the NR scheme to solve the equation. We call our approach **R**egualarized **N**ewton **R**aphson **I**nversion (RNRI). Figure 3 (left) depicts the effect of the number of running iterations on image reconstruction, in terms of PSNR, compared to other iterative methods. The figures illustrate that RNRI converges in only 1-2 iterations, achieving a PSNR close to the upper bound set by the diffusion model VAE. Figure 3 (right) shows the impact of prior on the residual. We depict residual curves for 50 random COCO images. Regularization yields superior outcomes, as evidenced by lower residuals.

4.2 Prompt-aware adjustment for consistent inversion

We now focus on an important subtlety that greatly impacts inversion quality. Our inversion process is designed to agree with a generation process for a prompt p. However, inversion is typically applied to photos and images that are not generated by the model. Such an image x is mapped into the latent z_0 in a way that is unaware of the prompt because the VAE encoder is prompt-agnostic. This means that the typical input to our inversion process may be inconsistent with the prompt we use. This problem is illustrated in Figure 4(a). Here, we randomly sampled z_T and ran the forward process with the prompt “A cute dog". This results in a denoised latent z_0 which is decoded into an image x. Then, z_0^E is obtained by passing x through the encoder E. In terms of L_2 distance, z_0^E differs only slightly from z_0, but it is not consistent with the prompt p (outside the dashed circle). As a result, applying inversion and reconstruction (dashed brown lines) from z_0^E yields a reconstructed latent $\hat{z}_0^E$ that is far from z_0. As a consequence, when $\hat{z}_0^E$ is decoded by the VAE, it results in a different image x' of “a cute dog". This illustrates that inversion becomes inconsistent when applied to prompt-agnostic latents. To quantify this effect, we repeated this experiment for 10k images and found the mean L_2 distance between z_0 and $\hat{z}_0^E$ ($||z_0 - \hat{z}_0^E||$) to be 99. In comparison, the mean distance between z_0 and z_0^E ($||z_0 - z_0^E||$) is only 16. This indicates that the inversion process produces latents that deviate significantly from the original z_0.

How can this inconsistency be resolved? We propose to employ a short preprocessing step of fixed-point iteration [5]. The objective is to find a fixed point z_0 for the implicit function $z_0 = h(z_0, p)$, with h representing the backward-forward process and p is the prompt that would be used for inversion. The initial guess for the fixed-point iterations is $z_0^E = E(x)$. The iteration count should be kept small to prevent excessive alteration of the latent z_0 as this could potentially affect the visual appearance of the original image. In practical terms, we observed that a brief cycle of fixed-point iteration, involving two noise-addition steps (forward process) followed by two denoising steps with the prompt p (backward process), proves effective.

This process is illustrated in Figure 4(b). Here, we pre-process z_0^E by two fixed-point iterations to obtain an improved latent $\tilde{z}_0$ (red lines). $\tilde{z}_0$ is decoded to a very similar image x and results with

consistent inversion (dashed green lines). For 10k images, we found that the L_2 distance between z_0 and $\tilde{z}_0$ (i.e., $||z_0 - \tilde{z}_0||$) is only 12. This illustrates that the proposed pre-processing method is capable of maintaining a highly consistent backward-forward process. More detailed in Appendix F.

Editing with large guidance scale.

Guidance scale plays a crucial role in editing tasks, with larger scales typically resulting in better and more natural edits [20]. Current iterative methods struggle to perform inversion with large guidance scales. This difficulty can be explained by the inconsistency between a given real image and a prompt, as described above. The guidance scale amplifies the influence of the prompt, intensifying this inconsistency and impeding convergence in current methods. [13] proposes the use of a different solver that requires significantly more iterations. [23] suggest inverting the image with a low guidance scale and then using a larger scale for editing specific regions identified by cross-attention maps. We demonstrate that prompt-aware adjustment enables iterative methods to perform inversion effectively by aligning a real image with a given prompt. Table 1 presents a comparison of inversion performance between different iterative methods, with and without prompt-aware adjustment. Guidance scale was set to 7.5 (the default for StableDiffusion v2.1). The results indicate that prompt-aware adjustment enhances the performance of all methods in terms of PSNR, number of iterations (#iter), and convergence time (T_{conv}).

Table 1: Comparing inversion performance of iterative methods with and without prompt-aware adjustment, using large guidance scale.

	PSNR	**#iter**	T_{conv} [sec]
w/o Prompt-Aware Adjustment			
Fixed-point [23]	10.2	diverge	inf
Gradient Descent [13]	12.5	diverge	inf
RNRI (ours)	19.3	3	2.8
w/ Prompt-Aware Adjustment			
Fixed-Point	25.4	5	4.1
Gradient Descent	25.5	20	16.3
RNRI (ours)	**25.8**	**2**	**2.61**

5 Experiments

We evaluate our approach on three main tasks: (1) **Image inversion and reconstruction:** Here, We assess the inversion fidelity by evaluating the quality of the reconstructed images. (2) **Real-time Image Editing:** We demonstrate the efficacy of our inversion scheme in image editing, highlighting its ability to facilitate real-time editing. (3) **Rare Concept Generation:** We illustrate how our method can be applied to improve the generation of rare concepts.

Compared Methods: We compared our approach with the following methods. (1) Standard **DDIM Inversion** [32]. (2) **Null-text** [22]. (3) **EDICT** [33]. (4) **AIDI** [23] (fixed-point based method). (5) **ReNoise** [10] (fixed-point based method) (6) **ExactDPM** [13] (gradient-based method). In all experiments, we used code and parameters provided by the respective authors.

Implementation details: To demonstrate the versatility of our approach, we conducted experiments on both the Latent Diffusion Model (LDM) using Stable Diffusion v2.1 and the Latent Consistency Model (LCM) using [30]. Input images were resized to 512 × 512. For LDM, we used 50 sampling steps, and for LCM, 4 steps. All baselines ran until convergence, while RNRI was run for *2 iterations per diffusion step*. All methods were tested on a single A100 GPU for a fair comparison. PyTorch's built-in gradient calculation was used for computing derivatives of Eq. 7.

5.1 Image Reconstruction

To evaluate the fidelity of our approach, we measure PSNR of images reconstructed from seed inversions. Specifically, we used the entire set of 5000 images from the MS-COCO-2017 validation dataset [17], along with their corresponding captions. For each image-caption pair, we first found the inverted latent and then used it to reconstruct the image using the same caption. Given that the COCO dataset provides multiple captions for each image, we used the first listed caption as a conditioning prompt. Figure 5 (left) shows PSNR of reconstructed images in relation to inversion time, demonstrating the performance of our approach compared to SoTA inversion methods. Time is measured on a single NVIDIA A100 GPU for all methods. The dashed black line is the upper

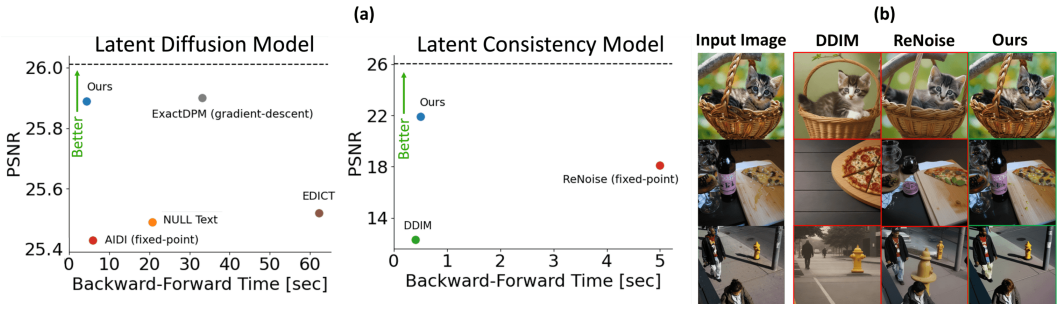

Figure 5: **(a) Inversion Results:** Mean reconstruction quality (y-axis, PSNR) and runtime (x-axis, seconds) of four inversion methods on the COCO2017 validation set. (**Left**) Performance on latent diffusion model. All methods run for 50 inversion steps. Our method reaches high PSNR with $\times 4$ to $\times 12$ shorter inversion-reconstruction time compared to other methods. (**Right**) Performance on latent consistency model. All methods run for 2 inversion steps. Our method archives the highest PSNR in less than 0.5 seconds, significantly faster than other methods, which typically take 7 to 10 times longer. This allows for real-time inversion and editing capabilities using our approach. **(b) Reconstruction qualitative results:** Comparing image inversion-reconstruction performance. While DDIM fails to preserve a close connection to the original image, ReNoise creates rather a blurry image reflected in lower PSNR.

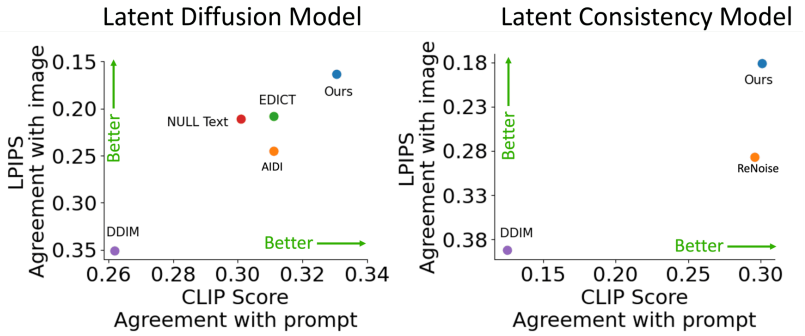

Figure 6: **Evaluation of editing performance:** RNRI achieves superior CLIP and LPIPS scores, indicating better compliance with text prompts and higher structure preservation.

bound induced by the Stable Diffusion VAE. It shows that our method is able to achieve comparable PSNR to recent methods and close to the upper bound of VAE, yet accomplishes this in the shortest amount of time. Furthermore, in contrast to other methods, our approach accurately inverts an image to a single latent vector without requiring additional memory. Figure 5(right) further demonstrates a qualitative comparison of reconstructed images. We provide in the Appendix E an additional analysis.

5.2 Real-Time Image Editing

Image editing from text is the task of making desirable changes in certain image regions, well blended into the image, while preserving the rest of the image intact. Current state-of-the-art methods for editing images start by inverting a real image into latent space and then operating on the latent. As a result, the quality of editing depends strongly on the quality of inversion [22].

We now evaluate the effect of inverting images with RNRI, compared to state-of-the-art inversion baselines. We used Prompt2Prompt [11] as our editing method. We show that RNRI outperforms all baselines in editing performance while requiring the shortest time (less than 0.5 seconds per image, in latent consistency models). This opens the door for **real-time editing capabilities**. Videos where we show real-time editing performance using RNRI can be found in supplementary material (attached zip file). We present below both qualitative and quantitative results.

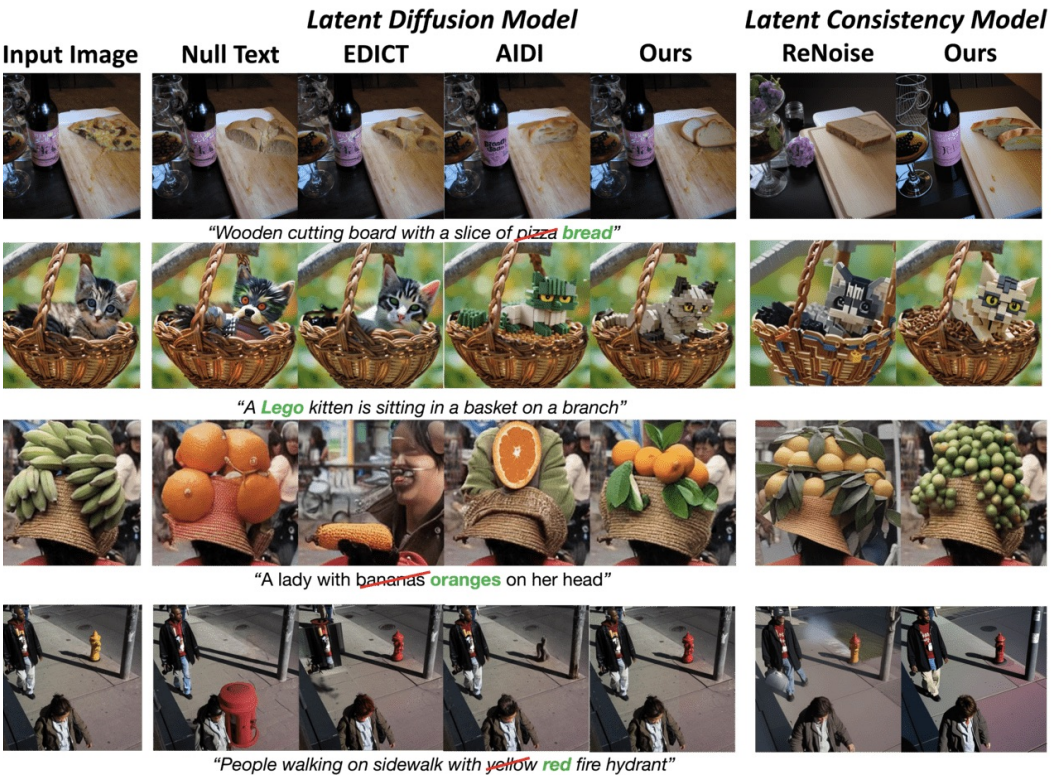

Figure 7: **Qualitative results of image editing**. RNRI edits images more naturally while preserving the structure of the original image. All baselines were executed until they reached convergence, whereas our approach was run for *two* iterations per diffusion step.

Qualitative results. Figure 7 gives a qualitative comparison between SoTA approaches and RNRI in the context of real image editing. RNRI excels in accurately editing with target prompts producing images with both high fidelity to the input image and adherence to the target prompt. The examples illustrate how alternative approaches may struggle to retain the original structure or tend to overlook crucial components specified in the target prompt. As an example, in the second row of Figure 7, RNRI exclusively converts the kitten into a Lego figure, whereas other methods either fail to achieve this or alter the basket and branch as well. In the third row, all LDM-based methods struggle to accurately substitute bananas with oranges, and ReNoise [10] alters the background. In contrast, RNRI accurately edits the object while maintaining the original background. More results in supp. E.

User Study. We further evaluated editing quality using human raters. We followed the evaluation process of [22] and asked human raters to rank edits made by three methods. 60 images were rated, 12 images were provided by the authors of [22] and the rest were randomly selected from the COCO dataset [17]. Three raters for each image were recruited through Amazon Mechanical Turk and were tasked to choose the image that better applies the requested text edit while preserving most of the original image. RNRI was preferred by raters in 40.4% of cases using Latent Diffusion Model (LDM), outperforming Null text [22](12.2%), EDICT [34] (17.8%), and AIDI [23] (29.6%). For Latent Consistency Model (LCM), RNRI was overwhelmingly favored, with 89.9% preference compared to ReNoise's [10] 10.1%.

LPIPS vs CLIP Score. Following [15, 33], we evaluate the results using two complementary metrics: LPIPS [35] to quantify the extent of structure preservation (lower is better) and a CLIP-based score to quantify how well the generated images comply with the text prompt (higher is better). Metrics are averaged across 100 MS-COCO images. Figure 6 illustrates that editing with RNRI yields a superior CLIP and LPIPS score, demonstrating the ability to perform state-of-the-art real image editing with superior fidelity. It further affirms the findings derived from the user study.

Table 2: **Image interpolation and centroid finding.** In interpolation, two images x^1, x^2 are inverted to generate images between their seeds z_T^1, z_T^2. In centroid-finding, a set of images is inverted to find their centroid. Acc and FID scores improved using RNRI as the inversion method.

	Interpolation [29]		**Centroid** [28]	
	ACC ↑	FID ↓	ACC↑	FID ↓
DDIM Inversion	51.59	6.78	67.24	5.48
AIDI [23]	52.01	6.13	68.14	5.32
RNRI (**ours**)	**54.98**	**5.91**	**70.18**	**4.59**

Table 3: **Inversion Quality Impact on Rare Concept Generation:** We assess image generation using a pre-trained classifier's accuracy, comparing NAO [28] (with DDIM inversion), AIDI [23], and RNRI. We report average per-class accuracy for Head (over 1M samples), Medium, and Tail (rare classes <10K samples). RNRI enhances rare and medium concept accuracy without sacrificing overall performance.

	ImageNet1k in LAION2B						
Methods	**Head** n=235 #>1M	**Medium** n=509 1M>#>10K	**Tail** n=256 10K>#	**Total Acc**	**FID**	$\hat{T}_{Init}$ (sec)	$\hat{T}_{Opt}$ (sec)
DDIM inversion	98.5	96.9	85.1	94.3	6.4	25	29
AIDI	98.5	97.0	85.3	94.4	6.9	24	28
RNRI (**ours**)	**98.6**	**97.9**	**89.1**	**95.8**	**6.3**	**17**	**25**

5.3 Seed Interpolation and Rare Concept Generation

Following the methodologies outlined in [28, 29], we attempt to generate images that are rare according to the diffusion training distribution. As demonstrated in these studies. This imbalance often leads to the generation of distorted or conceptually incorrect images. To address this issue, SeedSelect [29] takes a few images of a rare concept as input and uses a diffusion inversion module to iteratively refine the obtained seeds. These refined seeds are then used to generate new plausible images of the rare concept. NAO [28] extends this by introducing new paths and centroids for seed initialization. Both methods rely on DDIM Inversions, crucial for initial seed evaluation. Our work provides an alternative for precise inversion seeds, aiming for improved image quality and semantic accuracy. It's important to highlight that most inversion techniques [4, 15, 22, 33] are not applicable in this context as they necessitate extra parameters for image reconstruction, which impedes the straightforward implementation of interpolation and centroid identification as suggested by NAO.

We now provide results for seed interpolation and rare concept generation.

Interpolation and centroid finding: We evaluate RNRI for image interpolation and centroid finding. We follow the experimental protocol of [28] and compare images generated by DDIM with those produced by AIDI [23] and RNRI. Evaluation is conducted based on FID score and image accuracy, assessed using a pre-trained classifier. See details in [28]. Results are presented in Table 2. Notably, in comparison to DDIM and AIDI inversions, initializing with our seeds consistently results in higher-quality images both in interpolation paths and seed centroids.

Rare concept generation: We further show the effect of our seed inversions on the performance of NAO centroids with SeedSelect for rare concept generation. Specifically, we compared images generated by SeedSelect initialized with NAO using DDIM inversion (NAO(DDIM) in Table 3) and using RNRI (NAO(RNRI)). We followed the evaluation protocol of [28, 29] on ImageNet1k classes arranged by their prevalence in the LAION2B dataset [31]. This dataset serves as a substantial "in the wild" dataset employed in training foundation diffusion models, such as Stable Diffusion [26]. Image quality is measured by FID and accuracy of a pre-trained classifier. For more details see [28, 29].

The results, summarized in Table 3, demonstrate that our inversion method significantly boosts performance, both in accuracy and FID, compared to DDIM inversions. Furthermore, our inversions yield a more precise and effective initialization point for SeedSelect [29], resulting in notably quicker convergence (see Table 3) without compromising accuracy or image quality, in all categories (head, medium and tail) along with high gap at tail.

6 Limitations and Summary

Image inversion in diffusion models is vital for various applications like image editing, semantic augmentation, and generating rare concept images. Current methods often sacrifice inversion quality for computational efficiency, requiring significantly more compute resources for high-quality results. This paper presents Regularized Newton-Raphson Inversion (RNRI), a novel iterative approach that balances rapid convergence with superior accuracy, execution time, and memory efficiency. Using RNRI opens the door for real-time image editing. Despite its effectiveness, we encounter challenges. Despite demonstrating empirical convergence on a large-scale dataset, the model, like any iterative scheme, may fail to converge for certain images. We specifically identified such failure cases with incorrect prompts. (more details in Appendix G).

References

[1] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. eDiff-I: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 1, 2

[2] Aharon Ben-Tal and Arkadi Nemirovski. Lectures on modern convex optimization 2020. *SIAM, Philadelphia*, 2021. 2

[3] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 2

[4] Manuel Brack, Felix Friedrich, Katharina Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian Kersting, and Apolinaros Passos. Ledits++: Limitless image editing using text-to-image models. In *CVPR*, 2024. 3, 10

[5] J. Douglas Burden, Richard L.; Faires. Fixed-point iteration. 1985. 4, 6

[6] R.L. Burden, J.D. Faires, and A.M. Burden. *Numerical Analysis*. Cengage Learning, 2015. 2, 4

[7] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021. 4

[8] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *ICLR*, 2023. 2

[9] Rinon Gal, Moab Arar, Yuval Atzmon, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Designing an encoder for fast personalization of text-to-image models. *arXiv preprint arXiv:2302.12228*, 2023. 2

[10] Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. Renoise: Real image inversion through iterative noising. *arXiv preprint arXiv:2403.14602*, 2024. 3, 7, 9, 17

[11] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 2, 8

[12] Seongmin Hong, Kyeonghyun Lee, Suh Yoon Jeon, Hyewon Bae, and Se Young Chun. On exact inversion of dpm-solvers. *Accepted to CVPR*, 2024. 2

[13] Seongmin Hong, Kyeonghyun Lee, Suh Yoon Jeon, Hyewon Bae, and Se Young Chun. On exact inversion of dpm-solvers, 2024. 2, 3, 4, 7

[14] Yi Huang, Jiancheng Huang, Yifan Liu, Mingfu Yan, Jiaxi Lv, Jianzhuang Liu, Wei Xiong, He Zhang, Shifeng Chen, and Liangliang Cao. Diffusion model-based image editing: A survey. *ArXiv*, 2024. 3

[15] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. *arXiv:2304.06140*, 2023. 3, 9, 10

[16] Nick. "Thomas Simpson Kollerstrom. 'newton's method of approximation': An enduring myth. *The British Journal for the History of Science*, 1740. 2, 4, 5

[17] 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 *ECCV*, 2014. 7, 9

[18] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: a fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. 3

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

[20] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2022. 7

[21] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. *arXiv preprint arXiv:2305.16807*, 2023. 3

[22] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *CVPR*, 2023. 2, 3, 4, 7, 8, 9, 10

[23] Zhihong Pan, Riccardo Gherardi, Xiufeng Xie, and Stephen Huang. Effective real image editing with accelerated iterative diffusion inversion. In *ICCV*, 2023. 2, 3, 4, 7, 9, 10

[24] Zhimao Peng, Zechao Li, Junge Zhang, Yan Li, Guo-Jun Qi, and Jinhui Tang. Few-shot image recognition with knowledge transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 441–449, 2019. 2

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

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 2, 3, 4, 10

[27] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 2022. 1, 2

[28] Dvir Samuel, Rami Ben-Ari, Nir Darshan, Haggai Maron, and Gal Chechik. Norm-guided latent space exploration for text-to-image generation. *NeurIPS*, 2023. 2, 10, 13, 14

[29] Dvir Samuel, Rami Ben-Ari, Simon Raviv, Nir Darshan, and Gal Chechik. Generating images of rare concepts using pre-trained diffusion models. *AAAI*, abs/2304.14530, 2024. 2, 10, 11, 13, 14

[30] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation, 2023. 2, 7

[31] 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. *NeurIPS*, 2022. 10

[32] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, 2021. 2, 3, 4, 7

[33] Bram Wallace, Akash Gokul, and Nikhil Vijay Naik. EDICT: Exact diffusion inversion via coupled transformations. *CVPR*, 2022. 2, 3, 4, 7, 9, 10

[34] Guoqiang Zhang, Jonathan P Lewis, and W Bastiaan Kleijn. Exact diffusion inversion via bi-directional integration approximation. *arXiv:2307.10829*, 2023. 3, 9

[35] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 9

Supplemental Material

A Newton method for Multivariable Scalar Function

In this section, we will show the formulation of Newton's method for zero crossing of a multi-variable scalar function. For a vector $\mathbf{x} \in \mathbb{R}^n$ and a function $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ we are looking for the roots of the equation $f(\mathbf{x}) = 0$. Assuming that the function f is differentiable, we can use Taylor expansion:

$$f(\mathbf{x} + \delta) = f(\mathbf{x}) + \nabla f \, \delta^T + o(||\delta||^2) \tag{9}$$

For every $\delta, \mathbf{x} \in \mathbb{R}^n$. The idea of Newton's method in higher dimensions is very similar to the one-dimensional scalar function; Given an iterate $\mathbf{x}^k$, we define the next iterate $\mathbf{x}^{k+1}$ by linearizing the equation $f(\mathbf{x}^k) = 0$ around $\mathbf{x}^k$ as above, and solving the linearized equation $f(\mathbf{x}^k + \delta) = 0$. Dropping the higher orders in Equation (9) and solving the remaining linear equation for δ we get:

$$\delta = -\frac{1}{n} \left[\frac{1}{\frac{\partial f}{\partial x_1}}, \frac{1}{\frac{\partial f}{\partial x_2}}, ..., \frac{1}{\frac{\partial f}{\partial x_n}} \right] f(\mathbf{x}^k) \tag{10}$$

where x_i indicates the i-th component of vector $\mathbf{x}$. The above can be easily proven by substitution of δ into Equation (9). Using the relation, $\delta = \mathbf{x}^{k+1} - \mathbf{x}^k$ we get the final iterative scheme:

$$x_i^{k+1} = x_i^k - \frac{1}{n} \frac{f(\mathbf{x}^k)}{\frac{\partial f}{\partial x_i}(\mathbf{x}^k)} \tag{11}$$

Note that this equation is component-wise i.e. each component is updated separately, facilitating the solution.

B Seed Exploration & Rare concept generation

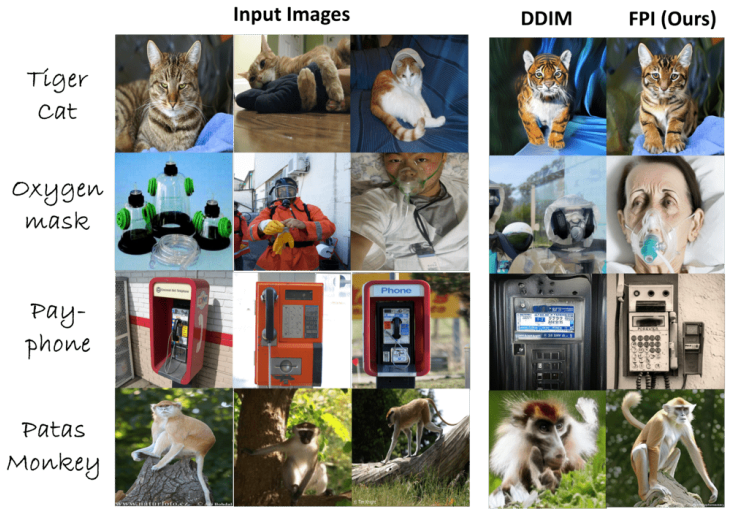

Fig. S 1: Generating rare concepts based on a few examples with the method of [28] that heavily depends on the diffusion-inversion quality. In our comparison, we evaluate RNRI alongside DDIM (refer to the discussion in the main paper). DDIM frequently struggles to produce a realistic representation of certain objects (such as Oxygen-mask or Patas Monkey) or an accurate depiction of specific concepts (like Tiger-cat or Pay-phone). However, the use of RNRI rectifies these issues in the results. For a detailed quantitative comparison, please refer to the main paper. See the main paper for a quantitative comparison.

Figure S1 further displays results for the rare-concept generation task introduced in [28, 29]. The objective is to enable the diffusion model to generate concepts rarely seen in its training set by using a few images of that concept. Both methods in [28, 29] utilize diffusion inversion for this purpose. In

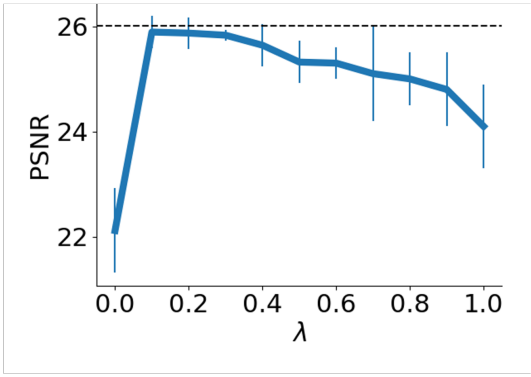

Fig. S 2: The influence of λ on reconstruction performance. We observe that employing no regularization ($\lambda = 0$) leads to poor reconstruction, while $\lambda = 0.1$ typically yields the highest reconstruction accuracy. As λ increases, reconstruction accuracy declines, possibly because assigning excessive weight to the prior undermines the root-finding objective.

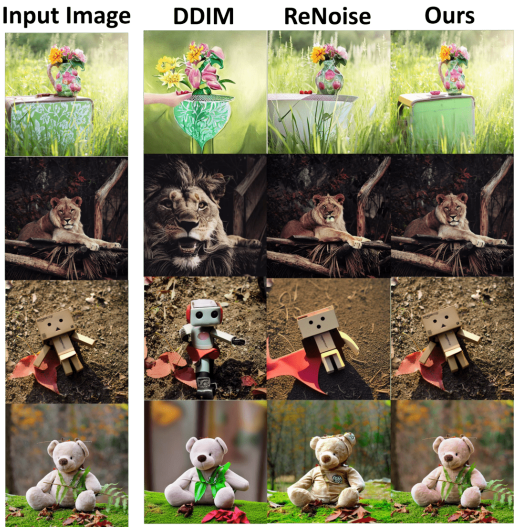

Fig. S 3: Qualitative comparison for **image reconstruction**. While DDIM reconstructed images are likely far from the original one, reconstruction from RenNoise still contains inconsistencies (the table at the 1st, lion image distortion or the texture in the teddy bear). RNRI shows nearly exact reconstruction. The accuracy in Reconstruction is later reflected in downstream tasks *e.g.* editing or long-tail generation. All methods were executed until they achieved convergence.

the main paper, we presented experiments showcasing the impact of our new inversion process on the outcomes of [28, 29].

We provide qualitative results based on NAO [28] centroid computation when utilizing RNRI, contrasting them with those obtained through DDIM inversions. The illustrations demonstrate that RNRI is capable of identifying high-quality centroids with correct semantic content compared to centroids found by DDIM.

C The Effect of the prior

In this section, we examine the impact of incorporating our regularization term in the objective function $\mathcal{F}$ in Eq. (7). We measure the effect in two aspects: 1) The remainder for the full objective function $\mathcal{F}$ in Eq. (7) at the predicted solution and the original residual (also known as fidelity term), $r(z_t) = z_t - f(z_t)$ in Eq. (6). Note that $r(z_t)$ indicates how accurately the DDIM equation is

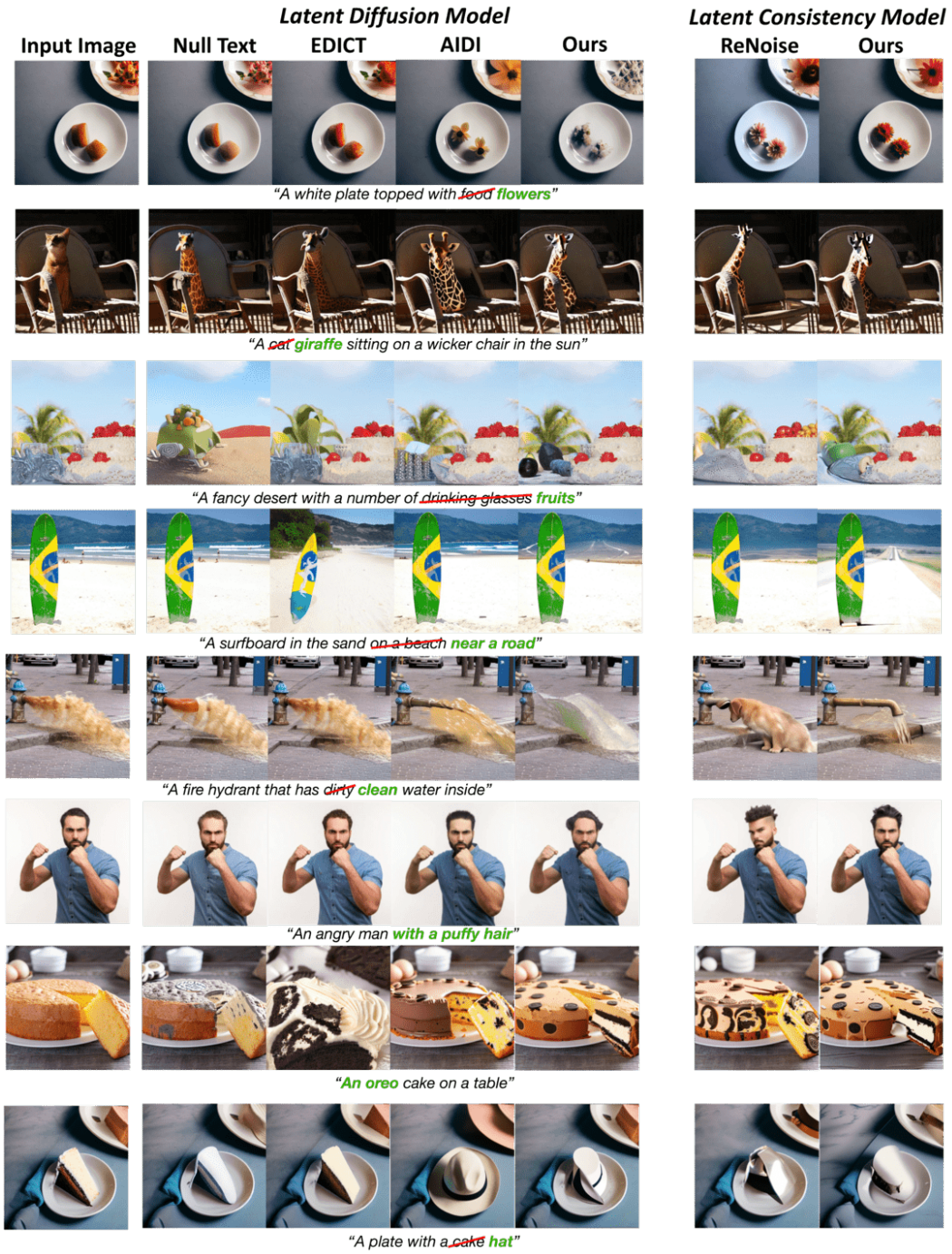

Fig. S 4: Qualitative comparison for **image editing:** Illustrates how alternative approaches may struggle to preserve the original structure or overlook crucial components specified in the target prompt, while RNRI succeeds in editing the image properly.

satisfied. Monitoring the residual r as shown in Figure S2(a) reveals that incorporating our prior not only aids in finding the correct solution (as shown by PSNR results) but also accelerates convergence.

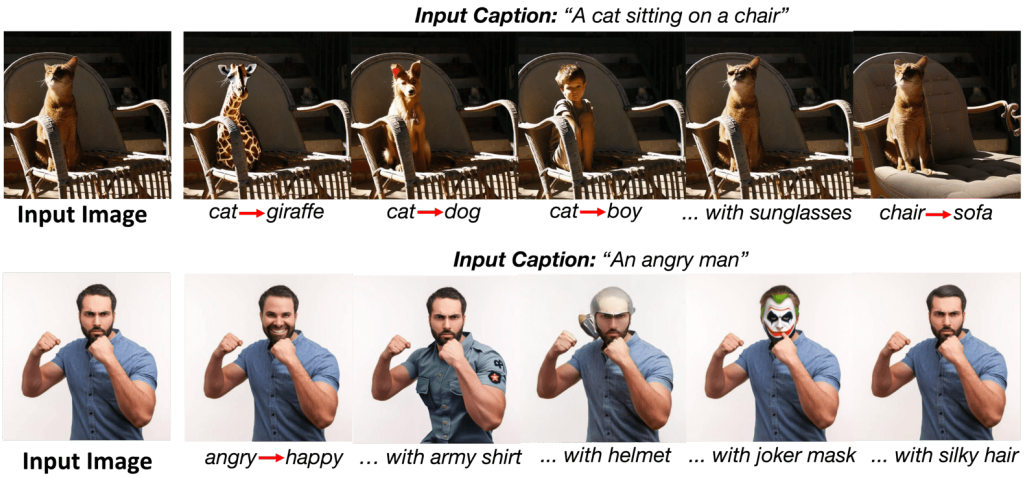

Fig. S 5: Various editing with same input: Note the RNRI capability in both subtle and extensive changes as one would expect from the particular prompt change.

Fig. S 6: Effect of prompt-aware adjustment on image inversion. Here we show the strong impact of our prompt-aware adjustment on the reconstruction results. Without the adjustment, the results are often unrealistic or far from the original image.

D Prior trade-off parameter λ

Figure S2(b) shows the impact of the weight parameter λ on the reconstruction accuracy. We observe that using no regularization ($\lambda = 0$) results in poor reconstruction, whereas setting $\lambda = 0.1$ achieves the highest reconstruction accuracy, underscoring the importance of the prior in the objective function. There is a gradual decrease with larger λ values, where at $\lambda = 1$ the PSNR starts to decline faster, likely because placing too much emphasis on the prior compromises the residual root-finding objective.

Fig. S 7: **Failure cases:** RNRI fails to converge where the prompt and image do not align.

E Additional Qualitative Results

In this section, we present additional qualitative comparisons involving RNRI and baseline methods. Figure 3 provides insights into inversion-reconstruction, showcasing that RNRI reaches a better reconstruction quality compared to DDIM and ReNoise [10].

Figures 4 provide further comparisons for real-image editing. These figures illustrate how alternative approaches may struggle to preserve the original structure or overlook crucial components specified in the target prompt. Looking at the Latent Diffusion results, in the first row of Figure S4, RNRI correctly transforms food into flowers, while other methods fail to do so. Row six shows an example of RNRI success (clean water instead of dirty), where other alternatives totally fail. The last row of Figure S4 depicts a failure case of our method, where it could not transform the cake into a hat as requested. Editing for Latent Consistency Models (LCM) is more challenging due to the extremely low number of time steps (aimed for speed-up). While ReNoise and RNRI achieve success in the first two rows, rows 3-6 demonstrate instances where ReNoise fails in comparison to RNRI. The last row illustrates a scenario where both methods fail.

Figure S5 shows the outcomes of various edits on the same image, providing additional confirmation that inversion with RNRI yields modification of the pertinent parts in the image while maintaining the original structure.

F The Effect of Prompt-Aware Adjustment

Figure S6 presents qualitative results demonstrating the impact of prompt-aware adjustment on image inversion and reconstruction, as proposed in Section 4.2 of the main paper. These results provide qualitative justification for the findings presented in the paper, highlighting that, without prompt-aware adjustment, reconstructed images differ from the original.

G RNRI Convergence

Here we provide information about an analysis to find failure cases for convergence We ran inversion and reconstruction, in scale, on COCO2017 (118k caption-image pairs) and found that in 95.4% of cases, RNRI successfully converged to a solution. Specifically, residuals went down from ~ 1 (for DDIM) to $< 10^{-4}$ indicating convergence, and the PSNR was > 25.7 indicating good solutions. The remaining 4.6% were samples with incorrect captions, see Fig. S7. These results imply that lack of convergence may indicate text and image miss-alignment which we consider for future work.