ReNoise: Real Image Inversion Through Iterative Noising – Supplementary Materials

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

1 Convergence Discussion

1.1 ReNoise As Contraction mapping

In this section, we discuss the proposed method from a convergence perspective. For completeness, we will present the entire discussion, including parts from the main paper.

Toy Example. We begin with the simple toy example, the diffusion of a shifted Gaussian. Given the initial distribution $\mu_0 \sim \mathcal{N}(a, I)$, where a is a non-zero shift value and I is the identity matrix. The diffusion process defines the family of distributions $\mu_t \sim \mathcal{N}(ae^{-t}, I)$, and the probability flow ODE takes the form $\frac{dz}{dt} = -ae^{-t}$ (see [3] for details). The Euler solver step at a state (z_t, t), and time step Δt moves it to $(z^{(1)}_{t+\Delta t}, t+\Delta t) = (z_t - ae^{-t} \cdot \Delta t, t+\Delta t)$. Notably, the backward Euler step at this point does not lead us to z_t. After applying the first renoising iteration, we get $(z^{(2)}_{t+\Delta t}, t+\Delta t) = (z_t - ae^{-(t+\Delta t)} \cdot \Delta t, t+\Delta t)$ and the backward Euler step at this point leads exactly to (z_t, t). Thus, in this simple example, the ReNoise algorithm successfully estimates the exact pre-image after a single step. While we cannot guarantee that in the general case, we will discuss some sufficient conditions for the algorithm's convergence and empirically verify them for the image diffusion model.

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 fixed z_{t-1}, the ReNoise mapping defined in Section 3 in the main paper can be written as $\mathcal{G} : z_t \to \text{InverseStep}(z_{t-1}, \epsilon_\theta(z_t, t))$. 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 maps after the denoising step to z_{t-1}, is a stationary point of this mapping. Given $z^{(1)}_t$, the first approximation of the next noise level z_t, our goal is to show that the sequence $z^{(k)}_t = \mathcal{G}^{k-1}(z^{(1)}_t), k \to \infty$ converges. As the mapping $\mathcal{G}$ is continuous, the limit point would be its stationary point. The definition of $\mathcal{G}$ gives us

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

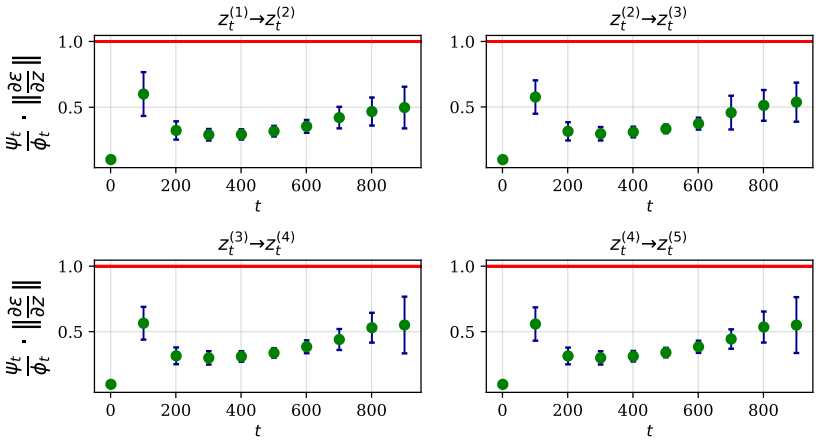

Fig. 1: Scaled Jacobian norm of the mapping $z_t^{(k)} \to z_t^{(k)}$ calculated for various noise levels t, iterations k. We report the average and the standard deviation calculated over 32 images. Values below 1 indicate the exponential convergence of the ReNoise algorithm.

where the norm is always assumed as the l_2-norm. For the ease of the notations, let us define $\Delta^{(k)} = z_t^{(k)} - z_t^{(k-1)}$. For convergence proof it is sufficient to show that the sum on 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 neighbour the convergence dynamics is defined by the scaled Jacobian norm $\frac{\psi_t}{\phi_t} \cdot \|\frac{\partial \epsilon_\theta}{\partial z}|_{z_t^{(k-1)}}\|$. Figure 1 shows 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, making ReNoise an almost identical operation.

Figure 7 in the main paper 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 $\mathcal{G}$.

Validation for the Averaging Strategy Notably, the proposed averaging strategy is aligned with the conclusions described in the main paper and also converges to the desired stationary point. To verify this claim we will show that if a sequence $z_{t+1}^{(k)}$ converges to some point z_{t+1}, then the averages $\bar{z}_{t+1}^{(k)} = \frac{1}{k}\sum_{i=1}^{k} z_{t+1}^{(i)}$ converges to the same point. That happens to be the stationary point of the operator $\mathcal{G}$. We demonstrate it with the basic and standard calculus. Assume that $z_{t+1}^{(k)} = \varepsilon_{t+1}^{(k)} + z_{t+1}$ with $\|\varepsilon_{t+1}^{(k)}\| \to 0$ as $k \to \infty$. For a fixed ε we need to show that there exists K so that $\|\bar{z}_{t+1}^{(k)} - z_{t+1}\| < \epsilon$ for any $k > K$. One has

$$\bar{z}_{t+1}^{(k)} - z_{t+1} = \sum_{i=1}^{k} \frac{\varepsilon_{t+1}^{(i)}}{k}$$

There exists m such that $\|\varepsilon_{t+1}^{(k)}\| < 0.5 \cdot \varepsilon$ once $k > m$. Then we have

$$\|\sum_{i=1}^{k} \frac{\varepsilon_{t+1}^{(i)}}{k}\| \leq \frac{\|\sum_{i=1}^{m} \varepsilon_{t+1}^{(i)}\|}{k} + \frac{\|\sum_{i=m+1}^{k} \varepsilon_{t+1}^{(i)}\|}{k} \leq$$

$$\frac{\|\sum_{i=1}^{m} \varepsilon_{t+1}^{(i)}\|}{k} + \frac{\varepsilon}{2}\frac{\cdot(k-m)}{k} \leq \frac{\|\sum_{i=1}^{m} \varepsilon_{t+1}^{(i)}\|}{k} + \frac{\varepsilon}{2}$$

given that m is fixed, we can always take k sufficiently large such that

$$\frac{\|\sum_{i=1}^{m} \varepsilon_{t+1}^{(i)}\|}{k} < \frac{\varepsilon}{2}$$

this ends the proof. The very same computation conducts a similar result if the elements' weights w_k are non-equal.

2 Implementation Details

We used BLIP-2 [4] to generate captions for the input images, which were then used as prompts for the diffusion models. In Table 1, we provide all hyperparameters of ReNoise inversion per model, optimized for the best reconstruction-editability trade-off.

Where $\{w_i\}$ are the renoising estimations averaging weights, and λ_{pair} and $\lambda_{\text{patch-KL}}$ are the weights we assign to each component of the edit enhancement loss:

$$\mathcal{L}_{\text{edit}} = \lambda_{\text{pair}} \cdot \mathcal{L}_{\text{pair}} + \lambda_{\text{patch-KL}} \cdot \mathcal{L}_{\text{patch-KL}}$$

Table 1: Implementation details of ReNoise with Stable Diffusion [9], SDXL [8], SDXL Turbo [10] and LCM LoRA [5].

Implementation details				
Model Name	**SD 1.4**	**SDXL**	**SDXL Turbo**	**LCM LoRA**
Noise Sampler	DDIM	DDIM	Ancestral-Euler	DDIM
No. denoising steps	50	50	4	4
No. renoising iterations	1	1	9	7
Weights for $t < 250$	$w_1, w_2 = 0.5$	$w_1, w_2 = 0.5$	$w_1, ..., w_4 = 0.25$	$w_1, ..., w_4 = 0.25$
Weights for $t > 250$	$w_2 = 1.0$	$w_2 = 1.0$	$w_8, ..., w_{10} = 0.33$	$w_6, ..., w_8 = 0.33$
λ_{pair}	10	10	10	20
$\lambda_{\text{patch-KL}}$	0.05	0.055	0.055	0.075

3 Additional Experiments

Editing Results With SDXL Turbo Figure 2 showcases additional image editing examples achieved using our ReNoise inversion method. These edits are accomplished by inverting the image with a source prompt, and then incorporating a target prompt that differs by only a few words during the denoising process.

Reconstruction and Speed We continue our evaluation of the reconstruction-speed tradeoff from Section 5.1 in the main paper. Figure 3 presents quantitative LPIPS results for the same configuration described in the main paper. As expected, LPIPS scores exhibit similar trends to the PSNR metric shown previously.

Image Editing Ablation Figure 5 visually illustrates the impact of edit enhancement losses and noise correction when editing inverted images using SDXL Turbo [10]. While achieving good reconstructions without the $\mathcal{L}_{\text{edit}}$ regularization, the method struggles with editing capabilities (second column). Although the $\mathcal{L}_{\text{edit}}$ regularization enhances editing capabilities, it comes at the cost of reduced reconstruction accuracy of the original image, as evident in the two middle columns. In the third column, we use $\mathcal{L}_{\text{KL}}$ as defined in pix2pix-zero [7]. While $\mathcal{L}_{\text{patch-KL}}$ surpasses $\mathcal{L}_{\text{KL}}$ in original image preservation, further improvements are necessary. These improvements are achieved by using the noise correction technique. To correct the noise, we can either override the noise ϵ_t in Equation 1 in the main paper (fifth column), or optimize it (sixth column). As observed, overriding the noise ϵ_t affects editability, while optimizing it achieves good results in terms of both reconstruction and editability. Therefore, in our full method, we use $\mathcal{L}_{\text{patch-KL}}$, $\mathcal{L}_{\text{pair}}$, and optimization-based noise correction.

Editing With ReNoise The ReNoise technique provides a drop-in improvement for methods (e.g., editing methods) that rely on inversion methods like DDIM [1], negative prompt inversion [6] and more. It seamlessly integrates with

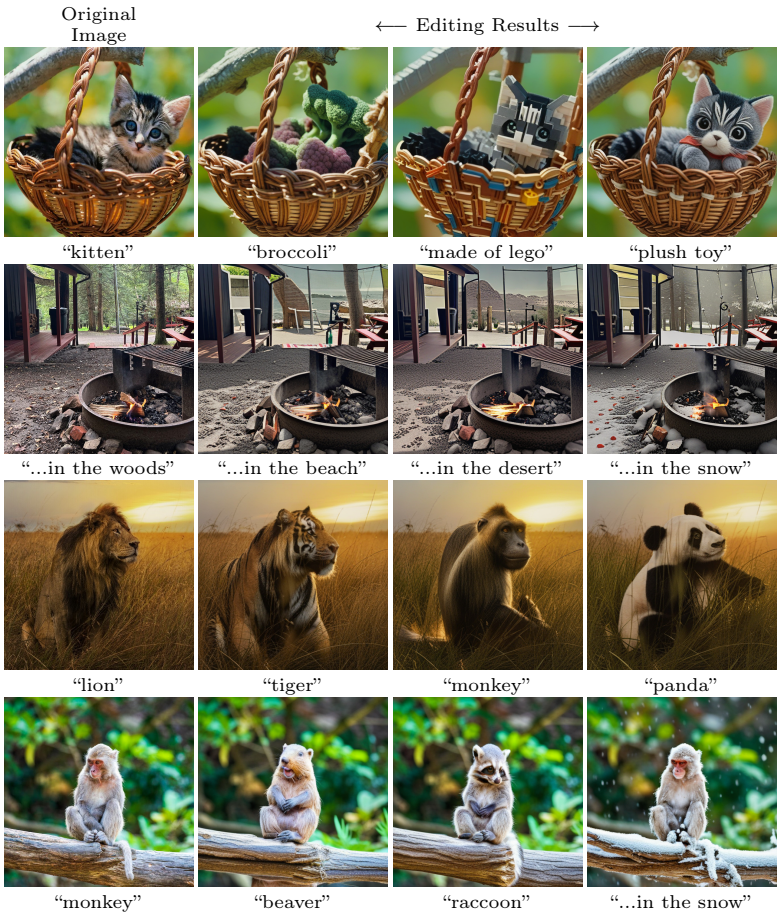

Fig. 2: SDXL Turbo editing results. Each row showcases one image. The leftmost image is the original, followed by three edited versions. The text below each edited image indicates the specific word or phrase replaced or added to the original prompt for that specific edit.

these existing approaches, boosting their performance without requiring extensive modifications. Figure 4 showcases image editing examples using Zero-Shot Image-to-Image Translation (pix2pix-zero) [7]. We compared inversions with both the pix2pix-zero inversion method and our ReNoise method. Our method demonstrably preserves finer details from the original image while improving editability, as exemplified by the dog-to-cat translation.

Inversion for Non-deterministic Samplers In Figure 6 we show more qualitative comparisons with "an edit-friendly DDPM" [2] where we utilize SDXL Turbo [10]. As can be seen, encoding a significant amount of information within only a few external noise vectors, ϵ_t, limits editability in certain scenarios, such as the ginger cat example. It is evident that the edit-friendly DDPM method struggles to deviate significantly from the original image in certain aspects while

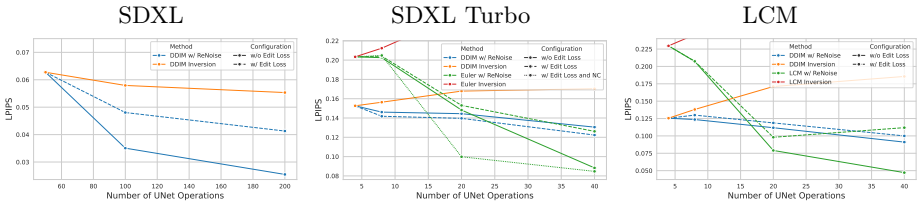

Fig. 3: Image reconstruction results comparing sampler reversing inversion techniques across different samplers (e.g., vanilla DDIM inversion) with our ReNoise method using the same sampler. The number of denoising steps remains fixed. However, the number of UNet passes varies, with the number of inversion steps increasing in the sampler reversing approach, and the number of renoising iterations increasing in our method. We present various configuration options for our method, including options with or without edit enhancement loss and Noise Correction (NC).

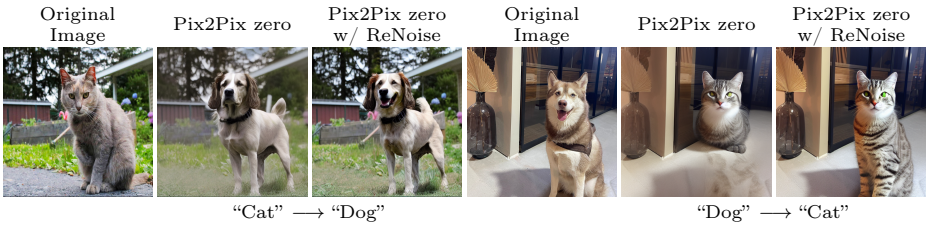

Fig. 4: Zero-Shot Image-to-Image Translation editing results. This figure compares editing results with Stable Diffusion [9] achieved using two inversion methods: pix2pix-zero [7] and our proposed ReNoise inversion. As observed, ReNoise inversion preserves image details while effectively incorporating the desired edits.

also failing to faithfully preserve it in others. For instance, it encounters difficulty in transforming the cat into a ginger cat while omitting the preservation of the decoration in the top left corner. In addition, the image quality of edits produced by edit-friendly DDPM is lower, as demonstrated in the dog example.

Improving DDIM Instabilities As mentioned in Section 3.1 of the main paper, DDIM inversion [1] can exhibit instabilities depending on the prompt. Figure 7 demonstrates this with image reconstruction on SDXL [8] using an empty prompt. Notably, incorporating even a single renoising iteration significantly improves inversion stability.

References

1. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis (2021)
2. Huberman-Spiegelglas, I., Kulikov, V., Michaeli, T.: An edit friendly ddpm noise space: Inversion and manipulations (2023)
3. Khrulkov, V., Ryzhakov, G., Chertkov, A., Oseledets, I.: Understanding ddpm latent codes through optimal transport. In: The Eleventh International Conference on Learning Representations (2022)

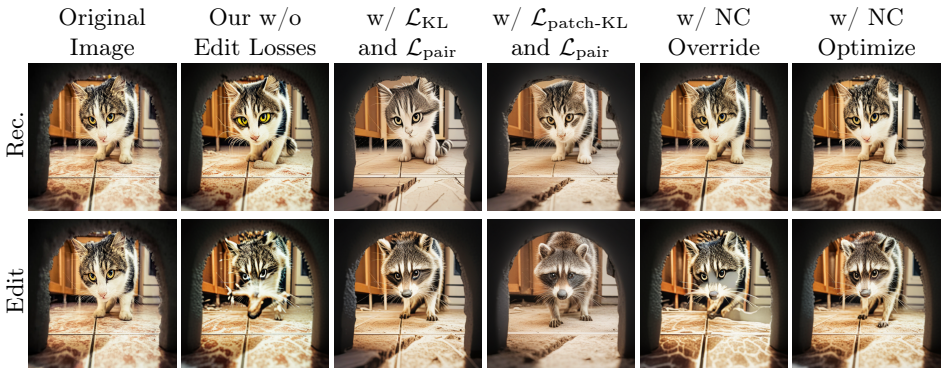

Fig. 5: Ablation study on SDXL Turbo for image editing. The first row displays reconstructed images, while the second row showcases edits replacing "cat" with "raccoon". Each column represents a different inversion configuration. From left to right, the second column demonstrates results with renoising iterations and estimations averaging. In the following two columns we employ different edit enhancement losses. Finally, the last two columns present results using both noise correction (NC) and $\mathcal{L}_{\text{edit}}$. In the fifth column, NC overrides the noise ϵ_t, while in the sixth column, we present the results of our full method, where NC optimizes ϵ_t.

4. 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)
5. 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)
6. Miyake, D., Iohara, A., Saito, Y., Tanaka, T.: Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models (2023)
7. 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
8. 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)
9. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models (2022)
10. Sauer, A., Lorenz, D., Blattmann, A., Rombach, R.: Adversarial diffusion distillation (2023)

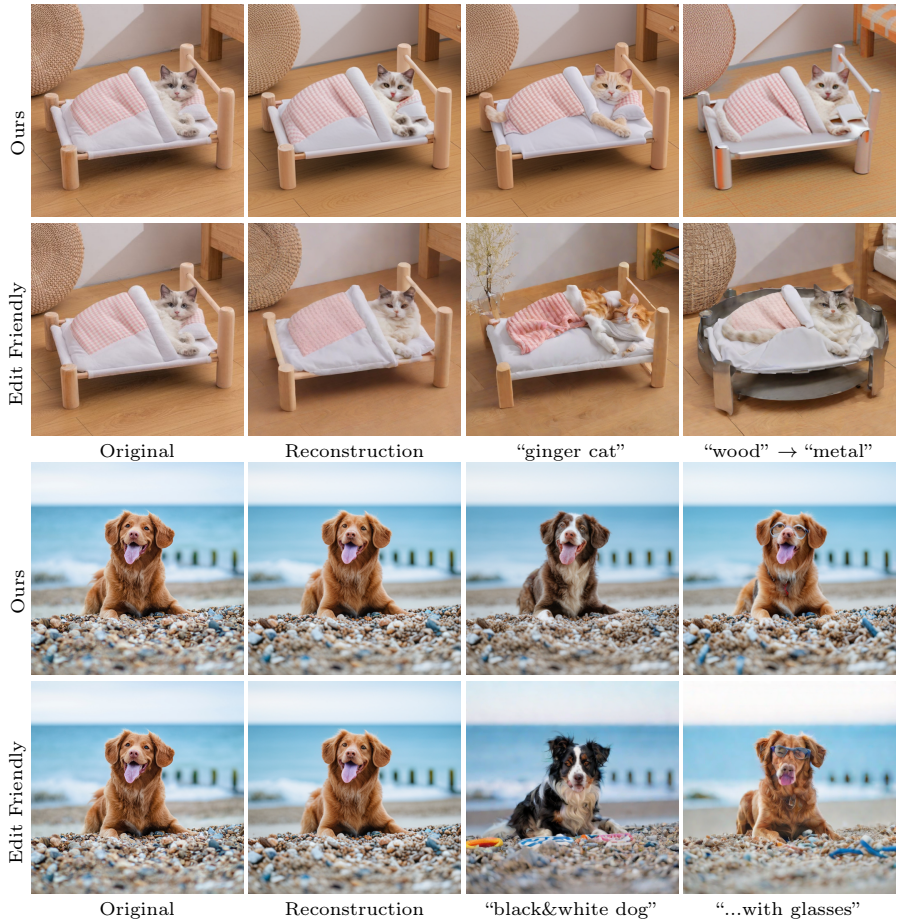

Fig. 6: Comparison with edit-friendly DDPM Inversion with SDXL Turbo. We invert two images with the prompts: "a cat laying in a bed made out of wood" (top) and "a dog sitting on the beach with its tongue out" (bottom) and apply two edits to each image.

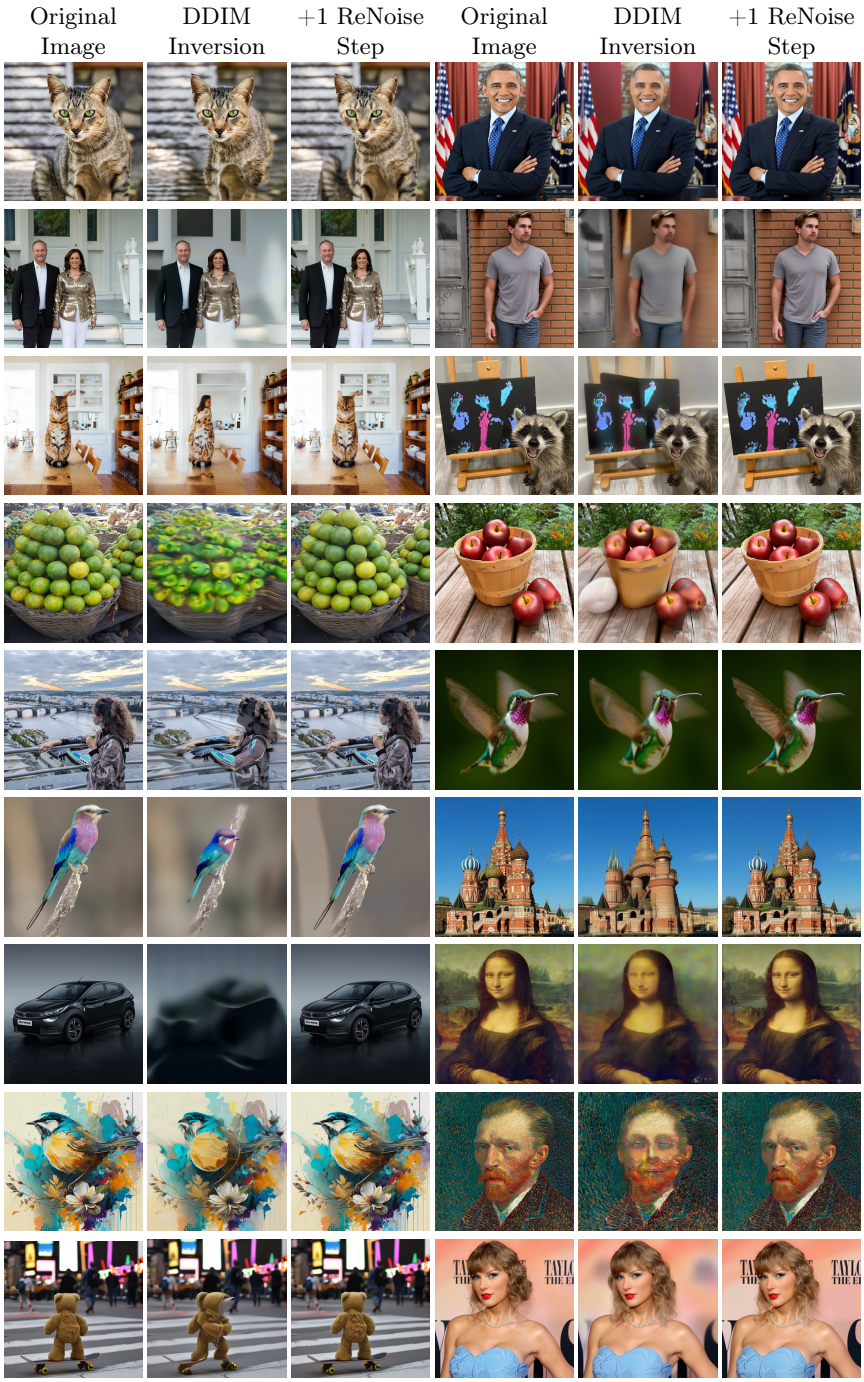

Fig. 7: Comparing reconstruction results with an empty prompt of plain DDIM inversion on SDXL to DDIM inversion with one ReNoise iteration.