

Negative-prompt Inversion: Fast Image Inversion for Editing with Text-guided Diffusion Models

Daiki Miyake^{1,2} Akihiro Iohara² Yu Saito² Toshiyuki Tanaka³

¹ The University of Tokyo, Japan ² DATAGRID Inc., Japan ³ Kyoto University, Japan

daiki.miyake@weblab.t.u-tokyo.ac.jp

{akihiro.iohara, yu.saito}@datagrid.co.jp

tt@i.kyoto-u.ac.jp

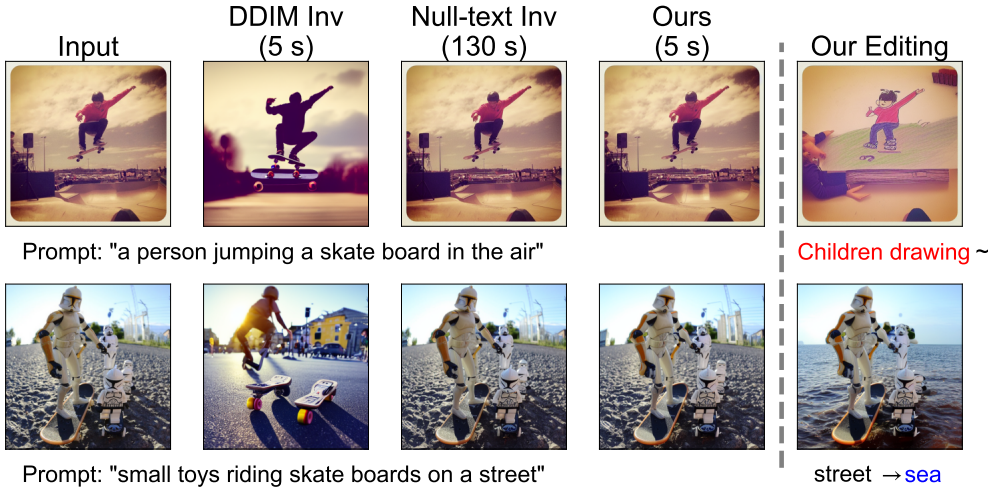


Figure 1. **Negative-prompt inversion.** Comparison in reconstruction fidelity and time between the proposed method (negative-prompt inversion; Ours), DDIM inversion [5, 26], and null-text inversion [19]. The rightmost column shows the results of image editing obtained using prompt-to-prompt [10] with our reconstruction.

Abstract

In image editing employing diffusion models, it is crucial to preserve the reconstruction fidelity to the original image while changing its style. Although existing methods ensure reconstruction fidelity through optimization, a drawback of these is the significant amount of time required for optimization. In this paper, we propose **negative-prompt inversion**, a method capable of achieving equivalent reconstruction solely through forward propagation without optimization, thereby enabling ultrafast editing processes. We experimentally demonstrate that the reconstruction fidelity of our method is comparable to that of existing methods, allowing for inversion at a resolution of 512 pixels and with 50 sampling steps within approximately 5 seconds, which is more than 30 times faster than null-text inversion. Reduction of the computation time by the proposed method further allows us to use a larger number of sampling steps in diffusion models to improve the reconstruction fidelity with a

moderate increase in computation time.

1. Introduction

Diffusion models [11] are known to yield high-quality results in the fields of image generation [5, 11, 23, 25, 27, 28], video generation [1, 9, 13, 14], and text-to-speech conversion [2, 3]. Text-guided diffusion models [16] are diffusion models conditional on given texts (“prompts”), which can generate data with various modalities that fit well with the prompts. It is known that by strengthening the text conditioning through classifier-guidance [5] or classifier-free guidance (CFG) [12], the fidelity to the text can be improved further. In image editing using text-guided diffusion models, elements in images, such as objects and styles, can be changed with high quality and diversity guided by text prompts.

In applications based on image editing methods, one must be able to generate images that are of high fidelity to

original images in the first place, including reproduction of their details, and then one will be able to perform appropriate editing of images according to the prompts therefrom. To achieve high-fidelity image generation, most existing research exploits optimization of parameters such as model weights, text embeddings, and latent variables, which results in high computational costs and memory usage.

In this paper, we propose a method that can obtain latent variables and text embeddings yielding high-fidelity reconstruction of real images while using only forward computations. Our method requires neither optimization nor backpropagation, enabling ultrafast processing and reducing memory usage. The proposed method is based on null-text inversion [19], which has the denoising diffusion implicit model (DDIM) inversion [5, 26] and CFG as its principal building blocks. Null-text inversion improves the reconstruction accuracy by optimizing an embedding which is used in CFG so that the diffusion process calculated by DDIM inversion aligns with the reverse diffusion process calculated using CFG. We discovered that the optimal embedding obtained by this method can be approximated by the embedding of the conditioning text prompt, and that editing also works by using an embedding of a source prompt instead of the optimized embedding.

Figure 1 shows a comparison between the proposed method and existing ones. Our method generated high-fidelity reconstructions when a real image and a corresponding prompt were given. DDIM inversion had noticeably lower reconstruction accuracy. Null-text inversion achieved high-quality results, nearly indistinguishable from the input image, but required much longer computation time. The proposed method, which we call **negative-prompt inversion**, allows for computation at the same speed as DDIM inversion, while achieving accuracy comparable to null-text inversion. Furthermore, combining our method with image editing methods such as prompt-to-prompt [10] allows ultrafast single-image editing (Editing).

We summarize our contributions as follows:

1. We propose a method for ultrafast reconstruction of real images with diffusion models, with no need of optimization at all.
2. We experimentally demonstrate that our method achieves visually equivalent reconstruction quality to existing methods while enabling a more than 30-fold increase in processing speed.
3. Combining our method with existing image editing methods like prompt-to-prompt allows ultrafast real image editing.

2. Related work

Image editing by diffusion models. In the field of image editing using diffusion models such as Imagen [25] and Stable Diffusion [23], Imagic [15], UniTune [30], and SINE [34] are models for editing compositional structures, as well as states and styles of objects, in a single image. These methods ensure fidelity to original images via fine-tuning models and/or text embeddings.

Prompt-to-prompt [10], another image editing method based on diffusion models, reconstructs original images via making use of null-text inversion. Null-text inversion successfully reconstructs real images by optimizing the null-text embedding (the embedding for unconditional prediction) at each prediction step. All these methods attempt to reconstruct real images by incorporating an optimization process, which typically takes several minutes to edit a single image.

Plug-and-Play [29] edits a single image without optimization. It obtains latent variables corresponding to the input image using DDIM inversion and reconstructs it according to the edited prompt, inserting attention and feature maps to preserve image structures. Our inversion method is independent of editing methods, allowing for the freedom to choose an editing method to be combined with, while maintaining a high-quality image structure regardless of the chosen editing method.

Image reconstruction by diffusion models. Textual Inversion [6] and DreamBooth [24] are methods that reconstruct common concepts from a few real images by fine-tuning the model. On the other hand, ELITE [32] and Encoder for Tuning (E4T) [7] seek text embeddings that reconstruct real images using an encoder. The former ones are aimed at concept acquisition, making them difficult to apply to reconstruction of the original image with high fidelity. Although the latter ones require less computation time compared with the former ones, the ease of editing operations is limited, as the corresponding text is not explicitly obtained.

Some previous works [4, 8] can reconstruct images without optimization in the inference stage. To improve reconstruction quality, noise map guidance [4] guides a path of the reverse diffusion process to align with the forward diffusion process using its gradient. On the other hand, ReNoise [8] improves reconstruction quality by using the backward Euler method (or the implicit Euler method) for inversion.

The proposed method realizes nearly the same reconstruction as null-text inversion, but with only forward computation, enabling image editing in just a few seconds. By combining our method with image editing methods such as prompt-to-prompt, it becomes possible to achieve flexible

and advanced editing using text prompts.

Note that there is an existing implementation [20] employing a similar idea to the proposed method. We would like to emphasize, however, that our work is the first to justify the proposed method both theoretically and experimentally.

3. Method

3.1. Overview

In this section, we describe our method for obtaining latent variables and text embeddings which reconstruct a real image using diffusion models without optimization. Our goal is that when given a real image I and an appropriate prompt P , we calculate latent variables (z_t) , where t is the index for the diffusion steps, in the reverse diffusion process so as to reconstruct I .

3.2. DDIM inversion

A diffusion model has a forward diffusion process over diffusion steps from 0 to T (e.g., $T = 1000$ in [11]), which degrades the representation z_0 of an original sample into a pure noise z_T , and an associated reverse diffusion process, which generates z_0 from z_T . In the training process, a degraded representation z_t for $t \in \{1, \dots, T\}$ is calculated by adding noise ϵ to z_0 , and the model is trained to predict the velocity field $\epsilon(z, t)$ at (z, t) associated with the Fokker-Planck equation governing the diffusion process. It should be noted that, although the added noise ϵ is random, the velocity field $\epsilon(z, t)$, to be learned by the model, is deterministic. See Appendix A.1, especially Proposition 1, for more details about the velocity field. In text-guided diffusion models, the model is further conditioned by an embedding C of a text prompt P , which is obtained via a text encoder like CLIP [22]. The loss function is the mean squared error (MSE) between the predicted velocity ϵ_θ and the actual noise ϵ ,

$$L(\theta) = \mathbb{E}_{t \sim U(1, T), \epsilon \sim \mathcal{N}(\mathbf{0}, I)} \|\epsilon - \epsilon_\theta(z_t, t, C)\|_2^2,$$

where $U(1, T)$ denotes the uniform distribution on the set $\{1, \dots, T\}$, and where $\mathcal{N}(\mu, \Sigma)$ denotes the multivariate Gaussian distribution with mean μ and covariance Σ . Minimizing the loss $L(\theta)$ with respect to the model parameter θ is expected to yield a model $\epsilon_\theta(z, t, C)$ which well approximates the conditional velocity field $\epsilon(z, t, C)$.

Stable Diffusion [23] considers diffusion processes in a latent space: during the training process, a latent representation z_0 is obtained by passing a sample x_0 through an encoder. In the inference stage, on the other hand, a sample x_0 is generated by passing the generated latent representation z_0 through a decoder.

CFG is used to strengthen text conditioning. During the computation of the reverse diffusion process, the null-text

embedding \emptyset , which corresponds to the embedding of a null text “”, is used as a reference for unconditional prediction to enhance the conditioning:

$$\tilde{\epsilon}_\theta(z_t, t, C, \emptyset) = \epsilon_\theta(z_t, t, \emptyset) + w(\epsilon_\theta(z_t, t, C) - \epsilon_\theta(z_t, t, \emptyset)), \quad (1)$$

where the guidance scale $w \geq 0$ controls strength of the conditioning.

In the inference phase, DDIM [26] iteratively calculates from the latent variable z_t at the diffusion step t the latent variable z_{t-1} at the diffusion step $(t-1)$ via

$$z_{t-1} = \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} z_t + \sqrt{\alpha_{t-1}} \left(\sqrt{\frac{1}{\alpha_{t-1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1} \right) \times \epsilon_\theta(z_t, t, C), \quad (2)$$

where $\alpha := (\alpha_1, \dots, \alpha_T) \in \mathbb{R}_{\geq 0}^T$ are hyper-parameters to determine noise scales at T diffusion steps. The forward process can also be represented in terms of $\epsilon_\theta(z_t, t, C)$ by inverting the reverse diffusion process (DDIM inversion) [5, 26], as

$$z_{t+1} = \sqrt{\frac{\alpha_{t+1}}{\alpha_t}} z_t + \sqrt{\alpha_{t+1}} \left(\sqrt{\frac{1}{\alpha_{t+1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1} \right) \times \epsilon_\theta(z_t, t, C). \quad (3)$$

3.3. Null-text inversion

DDIM is known to work well: Given an original sample, by performing the forward process starting from the representation z_0 of the sample to obtain z_T and then by inverting the forward process, one can reconstruct the original sample with high fidelity without CFG (i.e., $w = 1$ in (1)). Since CFG is useful to strengthen the text conditioning, it is desirable if one can reconstruct original samples well even when one uses CFG (i.e., $w > 1$). Simple application of CFG, however, degrades the fidelity of reconstructed samples. Null-text inversion enables us to faithfully reconstruct given samples even when using CFG, by optimizing the null-text embedding \emptyset at each diffusion step t .

In null-text inversion, we first calculate the sequence of latent variables $(z_t^*)_{t \in \{1, \dots, T\}}$ from z_0 via DDIM inversion. Next, we do initialization with $\bar{z}_T = z_T^*$ and $\emptyset_T = \emptyset$. We then iteratively optimize \emptyset_t for $t = T$ to 1 as follows: At each diffusion step t , assuming that we have \bar{z}_t , one calculates $z_{t-1}(\bar{z}_t, t, C, \emptyset_t)$ via DDIM (2) and CFG (1) with the null-text embedding \emptyset_t as

$$\begin{aligned} & z_{t-1}(\bar{z}_t, t, C, \emptyset_t) \\ &= \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} \bar{z}_t + \sqrt{\alpha_{t-1}} \left(\sqrt{\frac{1}{\alpha_{t-1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1} \right) \\ & \quad \times \tilde{\epsilon}_\theta(\bar{z}_t, t, C, \emptyset_t). \end{aligned} \quad (4)$$

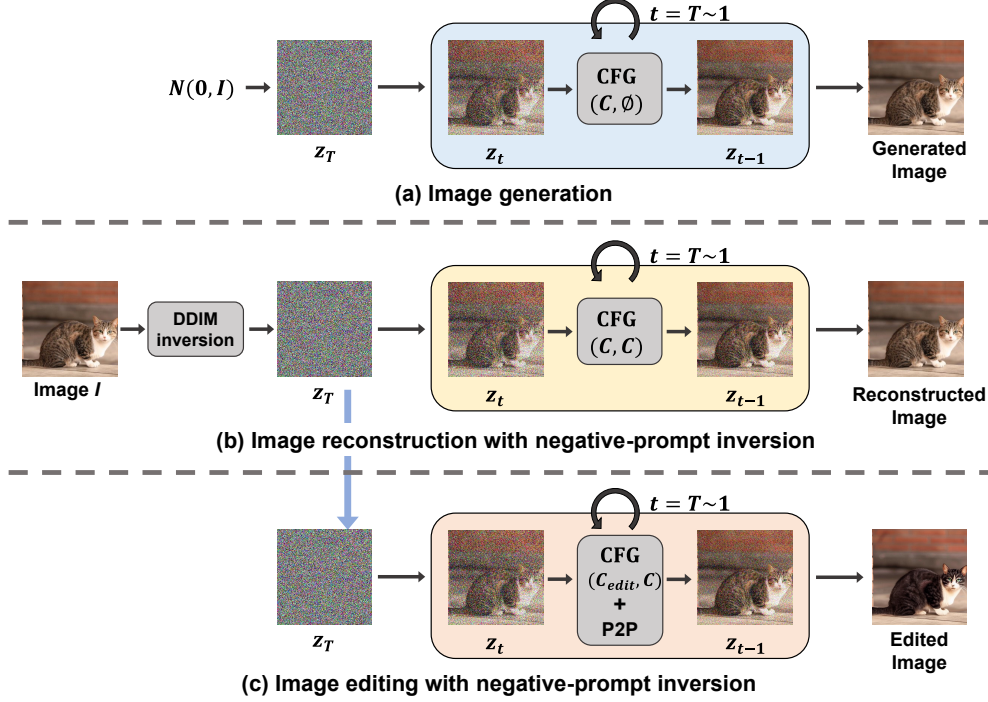


Figure 2. **Illustration of our framework.** (a) Image generation with CFG. A random noise z_T is sampled from a standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, then denoising z_t with CFG over diffusion steps from T to 1. $\text{CFG}(C, \emptyset)$ denotes that using a prompt embedding C for conditional prediction and the null-text embedding \emptyset for unconditional prediction. (b) Image reconstruction with negative-prompt inversion. We replace the null-text embedding \emptyset with the prompt embedding C in CFG. (c) Image editing with negative-prompt inversion. We use the edited prompt embedding C_{edit} as the text condition and use the original prompt embedding C instead of the null-text \emptyset in CFG with an image editing method such as prompt-to-prompt (P2P).

Then, we optimize \emptyset_t to minimize the MSE between the predicted $z_{t-1}(\bar{z}_t, t, C, \emptyset_t)$ and z_{t-1}^* :

$$\min_{\emptyset_t} \|z_{t-1}(\bar{z}_t, t, C, \emptyset_t) - z_{t-1}^*\|_2^2,$$

with the initialization $\emptyset_t = \emptyset_{t+1}$. After several updates (e.g., 10 iterations), we fix \emptyset_t and set $\bar{z}_{t-1} = z_{t-1}(\bar{z}_t, t, C, \emptyset_t)$. By performing the optimization at $t = T, \dots, 1$ sequentially, we can reconstruct the original image with high fidelity even when using CFG with $w > 1$. A downside of null-text inversion, on the other hand, is that the optimization of the null-text embedding \emptyset_t is time-consuming, as it should be performed at every diffusion step.

3.4. Negative-prompt inversion

The proposed method, **negative-prompt inversion**, utilizes the text prompt embeddings C instead of the optimized null-text embeddings $(\emptyset_t)_{t \in \{1, \dots, T\}}$ in null-text inversion. As a result, we can perform reconstruction with only forward computation without optimization, significantly reducing computation time.

We now discuss how one can avoid optimization in our

proposal, by more closely investigating the process of null-text inversion. Let us assume, for the following argument by induction, that at diffusion step t in null-text inversion one has \bar{z}_t that is close enough to z_t^* , so that one can regard $\bar{z}_t = z_t^*$ to hold. In null-text inversion, one obtains z_{t-1} from \bar{z}_t by moving one diffusion step backward using (4). Recall that z_t^* was calculated from z_{t-1}^* by moving one diffusion step forward in the diffusion process using (3):

$$z_t^* = \sqrt{\frac{\alpha_t}{\alpha_{t-1}}} z_{t-1}^* + \sqrt{\alpha_t} \left(\sqrt{\frac{1}{\alpha_t} - 1} - \sqrt{\frac{1}{\alpha_{t-1}} - 1} \right) \times \epsilon_{\theta}(z_{t-1}^*, t-1, C).$$

As we have assumed $\bar{z}_t = z_t^*$, one can substitute the above into (4), yielding

$$\bar{z}_{t-1} = z_{t-1}^* + \sqrt{\alpha_{t-1}} \left(\sqrt{\frac{1}{\alpha_{t-1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1} \right) \times (\tilde{\epsilon}_{\theta}(\bar{z}_t, t, C, \emptyset_t) - \epsilon_{\theta}(z_{t-1}^*, t-1, C)).$$

It implies that the discrepancy between \bar{z}_{t-1} and z_{t-1}^* in null-text inversion will be minimized when the predicted

velocity fields are equal:

$$\begin{aligned}\epsilon_{\theta}(\mathbf{z}_{t-1}^*, t-1, C) &= \tilde{\epsilon}_{\theta}(\bar{\mathbf{z}}_t, t, C, \varnothing_t) \\ &= w\epsilon_{\theta}(\bar{\mathbf{z}}_t, t, C) + (1-w)\epsilon_{\theta}(\bar{\mathbf{z}}_t, t, \varnothing_t)\end{aligned}$$

If furthermore we are allowed to assume that the predicted velocity fields at adjacent diffusion steps are equal, i.e., $\epsilon_{\theta}(\mathbf{z}_{t-1}^*, t-1, C) = \epsilon_{\theta}(\mathbf{z}_t^*, t, C) = \epsilon_{\theta}(\bar{\mathbf{z}}_t, t, C)$, then we can deduce that at the optimum the conditional and unconditional predictions are equal:

$$\epsilon_{\theta}(\bar{\mathbf{z}}_t, t, C) = \epsilon_{\theta}(\bar{\mathbf{z}}_t, t, \varnothing_t) \quad (5)$$

Of course one cannot expect the exact equality $\epsilon_{\theta}(\mathbf{z}_{t-1}^*, t-1, C) = \epsilon_{\theta}(\mathbf{z}_t^*, t, C)$ to hold, since the velocity field $\epsilon(\mathbf{z}, t, C)$ depends on \mathbf{z} and t . One can nevertheless expect that the equality holds approximately because of the continuity of the velocity field $\epsilon(\mathbf{z}, t, C)$ in (\mathbf{z}, t) . The optimized \varnothing_t can therefore be approximated by the prompt embedding C , so that we can discard the optimization of the null-text embedding \varnothing_t in null-text inversion altogether, simply by replacing the null-text embedding \varnothing_t with C . See Appendix A for more details on a theoretical justification and empirical validation in practical settings.

The argument so far has the following two consequences:

1. For reconstruction, letting $\varnothing_t = C$ amounts to not using CFG at all (since $\tilde{\epsilon}_{\theta}(\mathbf{z}_t, t, C, C) = \epsilon_{\theta}(\mathbf{z}_t, t, C)$ holds for any w). The above argument can thus be regarded as providing a justification to the empirically well-known observation that DDIM works well without CFG.
2. For editing, optimizing \varnothing_t in null-text inversion can be replaced by the simple substitution $\varnothing_t = C_{\text{src}}$ and $C = C_{\text{edit}}$ during the sampling process, where C_{src} and C_{edit} denote an embedding of a source prompt and an edited prompt, respectively.

Figure 2 illustrates our framework. (a) represents the image generation using CFG, while (b) represents our proposal, negative-prompt inversion, which replaces the null-text embedding with the input prompt embedding C . Additionally, in the case of image editing like prompt-to-prompt (P2P), we can set the embedding C_{edit} of an edited prompt as the text condition and set the original prompt embedding C as the negative-prompt embedding instead of the null-text embedding, as shown in Fig. 2 (c).

4. Experiments

4.1. Setting

In this section, we evaluate the proposed method qualitatively and quantitatively. We experimented it using Stable Diffusion v1.5 in Diffusers [31] implemented with PyTorch [21]. Our code used in the experiments is provided in

Supplementary Material. Following [19], we used 100 images and captions, randomly selected from validation data in COCO dataset [17], in our experiments. The images were trimmed to make them square and resized to 512×512 . Unless otherwise specified, in both DDIM inversion and sampling we set the number of the sampling steps to be 50 via using the stride of 20 over the $T = 1000$ diffusion steps.

We compared our method with DDIM inversion followed by DDIM sampling with CFG and null-text inversion, and evaluated their reconstruction quality with peak signal-to-noise ratio (PSNR) and learned perceptual image patch similarity (LPIPS) [33], whereas we evaluated their editing quality with CLIP score [22]. See Appendix B for our setting of null-text inversion. The inference speed was measured on one NVIDIA RTX A6000 connected to one AMD EPYC 7343 (16 cores, 3.2 GHz clockspeed).

4.2. Reconstruction

The left three columns of Table 1 shows PSNR, LPIPS, and inference time of reconstruction by the three methods compared. In terms of PSNR (higher is better) and LPIPS (lower is better), the reconstruction quality of the proposed method was slightly worse than that of null-text inversion but far better than that of DDIM inversion. On the other hand, the inference speed was 30 times as fast as that of null-text inversion. This remarkable acceleration is achieved since the iterative optimization and backpropagation processing required for null-text inversion are not necessary for our method.

In Fig. 3, the left four columns display examples of reconstruction by the three methods. DDIM inversion reconstructed images with noticeable differences from the input images, such as object position and shape. In contrast, null-text inversion and negative-prompt inversion (Ours) were capable of reconstructing images, with results that were nearly identical to the input images, and the proposed method achieved a high reconstruction quality comparable to that of null-text inversion. See Appendix C.1 for additional reconstruction examples. These results suggest that the proposed method can achieve reconstruction quality nearly equivalent to null-text inversion, with a speedup of over 30 times. Additionally, we also measured the memory usage of the three methods, and found that our method and DDIM inversion used approximately half as much memory as null-text inversion.

4.3. Editing

We next demonstrate the feasibility of editing real images by combining our inversion method with existing image editing methods. Our method is independent of the image editing approach and is principally compatible with any method that uses CFG, allowing for the selection of an appropriate image editing method depending on the objec-

Table 1. **Evaluation of reconstruction/editing quality and speed in each method.** \pm represents 95% confidence intervals. Note that as DDIM inversion and ours perform the same process, they are theoretically at the same speed.

Method	PSNR \uparrow	LPIPS \downarrow	Speed (s)	CLIP \uparrow
Imagic	17.17 ± 0.66	0.356 ± 0.025	552.86 ± 0.16	22.99 ± 0.77
DDIM inversion	14.05 ± 0.34	0.528 ± 0.022	4.61 ± 0.03	25.10 ± 0.74
Null-text inversion	26.11 ± 0.81	0.075 ± 0.007	129.77 ± 2.97	24.07 ± 0.72
Ours	23.38 ± 0.66	0.160 ± 0.016	4.63 ± 0.02	23.77 ± 0.74

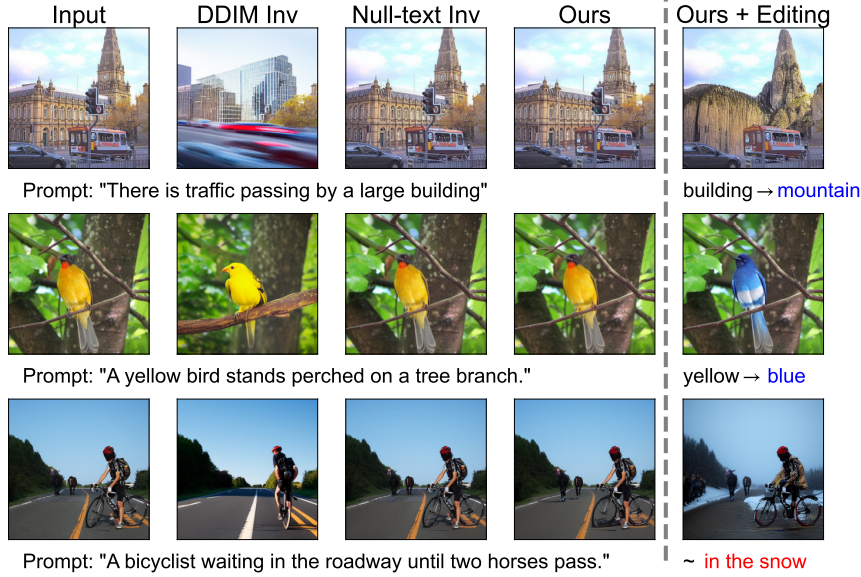


Figure 3. **Evaluation of reconstructed images.** The left 4 columns show the reconstruction results of each method, and the right column shows the image editing results using our method and prompt-to-prompt. The editing prompts are described below the edited images, that were created by replacing words or adding new words to the original prompt. Our method reconstructed input images as well as null-text inversion and edited images also preserved the structure of the input images.

tive. Here, we verify the effectiveness of our method for real-image editing using prompt-to-prompt [10] in the same manner as in [19].

The rightmost column of Table 1 shows CLIP scores of editing results by prompt-to-prompt with the three methods compared. Taking account of the standard errors, one can see that the proposed method and null-text inversion achieved almost the same CLIP scores. Although the score of DDIM inversion was the best, by considering the scores in conjunction with reconstruction quality, the editing quality of the proposed method was comparable to that of null-text inversion. In addition, we also compared our method with Imagic [15] as another editing method. The editing quality of the proposed method was also better than that of Imagic. For qualitative evaluation, the rightmost column of Fig. 3 shows examples of real-image editing via prompt-to-prompt using the proposed method. The proposed method managed to maintain the composition while editing the image according to the modified prompt, such as replacing the

objects and changing the background. Additional editing examples are provided in Appendices C.2 and C.3. These observations show that our inversion method can be combined with editing methods like prompt-to-prompt to enable ultrafast real-image editing.

4.4. Number of sampling steps

As the proposed method allows ultrafast reconstruction/editing, one may be able to use a larger number of sampling steps to further improve reconstruction quality, at the expense of reduced speed. To investigate the relationship between the number of sampling steps and reconstruction quality, we measured the PSNR and LPIPS using five different sampling steps: 20, 50, 100, 200, and 500.

Figure 4 shows PSNR, LPIPS, and speed versus the number of sampling steps by the three methods. Although results with high enough quality were obtained with 50 sampling steps, increasing the number of sampling steps further improved the reconstruction quality of the proposed

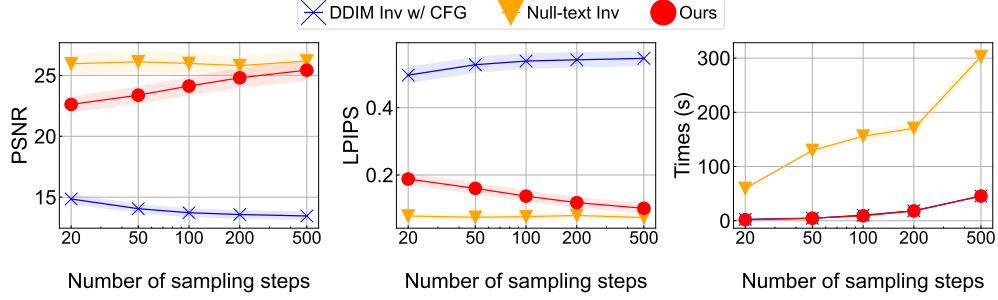


Figure 4. **Reconstruction quality and speed versus the number of sampling steps.** Higher PSNR is better (left), lower LPIPS is better (middle), and shorter execution time is better (right). Shadings indicate 95% confidence intervals.



Figure 5. **Reconstructed images when changing the number of sampling steps.** The images became more similar to the input images as the number of sampling steps increased.

method, approaching that of null-text inversion. It should be noted that the total execution time is roughly given by the product of the execution time per sampling step and the number of sampling steps, so that even if the proposed inversion method is performed with 500 sampling steps, it would still take less time than executing null-text inversion with 50 sampling steps thanks to the $30\times$ speedup. In fact, Fig. 4 right shows the time taken for inversion; with 500 sampling steps, it took 46 seconds, which is still approximately three times faster than the null-text inversion with 50 sampling steps, which took 130 seconds. We would like to note that in Fig. 4 right the execution time of null-text inversion was not proportional to the number of sampling steps, since in our experimental setting the early stopping employed in the null-text optimization was more effective as the number of sampling steps became larger.

Figure 5 describes how the reconstructed image changed as the number of sampling steps was increased. Even with a small number of sampling steps, such as 20, the input image’s objects and composition were successfully reconstructed. Focusing on the finer details, for example, the head of the bed and the desk in the first row, and the wall color and pipes on the wall in the second row, we observe

that the reconstruction quality improved as the number of sampling steps was increased. This improvement is generally imperceptible at first glance, suggesting that conventionally adopted numbers of sampling steps, such as 20 and 50 sampling steps, yield sufficiently satisfactory reconstruction results for practical purposes.

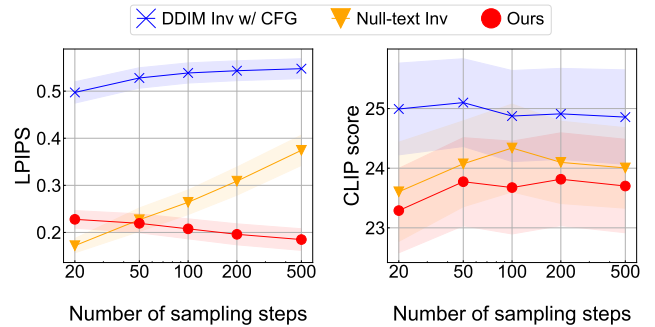


Figure 6. **Editing quality versus the number of sampling steps.** lower LPIPS is better (left), and higher CLIP scores is better (right). Shadings indicate 95% confidence intervals.

To evaluate image editing quality against sampling steps,

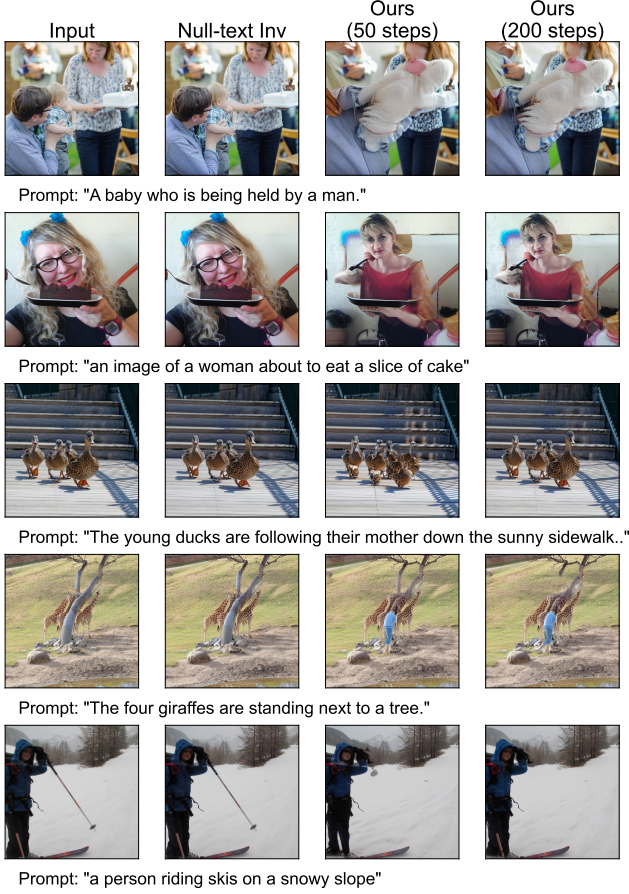


Figure 7. Additional failure cases of our method.

we measured LPIPS and CLIP scores. We calculated LPIPS between the edited images and their original counterparts, and CLIP scores between the edited images and the corresponding editing prompts. These measures are essential for evaluation, as image editing quality can be assessed by how well it preserves the original image structure and how faithfully it adheres to the editing prompt. Figure 6 illustrates LPIPS and CLIP scores as a function of the number of sampling steps for the three methods. In terms of LPIPS, the proposed method better preserved image structure compared with null-text inversion when the number of sampling steps exceeded 50. Regarding CLIP scores, our method achieved comparable results to null-text inversion, considering the confidence interval. Although DDIM inversion achieved the highest CLIP score, its overall editing quality was inferior, as evidenced by its poorer LPIPS results. Considering both measures, the proposed method demonstrated superior image editing quality compared with null-text inversion when the number of sampling steps exceeded 50.

5. Limitations

A limitation of the proposed method is that the average reconstruction quality does not reach that of null-text inversion. As demonstrated in the previous section, the difference is generally imperceptible at first glance; however, there were instances where our inversion method failed significantly.

Figure 7 shows failure cases of our method. In all the cases shown, our method failed to reconstruct the images in 50 sampling steps, whereas null-text inversion successfully reconstructed them. The first two rows show failures due to the disappearance of people, where the objects were either reconstructed as non-human or as different persons. The third and fourth rows show failures due to the color gradient being reconstructed as separate objects, such as a single duck being reconstructed as scattered pieces, and a tree trunk being reconstructed as a different object. The last row shows a failure due to the disappearance of a tiny object, where one of the ski poles was missing. The failures of reconstruction of humans could be attributed to characteristics of Stable Diffusion’s AutoEncoder. In such cases, employing a more effective encoder-decoder pair may result in improvements. Moreover, as can be observed in the duck example, the reconstruction quality can be improved by increasing the number of sampling steps.

Although failures in post-reconstruction image editing may occur, our inversion method is independent of editing methods, making the related discussion beyond the scope of this paper.

6. Conclusions

We have proposed negative-prompt inversion, which enables real-image inversion in diffusion models without the need for optimization. Experimentally, it produced visually high-fidelity reconstruction results comparable to inversion methods requiring optimization, while achieving a remarkable speed-up of over 30 times. Furthermore, we discovered that increasing the number of sampling steps further improved the reconstruction quality while maintaining faster computational time than existing methods.

On the basis of these results, our method provides a practical approach for real-image reconstruction. This utility excels in high-computational-cost scenarios, such as video editing, where our method proves to be even more beneficial. Moreover, by parallelizing multiple GPUs and optimizing the program, there is potential for our method to achieve higher throughput and lower latency, where even the real-time processing would be possible. Although the proposed approach reduces computational costs and is available to any user, it does not encourage socially inappropriate use.

References

- [1] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22563–22575, 2023. arXiv:2304.08818v1 [cs.CV]. 1
- [2] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, and William Chan. WaveGrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations*, 2021. arXiv:2009.00713v2 [eess.AS]. 1
- [3] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, Najim Dehak, and William Chan. WaveGrad 2: Iterative refinement for text-to-speech synthesis. In *Proceedings of Interspeech 2021*, pages 3765–3769, 2021. arXiv:2106.09660v2 [eess.AS]. 1
- [4] Hansam Cho, Jonghyun Lee, Seoung Bum Kim, Tae-Hyun Oh, and Yonghyun Jeong. Noise map guidance: Inversion with spatial context for real image editing. In *International Conference on Learning Representations*, 2023. arXiv:2402.04625v1 [cs.CV]. 2
- [5] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021. arXiv:2105.05233v4 [cs.LG]. 1, 2, 3
- [6] 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. In *Proceedings of the 11th International Conference on Learning Representations*, 2023. 2
- [7] Rinon Gal, Moab Arar, Yuval Atzmon, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. Encoder-based domain tuning for fast personalization of text-to-image models. *ACM Transactions on Graphics*, 42(4):150 (13 pages), 2023. arXiv:2302.12228v3 [cs.CV]. 2
- [8] Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. ReNoise: Real image inversion through iterative noising. arXiv:2403.14602v1 [cs.CV], 2024. 2
- [9] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weibach, and Frank Wood. Flexible diffusion modeling of long videos. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27953–27965. Curran Associates, Inc., 2022. arXiv:2205.11495v3 [cs.CV]. 1
- [10] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-Prompt image editing with cross attention control. In *Proceedings of the 11th International Conference on Learning Representations*, 2023. arXiv:2208.01626v1 [cs.CV]. 1, 2, 6
- [11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020. arXiv:2006.11239v2 [cs.LG]. 1, 3, 11
- [12] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. arXiv:2207.12598v1 [cs.LG]. 1
- [13] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 8633–8646. Curran Associates, Inc., 2022. arXiv:2204.03458v2 [cs.CV]. 1
- [14] Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling. *Transactions on Machine Learning Research*, Nov. 2022. arXiv:2206.07696v3 [cs.CV]. 1
- [15] Bahjat Kavar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6007–6017, 2023. arXiv:2210.09276v3 [cs.CV]. 2, 6
- [16] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. DiffusionCLIP: Text-guided diffusion models for robust image manipulation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2426–2435, 2022. arXiv:2110.02711v6 [cs.CV]. 1
- [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 David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. arXiv:1405.0312v3 [cs.CV]. 5
- [18] 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 *International Conference on Learning Representations*, 2022. arXiv:2108.01073v2 [cs.CV]. 15
- [19] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6038–6047, 2023. GitHub: <https://null-text-inversion.github.io/>. 1, 2, 5, 6, 14
- [20] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. arXiv:2302.03027v1 [cs.CV]. 3
- [21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmai-

- son, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems* 32, pages 8024–8035. Curran Associates, Inc., 2019. arXiv:1912.01703v1 [cs.LG]. 5
- [22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Clueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 8748–8763. PMLR, 2021. arXiv:2103.00020v1 [cs.CV]. 3, 5
- [23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022. arXiv:2112.10752v2 [cs.CV]. 1, 2, 3
- [24] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22500–22510, 2023. arXiv:2208.12242v2 [cs.CV]. 2
- [25] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 36479–36494. Curran Associates, Inc., 2022. 2205.11487v1 [cs.CV]. 1, 2
- [26] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. arXiv:2010.02502v4 [cs.LG]. 1, 2, 3, 11
- [27] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, 2019. arXiv:1907.05600v3 [cs.LG]. 1
- [28] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. arXiv:2011.13456v2 [cs.LG]. 1
- [29] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1921–1930, 2023. arXiv:2211.12572v1 [cs.CV]. 2, 15
- [30] Dani Valevski, Matan Kalman, Yossi Matias, and Yaniv Leviathan. UniTune: Text-driven image editing by fine tuning an image generation model on a single image. *ACM Transactions on Graphics*, 42(4):128 (10 pages), 2023. arXiv:2210.09477v3 [cs.CV]. 2
- [31] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. <https://github.com/huggingface/diffusers>, 2022. 5
- [32] Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. ELITE: Encoding visual concepts into textual embeddings for customized text-to-image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. arXiv:2302.13848v1 [cs.CV]. 2
- [33] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. arXiv:1801.03924v2 [cs.CV]. 5
- [34] Zhixing Zhang, Ligong Han, Arnab Ghosh, Dimitris Metaxas, and Jian Ren. SINE: SINGLE image Editing with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6027–6037, 2023. arXiv:2212.04489v1 [cs.CV]. 2

Supplementary Material: Negative-prompt Inversion: Fast Image Inversion for Editing with Text-guided Diffusion Models

A. Justifying arguments

A.1. Theoretical consideration

In this appendix, we firstly provide a continuous-time description of DDPM and DDIM processes. We start with the stochastic differential equation describing continuous-time random diffusion of particles in a D -dimensional space:

$$dz = -\gamma_t z + \sqrt{2\gamma_t} dW, \quad (6)$$

where W is the D -dimensional Wiener process and where the time-dependent decay parameter $\gamma_t > 0$ is a deterministic and integrable function of t . If one lets γ_t to be independent of t , then (6) describes what is called the Ornstein-Uhlenbeck (OU) process, so that (6) can be regarded as a generalized version of the OU process. The distribution $p_t(z)$ of the random particles following the diffusion process (6) at time t is known to follow the Fokker-Planck equation

$$\frac{\partial p_t}{\partial t} = \gamma_t \{ \nabla(z p_t) + \Delta p_t \}. \quad (7)$$

The solution of (7) given the initial condition $p_0(z) = \delta(z|_{t=0} - z_0)$, i.e., all the random particles are located at the position z_0 at time 0, or equivalently, one starts the diffusion process with a sample located at z_0 , is evaluated as

$$p_t(z | z|_{t=0} = z_0) = \mathcal{N}(\sqrt{\alpha_t} z_0, (1 - \alpha_t) \mathbf{I}), \quad (8)$$

where

$$\alpha_t := \exp\left(-\int_0^t \gamma_s ds\right). \quad (9)$$

We also write it as

$$z_t|_{z_0} \sim \mathcal{N}(\sqrt{\alpha_t} z_0, (1 - \alpha_t) \mathbf{I}). \quad (10)$$

Furthermore, for $s \leq t$, the conditional distribution of the particles at time t conditional on the particle located at z_s at time s is given by

$$z_t|_{z_s} \sim \mathcal{N}\left(\sqrt{\frac{\alpha_t}{\alpha_s}} z_s, \left(1 - \frac{\alpha_t}{\alpha_s}\right) \mathbf{I}\right). \quad (11)$$

Comparing these formulas with those in [11, Section 2] reveals that discretizing the above process in time will give us the formulation of DDPM.

Assuming that the Fokker-Planck equation (7) is given, the corresponding random process is not unique, and there

are several other random processes which are consistent with (7) than the above generalized OU process (6). For example, we may take a specific time instant $t = T > 0$ and require the particle position z_T at time T given the initial position z_0 at time $t = 0$ to follow the Gaussian distribution

$$z_T|_{z_0} \sim \mathcal{N}(\sqrt{\alpha_T} z_0, (1 - \alpha_T) \mathbf{I}), \quad (12)$$

and then determine the particle position z_t at any time $t \geq 0$ as

$$z_t = \sqrt{\frac{1 - \alpha_t}{1 - \alpha_T}} z_T + \left(\sqrt{\alpha_t} - \sqrt{\frac{\alpha_T}{1 - \alpha_T}} \sqrt{1 - \alpha_t} \right) z_0. \quad (13)$$

One can then confirm that the conditional distribution of the particle position z_t at time t conditional on z_0 is given by $\mathcal{N}(\sqrt{\alpha_t} z_0, (1 - \alpha_t) \mathbf{I})$, which demonstrates that the distribution of the particles following the above process also satisfies the same Fokker-Planck equation (7). One can furthermore show that discretizing the above process in time will give us the formulation of DDIM [26].

When considering z_t given z_0 and z_T , let $d_t := (z_t - \sqrt{\alpha_t} z_0) / \sqrt{1 - \alpha_t}$ be the normalized noise component in z_t relative to $\sqrt{\alpha_t} z_0$. One can show, by rearranging terms in (13), that $d_t = d_T$ holds for any t . Letting $d := d_t$ due to the independence of d_t on t , one can furthermore show, via (12), that d given z_0 follows the standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. In other words, given z_0 and z_T , the normalized noise component d_t in DDIM does not depend on t . Therefore, the diffusion paths in DDIM are straight half-lines $\{z_t = \sqrt{\alpha_t} z_0 + \sqrt{1 - \alpha_t} d : t \geq 0, d \sim \mathcal{N}(\mathbf{0}, \mathbf{I})\}$ starting from z_0 with random velocity $d \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Assuming that z_t is available, the model $\epsilon_\theta(z_t, t)$ attempts to estimate the velocity d_t from z_t , which in turn yields an estimate $f_\theta^{(t)}(z_t) := (z_t - \sqrt{1 - \alpha_t} \epsilon_\theta(z_t, t)) / \sqrt{\alpha_t}$ of z_0 , and then one can use it to estimate z_s for any s by plugging it into the equality $d_t = d_s$. Specifically, z_s is estimated via

$$\begin{aligned} z_s &= \sqrt{\alpha_s} z_0 + \sqrt{1 - \alpha_s} \frac{z_t - \sqrt{\alpha_t} z_0}{\sqrt{1 - \alpha_t}} \\ &\approx \sqrt{\alpha_s} \left(\frac{z_t - \sqrt{1 - \alpha_t} \epsilon_\theta(z_t, t)}{\sqrt{\alpha_t}} \right) + \sqrt{1 - \alpha_s} \epsilon_\theta(z_t, t) \\ &= \sqrt{\frac{\alpha_s}{\alpha_t}} z_t + \sqrt{\alpha_s} \left(\sqrt{\frac{1 - \alpha_s}{\alpha_s}} - \sqrt{\frac{1 - \alpha_t}{\alpha_t}} \right) \epsilon_\theta(z_t, t). \end{aligned} \quad (14)$$

When one takes $s = t \pm 1$, the above formula is reduced to

$$z_{t\pm 1} \approx \sqrt{\frac{\alpha_{t\pm 1}}{\alpha_t}} z_t + \sqrt{\alpha_{t\pm 1}} \left(\sqrt{\frac{1}{\alpha_{t\pm 1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1} \right) \times \epsilon_\theta(z_t, t), \quad (15)$$

which corresponds to (3) and (2) in the main text.

The argument presented so far is based on conditioning on sample z_0 , which is not justifiable in the actual process of DDIM sampling where there exists more than one sample and where the model does not look at z_0 . We thus extend the above argument via assuming z_0 to be generated according to a certain probability distribution $p(z_0)$. More concretely, we assume $z_0 \sim p(z_0)$ and $\mathbf{d} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, which induces the diffusion path $z_t = \sqrt{\alpha_t} z_0 + \sqrt{1 - \alpha_t} \mathbf{d}$, $t \geq 0$, in DDIM according to the above discussion. Consequently, at position z and at time t , the “velocity field” $\epsilon(z, t)$ to be learned by the model $\epsilon_\theta(z, t)$ is not determined by a single sample z_0 but given by the posterior mean of $\mathbf{d} = (z - \sqrt{\alpha_t} z_0) / \sqrt{1 - \alpha_t}$ with respect to the posterior distribution of z_0 given z , which is obtained from the prior distributions $z_0 \sim p(z_0)$ and $\mathbf{d} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, as well as the likelihood $p(z | z_0, \mathbf{d}) = \delta(z - \sqrt{\alpha_t} z_0 - \sqrt{1 - \alpha_t} \mathbf{d})$.

Proposition 1. Assume $z_0 \sim p(z_0)$ and $\mathbf{d} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Then the velocity field $\epsilon(z, t)$ in DDIM at position z and at time t , which is to be learned by the model $\epsilon_\theta(z, t)$, is given by

$$\epsilon(z, t) = \frac{\left\langle \frac{z - \sqrt{\alpha_t} z_0}{\sqrt{1 - \alpha_t}} p_G \left(\frac{z - \sqrt{\alpha_t} z_0}{\sqrt{1 - \alpha_t}} \right) \right\rangle_{z_0}}{\left\langle p_G \left(\frac{z - \sqrt{\alpha_t} z_0}{\sqrt{1 - \alpha_t}} \right) \right\rangle_{z_0}}, \quad (16)$$

where

$$p_G(\mathbf{d}) = \frac{1}{(2\pi)^{D/2}} e^{-\|\mathbf{d}\|_2^2/2} \quad (17)$$

denotes the probability density function of the D -dimensional standard Gaussian distribution, and where $\langle \cdot \rangle_{z_0}$ denotes expectation with respect to $z_0 \sim p(z_0)$.

Proof. The joint distribution of z_0 and z is given by

$$\begin{aligned} p(z_0, z) &= \int p(z | z_0, \mathbf{d}) p(z_0) p_G(\mathbf{d}) d\mathbf{d} \\ &= \int \delta(z - \sqrt{\alpha_t} z_0 - \sqrt{1 - \alpha_t} \mathbf{d}) p(z_0) p_G(\mathbf{d}) d\mathbf{d} \\ &= p_G \left(\frac{z - \sqrt{\alpha_t} z_0}{\sqrt{1 - \alpha_t}} \right) p(z_0), \end{aligned} \quad (18)$$

from which the posterior distribution of z_0 given z is obtained as

$$p(z_0 | z) = \frac{p_G \left(\frac{z - \sqrt{\alpha_t} z_0}{\sqrt{1 - \alpha_t}} \right) p(z_0)}{\left\langle p_G \left(\frac{z - \sqrt{\alpha_t} z_0}{\sqrt{1 - \alpha_t}} \right) \right\rangle_{z_0}}, \quad (19)$$

The velocity $\epsilon(z, t)$ at z and t , to be learned by the model, is given by the posterior mean of $\mathbf{d} = (z - \sqrt{\alpha_t} z_0) / \sqrt{1 - \alpha_t}$, which is represented as (16), proving the proposition. \square

It should be noted that the velocity field $\epsilon(z, t)$ is deterministic: Equation (16) shows that although it depends on the prior distribution $p(z_0)$ and γ_t via α_t as in (9) it is a non-random quantity. Despite its complex appearance, one can see that the velocity field $\epsilon(z, t)$ in (16) is continuous, and even continuously differentiable, in z and $t > 0$. This continuity implies that, for $t, s > 0$, when $|\alpha_t - \alpha_s|$ and $\|z - z'\|$ are small, one can expect $\epsilon(z, t) \approx \epsilon(z', s)$ to hold.

In what follows, we provide a justifying argument for the proposed method, via extending the argument so far by incorporating conditioning into the model. It is straightforward to incorporate conditioning in the DDIM inversion and sampling formulae (15), by replacing the model $\epsilon_\theta(z_t, t)$ without conditioning with the conditional model $\epsilon_\theta(z_t, t, C)$, as shown in (3) and (2) in the main text, where C is the prompt embedding. In various applications, on the other hand, the reverse process using the DDIM sampling formula (2) is often combined with CFG to strengthen the effects of the conditioning, where the conditional model $\epsilon_\theta(z_t, t, C)$ is further replaced with

$$\begin{aligned} \tilde{\epsilon}_\theta(z_t, t, C, \emptyset) &= \epsilon_\theta(z_t, t, \emptyset) \\ &\quad + w (\epsilon_\theta(z_t, t, C) - \epsilon_\theta(z_t, t, \emptyset)), \end{aligned} \quad (20)$$

where $w \geq 0$ is the guidance scale, which controls the strength of the conditioning, and where \emptyset is the null-text embedding.

The first step of null-text inversion is to obtain z_t^* for $t = 1, \dots, T$ by initializing $z_0^* = z_0$ and successively applying the forward process derived as the DDIM inversion formula

$$\begin{aligned} z_t^* &= \sqrt{\frac{\alpha_t}{\alpha_{t-1}}} z_{t-1}^* + \sqrt{\alpha_t} \left(\sqrt{\frac{1}{\alpha_t} - 1} - \sqrt{\frac{1}{\alpha_{t-1}} - 1} \right) \\ &\quad \times \epsilon_\theta(z_{t-1}^*, t-1, C), \end{aligned} \quad (21)$$

which is the same as (3) in the main text. Next, starting from $\bar{z}_T = z_T^*$, we calculate the reverse diffusion process to obtain \bar{z}_t in the backward direction, while optimizing the null-text embedding \emptyset_t at each diffusion step so that \bar{z}_t well reproduces z_t^* . More specifically, for $t = T, T-1, \dots, 1$, \bar{z}_{t-1} is calculated via combining the DDIM sampling (15) and CFG (20) as

$$\begin{aligned} &z_{t-1}(\bar{z}_t, t, C, \emptyset_t) \\ &= \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} \bar{z}_t + \sqrt{\alpha_{t-1}} \left(\sqrt{\frac{1}{\alpha_{t-1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1} \right) \\ &\quad \times \tilde{\epsilon}_\theta(\bar{z}_t, t, C, \emptyset_t), \end{aligned} \quad (22)$$

which is the same as (4) in the main text.

The null-text embedding \varnothing_t is optimized to minimize the MSE between $\mathbf{z}_{t-1}(\bar{\mathbf{z}}_t, t, C, \varnothing_t)$ and \mathbf{z}_{t-1}^* as

$$\min_{\varnothing_t} \|\mathbf{z}_{t-1}(\bar{\mathbf{z}}_t, t, C, \varnothing_t) - \mathbf{z}_{t-1}^*\|_2^2. \quad (23)$$

The following proposition shows that the choice $\varnothing_t = C$ does minimize the MSE between $\mathbf{z}_{t-1}(\bar{\mathbf{z}}_t, t, C, \varnothing_t)$ and \mathbf{z}_{t-1}^* under an ideal situation.

Proposition 2. *Assume that there is only one sample, and that the guidance scale w in CFG is not equal to 1. For any t , if the model $\epsilon(\mathbf{z}, t, C)$ is able to correctly predict the velocity field and if $\mathbf{z}_t^* = \bar{\mathbf{z}}_t$ holds true, then the difference between $\mathbf{z}_{t-1}(\bar{\mathbf{z}}_t, t, C, \varnothing_t)$ and \mathbf{z}_{t-1}^* in null-text inversion is made equal to zero if and only if $\epsilon_\theta(\bar{\mathbf{z}}_t, t, \varnothing_t)$ is equal to $\epsilon_\theta(\bar{\mathbf{z}}_t, t, C)$.*

Proof. The difference between $\mathbf{z}_{t-1}(\bar{\mathbf{z}}_t, t, C, \varnothing_t)$ and \mathbf{z}_{t-1}^* is expressed as

$$\begin{aligned} & \mathbf{z}_{t-1}(\bar{\mathbf{z}}_t, t, C, \varnothing_t) - \mathbf{z}_{t-1}^* \\ &= \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} \bar{\mathbf{z}}_t + \sqrt{\alpha_{t-1}} \left(\sqrt{\frac{1}{\alpha_{t-1}}} - 1 - \sqrt{\frac{1}{\alpha_t}} - 1 \right) \\ & \quad \times \tilde{\epsilon}_\theta(\bar{\mathbf{z}}_t, t, C, \varnothing_t) - \mathbf{z}_{t-1}^* \\ &= \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} \mathbf{z}_t^* + \sqrt{\alpha_{t-1}} \left(\sqrt{\frac{1}{\alpha_{t-1}}} - 1 - \sqrt{\frac{1}{\alpha_t}} - 1 \right) \\ & \quad \times \tilde{\epsilon}_\theta(\bar{\mathbf{z}}_t, t, C, \varnothing_t) - \mathbf{z}_{t-1}^* \\ &= \sqrt{\alpha_{t-1}} \left(\sqrt{\frac{1}{\alpha_{t-1}}} - 1 - \sqrt{\frac{1}{\alpha_t}} - 1 \right) \\ & \quad \times (\tilde{\epsilon}_\theta(\bar{\mathbf{z}}_t, t, C, \varnothing_t) - \epsilon_\theta(\mathbf{z}_{t-1}^*, t-1, C)). \end{aligned} \quad (24)$$

In the second line of the above equation we used the assumption $\mathbf{z}_t^* = \bar{\mathbf{z}}_t$, and in the third line we substituted (21) into \mathbf{z}_t^* above.

As described above, the model $\epsilon_\theta(\mathbf{z}_t, t, C)$ attempts to estimate noise \mathbf{d}_t from \mathbf{z}_t , and the assumption that the model correctly predicts the velocity, together with the discussion at the beginning of this section, implies that $\epsilon_\theta(\mathbf{z}_t^*, t, C) = \epsilon_\theta(\mathbf{z}_{t-1}^*, t-1, C)$ should hold. One therefore has

$$\begin{aligned} & \epsilon_\theta(\mathbf{z}_{t-1}^*, t-1, C) - \tilde{\epsilon}_\theta(\bar{\mathbf{z}}_t, t, C, \varnothing_t) \\ &= \epsilon_\theta(\mathbf{z}_t^*, t, C) - \tilde{\epsilon}_\theta(\bar{\mathbf{z}}_t, t, C, \varnothing_t) \\ &= \epsilon_\theta(\bar{\mathbf{z}}_t, t, C) - \tilde{\epsilon}_\theta(\bar{\mathbf{z}}_t, t, C, \varnothing_t) \\ &= (1-w)(\epsilon_\theta(\bar{\mathbf{z}}_t, t, C) - \epsilon_\theta(\bar{\mathbf{z}}_t, t, \varnothing_t)). \end{aligned} \quad (25)$$

As we have assumed $w \neq 1$, $\mathbf{z}_{t-1}^* - \mathbf{z}_{t-1}(\bar{\mathbf{z}}_t, t, C, \varnothing_t)$ is proportional to $\epsilon_\theta(\bar{\mathbf{z}}_t, t, C) - \epsilon_\theta(\bar{\mathbf{z}}_t, t, \varnothing_t)$, and it is made equal to zero if and only if $\epsilon_\theta(\bar{\mathbf{z}}_t, t, C)$ and $\epsilon_\theta(\bar{\mathbf{z}}_t, t, \varnothing_t)$ are equal. \square

Since we initialize $\bar{\mathbf{z}}_T = \mathbf{z}_T^*$ at diffusion step T , recursive application of Proposition 2 shows, under the ideal situation that the model has learned perfectly, that one will have $\bar{\mathbf{z}}_t = \mathbf{z}_t^*$ for all t via letting $\varnothing_t = C$. In other words, one can regard that null-text inversion optimizes the unconditional prediction to approach the conditional prediction at each diffusion step.

Under practical situations, one can no longer expect the exact equality $\epsilon_\theta(\mathbf{z}_t^*, t, C) = \epsilon_\theta(\mathbf{z}_{t-1}^*, t-1, C)$ to hold. One can still expect, however, that the above equality approximately holds: One typically takes small timesteps so that $\alpha_{t-1} \approx \alpha_t$ and $\mathbf{z}_{t-1}^* \approx \mathbf{z}_t^*$, so that the argument given after Proposition 1 assures that the above equality holds approximately.

A.2. Empirical evaluations

The assumption of perfect learning of the model adopted in Proposition 2 in the previous section is certainly too strong to be applied to practical situations. We have already discussed the issue of conditioning on \mathbf{z}_0 in the previous section. Another reason is that it is almost always the case that the model learns only approximately. Accordingly, what one can expect in practice would be that $\bar{\mathbf{z}}_t = \mathbf{z}_t^*$ holds only approximately, which would then make the validity of the optimality of $\varnothing_t = C$ in null-text inversion rather questionable. In this section, we investigate empirically how good the prompt-text embedding C is compared with the optimized null-text embedding \varnothing_t , in terms of the velocity prediction by the model, as well as their representation in the embedding space. In the experiments in this section, we used the same 100 image-prompt pairs from the COCO dataset as those used in the experiments in the main text.

We first investigated how close the velocity prediction $\epsilon_\theta(\mathbf{z}_t, t, \varnothing_t)$ using the optimized null-text embedding \varnothing_t and the prediction $\epsilon_\theta(\mathbf{z}_t, t, C)$ using the prompt embedding C are. More specifically, we performed null-text inversion, starting from \mathbf{z}_T^* obtained via DDIM inversion using the embedding C , and with the resulting sequences $(\bar{\mathbf{z}}_t)_{t \in \{1, \dots, T\}}$ and $(\varnothing_t)_{t \in \{1, \dots, T\}}$ we evaluated the L_1 distance between $\epsilon_\theta(\bar{\mathbf{z}}_t, t, \varnothing_t)$ and $\epsilon_\theta(\bar{\mathbf{z}}_t, t, C)$. For comparison, we also calculated the L_1 distance between $\epsilon_\theta(\bar{\mathbf{z}}_t, t, \varnothing_t)$ and the velocity prediction $\epsilon_\theta(\bar{\mathbf{z}}_t, t, C')$ obtained using the embeddings C' of the prompts associated with images other than the target image, as well as the L_1 distance between $\epsilon_\theta(\bar{\mathbf{z}}_t, t, C)$ and $\epsilon_\theta(\bar{\mathbf{z}}_t, t, C')$.

Figure 8 left shows the mean L_1 distance of the predicted velocities. The predicted velocities using the optimized embeddings $(\varnothing_t)_{t \in \{1, \dots, T\}}$ were closer to those using C than those using C' , with a smaller distance than the distance between the predicted velocity using C and that using C' . One observes that the distance between the velocity predictions using \varnothing_t and C became larger as t became smaller,

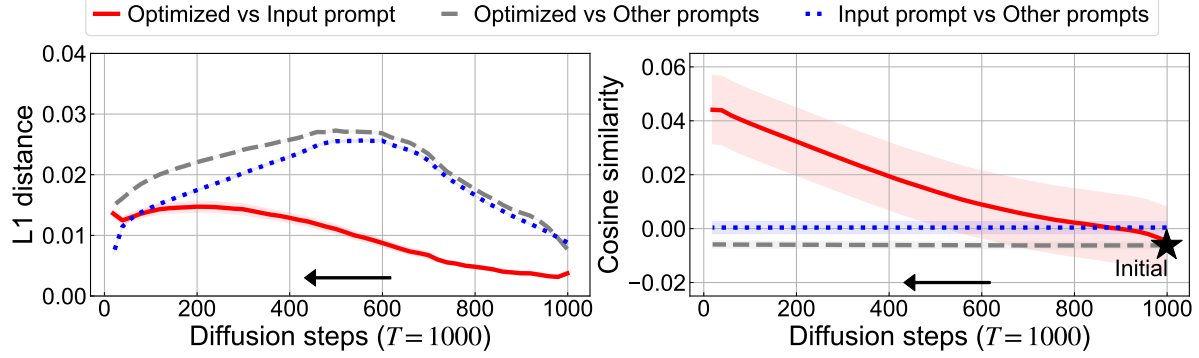


Figure 8. **Similarity between the optimized null-text and the input prompt.** (Left) The mean L_1 distance between predicted velocities using the optimized null-text embedding and the input prompt. (Right) The mean similarity between the optimized null-text embedding and the input prompt. The **solid red** line shows the similarity between the optimized null-text and the input prompt. The **dashed line** shows the similarity between the optimized null-text and the other prompts. The **dotted line** shows the similarity between the input prompt and the other prompts. **Initial** represents the starting point of optimization, and optimization was performed in the order indicated by the direction of the arrow in 50 sampling steps. Shaded regions indicate 95% confidence intervals.

which would be ascribed to the accumulation of optimization errors. One can also notice that the distance between the velocity predictions using $(\varnothing_t)_{t \in \{1, \dots, T\}}$ and C were larger than that between those using C and C' near $t = 0$. Velocity predictions near $t = 0$, however, would have almost no impact on generated samples since they are added at very small scales. The results suggest that the predicted velocity $\epsilon_\theta(\bar{z}_t, t, \varnothing_t)$ using the optimized embedding \varnothing_t in null-text inversion can be well approximated by the velocity prediction $\epsilon_\theta(\bar{z}_t, t, C)$ using the embedding C of the input prompt in (5).

We next calculated the cosine similarity in the 768-dimensional embedding space between the embeddings C for 100 prompts and optimized embeddings $(\varnothing_t)_{t \in \{1, \dots, T\}}$ for each image. For each embedding sequence we took its average along the length of the sequence, and we centered the resulting average 768-dimensional prompt embeddings by subtracting the mean of 25,014 prompt embeddings, which are all the prompts included in the COCO validation dataset, and took a mean of embeddings over all tokens included in each prompt as the prompt embedding. Figure 8 right shows the mean cosine similarity. As t became smaller, the similarity between the optimized null-text embedding (\varnothing_t) and the embedding C of the given prompt became positive, whereas the similarity between (\varnothing_t) and embeddings C' of the prompts for images other than the target image, as well as that between C and C' , remained around zero. (We postulate that the small negative values of the similarity between C and C' throughout the entire range of t are due to the bias induced from the centering.) This suggests that, although the implicit “meaning” represented by the optimized null-text embedding was almost orthogonal to the “meanings” of those of randomly-chosen prompts, it was closer to the “meaning” represented

by the input prompt embedding C in the region distant from $t = T$, as can be observed by the larger values of similarity between the optimized null-text embedding and the embedding of the input prompt (Optimized vs Input prompt). In the region distant from $t = T$, except the region near $t = 0$, the model is thought to generate detailed information about the image, which should be crucial in obtaining a high-quality reconstruction, so that the higher values of similarity in this region would suggest that embeddings that would be good in the sense of yielding a good reconstruction are closer to the embedding C of the target prompt. In the large- t region, on the other hand, the optimized null-text embedding (\varnothing_t) had small similarity with the embedding C of the given prompt, which can be ascribed to the fact that the null-text optimization is initialized with the same null-text embedding \varnothing , and is performed from $t = T$ down to $t = 1$. Note that, in the large- t region, the similarity values were around zero because early stopping in optimizing \varnothing_t was effective and optimization barely progressed.

From these results, we can say that the optimized embedding \varnothing_t becomes semantically similar to the input prompt embedding C as the optimization progresses. Therefore, it has been confirmed that our inversion method approximates null-text inversion.

B. Implementation details

In our experiments, for the null-text inversion, we used the same settings at 50 sampling steps as those in the implementation available on the GitHub page of [19]. Optimization was performed with the Adam optimizer, and the learning rate was set to reach 5×10^{-3} at the last sampling step, changing linearly by the factor of 10^{-4} with the number of sampling steps. We further employed early stopping,

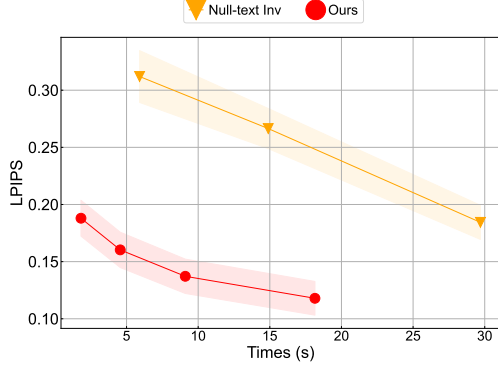


Figure 9. Comparison of LPIPS when calculation time was limited to less than 30 seconds.

and the threshold for early stopping was increased linearly in the number of sampling steps from 10^{-5} by the factor of 2×10^{-5} . We observed that when scheduling the learning rate and threshold with a function of diffusion steps, the reconstruction quality was getting worse. See our code included in SM for more detailed implementation settings of our experiments.

C. Additional experimental results

C.1. Comparison of reconstructed images

Figure 9 shows a comparison of LPIPS between null-text inversion and our method when the computation time was limited to less than 30 seconds. The number of sampling steps in null-text inversion was 2, 5, and 10. LPIPS of null-text inversion below 10 sampling steps was degraded, and our method outperformed it. Under the constraint of allowing feasible processing times, the reconstruction quality of our method was better than that of null-text inversion.

Figure 10 shows additional images reconstructed by the three methods compared. All the results show that DDIM inversion produced reconstructions that were not similar to the input images, while null-text inversion almost perfectly reconstructed the input images, and that our method also yielded results which were close to the reconstructions by null-text inversion.

C.2. Comparison of edited images using prompt-of-prompt

Figures 11 and 12 show additional images edited by prompt-to-prompt. As can be seen, DDIM inversion failed to perform editing while maintaining the details of the original images. On the other hand, null-text inversion and the proposed method were both capable of editing while maintaining details of the original images, including object replacement, style changes, size changes, and pose changes.

C.3. Comparison of edited images using other editing methods

We demonstrate the advantage of the proposed method that it can be combined with various editing methods. For this purpose, we performed editing experiments by combining the proposal with other editing methods, SDEdit [18] and Plug-and-Play [29]. In SDEdit, a certain ratio t_0 is used as a hyperparameter to add noise to the sample z_0 , and the latent variable z_t at the diffusion step $t = t_0 \cdot T$ is obtained, which is then reconstructed by tracing the inverse diffusion process. For image editing, z_0 is obtained from the original image and an edited prompt is used during the inverse diffusion process calculation. We set the noisy sample z_t calculated by DDIM inversion for null-text inversion and our negative-prompt inversion since they assume starting the sampling from z_T calculated by DDIM inversion. In Plug-and-Play, the null-text is used as a prompt for DDIM inversion. To combine the proposed method with it, we employed a prompt for the original image instead of the null-text for DDIM inversion.

Figure 13 shows images edited by SDEdit. As can be observed, SDEdit could not reconstruct the input images, while negative-prompt inversion and the proposed method were able to reconstruct details of the input images and appropriately edit them as specified by the prompts. Next, Figure 14 shows images edited by Plug-and-Play. Although the results of the proposed method were not generally better, the first, second, and fourth rows show better reconstruction quality and editing results in combination with the proposed method than the original method.

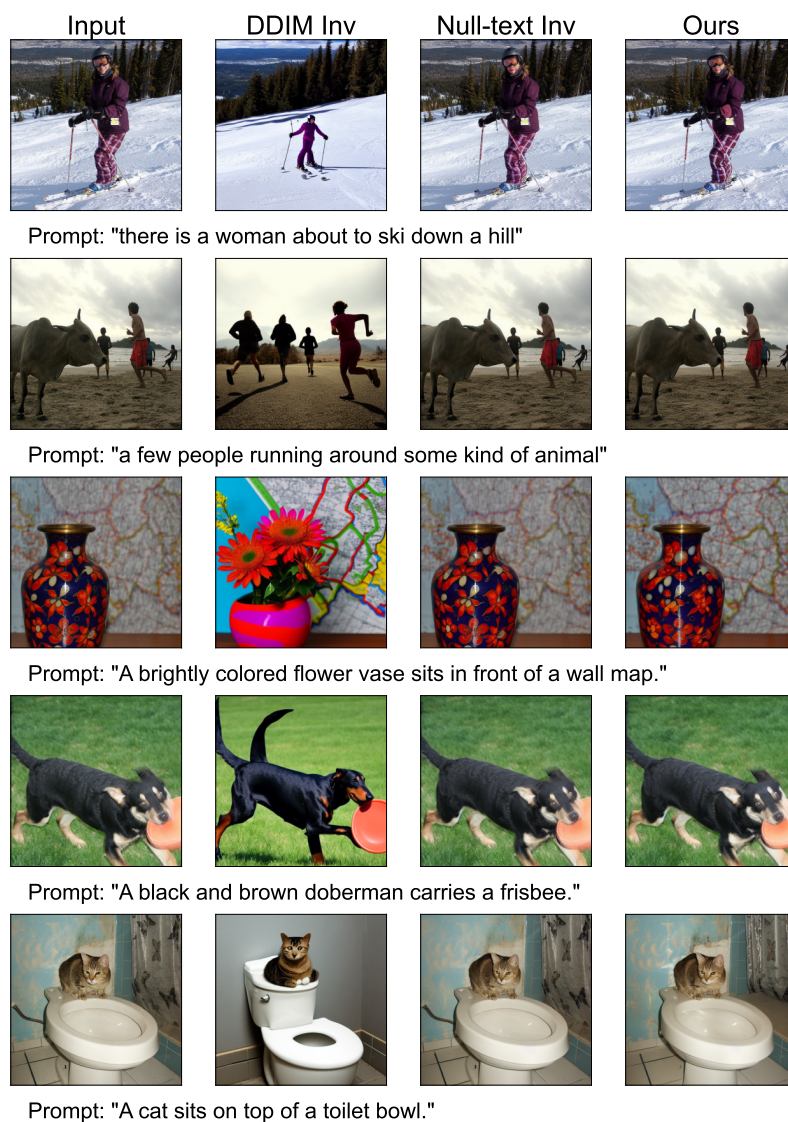


Figure 10. Additional results of reconstructed images by the three methods.

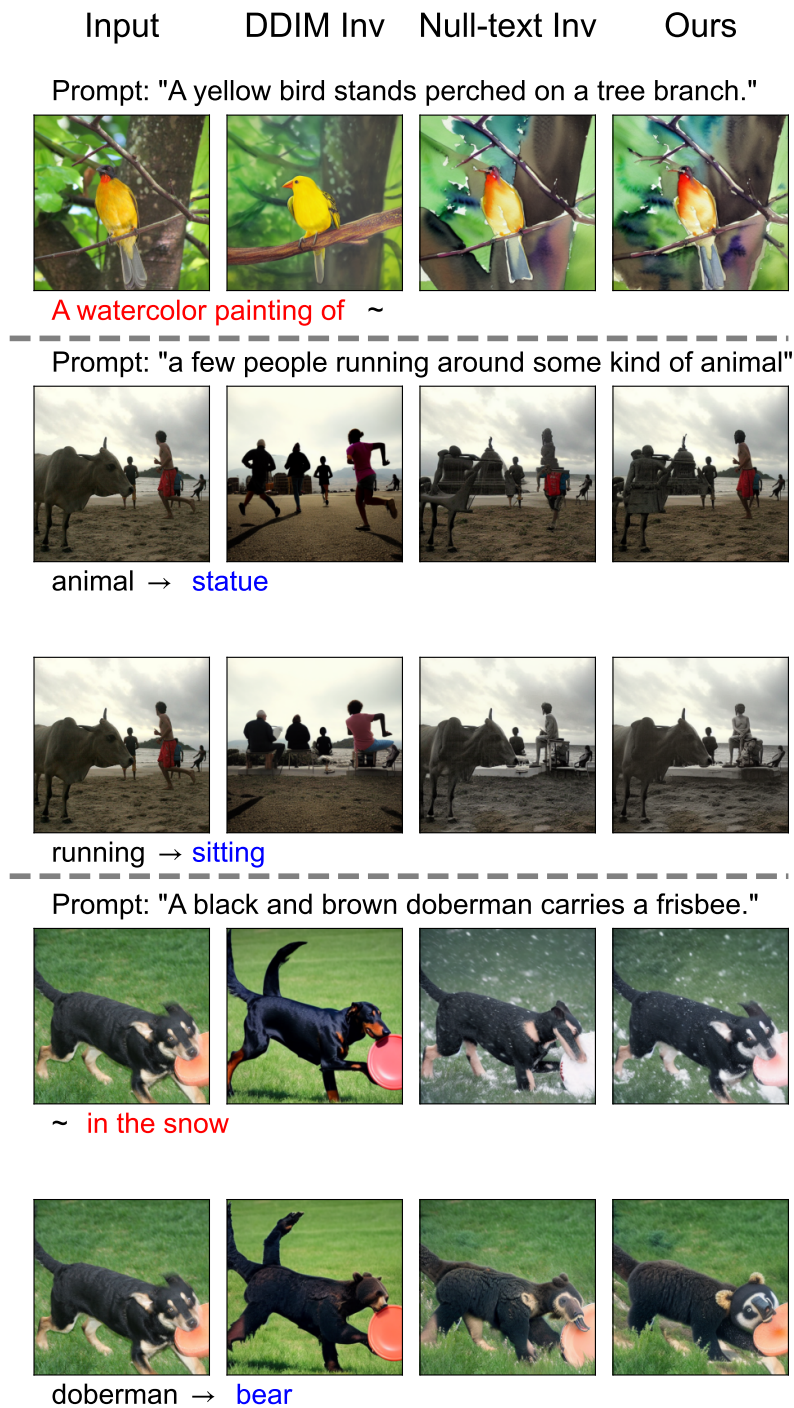


Figure 11. Additional results of edited images by prompt-to-prompt combined with the three methods.

Input DDIM Inv Null-text Inv Ours

Prompt: "A blue fire hydrant is on the brick sidewalk near trees."



A **big** blue fire hydrant ~

Prompt: "A dog chewing on top of a large white ball."



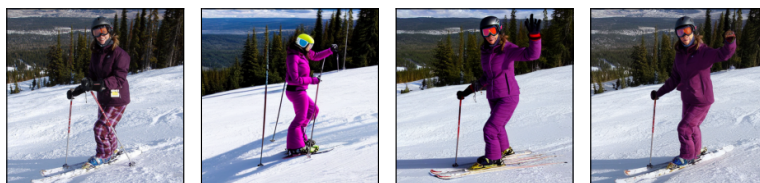
top → **side**

Prompt: "The longhorn sheep are grazing on the mountain."



grazing → **jumping**

Prompt: "there is a woman about to ski down a hill"



~ **with her hands spread**

Prompt: "A man with a tennis racket walks from the net of a tennis court."



~ **with his hand raised**

Figure 12. Additional results of edited images by prompt-to-prompt combined with the three methods.



Figure 13. Additional results of edited images by SDEdit combined with null-text inversion or the proposed method.



Figure 14. Additional results of edited images by Plug-and-Play combined with the proposed method.