Diffusion Inversion Report - Generative Image Editing via Inversion

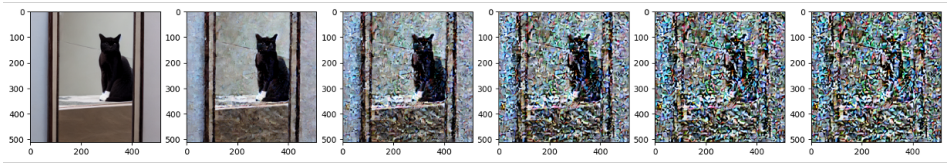
Ziyu Wu, Heyuan Liu

March 24, 2025

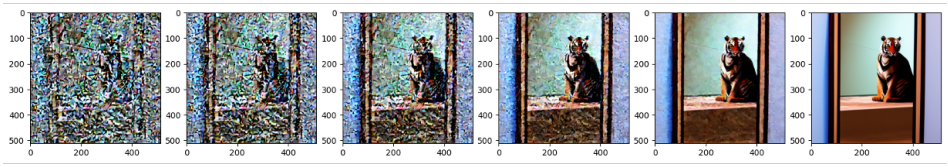
1 Background

1.1 Introduction

The goal of this project is to modify the source image according to the target prompt, ensuring that the generated image retains most of the original features while incorporating the changes described by the target prompt. As shown in the example in Figure 1



(a) Noising progress



(b) Denoising progress

Figure 1: Example: I want to change an image of "a cat sitting next to a mirror" to "a tiger sitting next to a mirror"

1.2 Theoretical Background

Denoising Diffusion Probabilistic Models (DDPM) [HJA20] and Denoising Diffusion Implicit Models (DDIM) [SME20] are two widely used frameworks for generative modeling via diffusion processes. Both methods rely on iterative denoising steps to transform noise into structured data, such as images, but they differ significantly in their formulation and sampling behavior.

DDPM: Stochastic Sampling via SDE DDPM is based on a stochastic differential equation (SDE) and involves a probabilistic reverse process. The reverse denoising step in DDPM can be written as:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z$$

where $z \sim \mathcal{N}(0, I)$ is a random noise term. This stochasticity introduces randomness at every denoising step, leading to diverse but slightly less controllable outputs. The noise term $\sigma_t z$ is essential for the stochastic behavior of DDPM.

DDIM: Deterministic Sampling via ODE In contrast, DDIM reformulates the reverse process as an ordinary differential equation (ODE), leading to a deterministic sampling path. The reverse denoising equation in DDIM is:

$$x_{t-1} = \sqrt{\alpha_{t-1}} \left(\frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta(x_t, t)}{\sqrt{\alpha_t}} \right) + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \epsilon_\theta(x_t, t)$$

In its fully deterministic variant (when $\sigma_t = 0$), this becomes:

$$x_{t-1} = \sqrt{\alpha_{t-1}} \left(\frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta(x_t, t)}{\sqrt{\alpha_t}} \right)$$

This formulation allows for *determined sampling*, which significantly speeds up the generation process while maintaining visual fidelity. As shown in Figure 1, the DDIM process generates a smooth transition from noise x_t to the target image without introducing new randomness during sampling.

2 Backbone Model and Baseline Results

2.1 Model and Data

2.1.1 Backbone Model

The "Stable Diffusion" model is from the `https://huggingface.co/CompVis/stable-diffusion-v1-4`,

2.1.2 Data Source

The data is from `https://huggingface.co/datasets/vinesmsuic/GenAI-Bench_image_edition_processed`, and in this project, we focus mainly on the "target prompt" and "source image"

2.2 Evaluation Metrics

Following the instructions from the proposal, we implemented the metrics named SSIM and CLIP (both image-image and image-text), as shown in Algorithm 1 and 2.

Algorithm 1 Compute SSIM (Structural Similarity Index)

Require: Target image X, Generated image Y
Ensure: SSIM score s
Convert X, Y to grayscale if necessary
if size(X) $\neq$ size(Y) **then**
    Resize Y to match X
end if
Compute means μ_X, μ_Y; variances σ_X^2, σ_Y^2; and covariance σ_{XY}
Compute $s = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)}$
return s

Algorithm 2 Compute CLIP Similarity

Require: Prompt T, Generated image Y, Target image X, CLIP model $\mathcal{M}$
Ensure: S_{text_image}, S_{image_image}
$\mathbf{T} \leftarrow \tau(T)$
$\mathbf{F}_Y \leftarrow \mathcal{M}.encode\_image(\psi(Y))$
$\mathbf{F}_X \leftarrow \mathcal{M}.encode\_image(\psi(X))$
$\mathbf{F}_T \leftarrow \mathcal{M}.encode\_text(\mathbf{T})$
Normalize all features to unit norm
$S_{text\_image} \leftarrow \mathbf{F}_Y \cdot \mathbf{F}_T^\top$
$S_{image\_image} \leftarrow \mathbf{F}_Y \cdot \mathbf{F}_X^\top$
return S_{text_image}, S_{image_image}

3 DDIM Inversion

In this section, our objective is using DDIM inversion to identify the inverted noise X_T for the source image. From this noise, regenerate images using the target prompt to incorporate specified modifications.

To achieve this, via diffusion-based methods, we adopt a two-stage process: (1) inversion and (2) regeneration. The inversion step aims to identify the latent noise x_T corresponding to a given source image. We implement this process using the `DDIMInverseScheduler` provided by the `diffusers` library from Hugging Face[1]. This scheduler performs the reverse diffusion process, gradually mapping the source image to its noise representation.

Once the inverted noise is obtained, we utilize the standard `DDIMScheduler` to regenerate images from x_T, conditioned on the new target prompt. This forward denoising process enables the incorporation of semantic modifications into the generated image, results in the example shown in Figure 1.

In our experiments, the default settings of both the regular and reverse DDIM pipelines generally yield satisfactory results. However, in some cases, there are noticeable mismatches between the target prompt and the generated image. For instance, as shown in Figure 2, when using the prompt *"Outdoor show with black sheep prominent on grass field"*, the output image in Figure 2c contains a sheep with a white coat, which contradicts the prompt. After adjusting the `guidance_scale` parameter in the DDIM pipeline, the result shown in Figure 2d better aligns with the prompt. Nevertheless, the overall image quality remains suboptimal.

(a) Source Image (b) Baseline Image

(c) Default DDIM (d) Modified DDIM (guidance scale)

Figure 2: Examples of DDIM inversion

[1] `https://github.com/huggingface/diffusers`

4 Null-Text Inversion

4.1 Principle

In practice, DDIM Inversion introduces errors at each step. For the unconditional expansion model, the accumulated errors can be ignored. However, for classifier-free guidance ($w > 1$), the accumulation of errors increases significantly, causing the final generated image to deviate from the original target and produce visual artifacts. To address this, Pivotal Inversion is proposed to correct the error accumulation in classifier-free guidance.

Experiments show that for classifier-free guidance expansion, when the guidance scale $w = 1$ (null-text), the DDIM Inversion trajectory $T1$ denoted as $(z_0^*, ..., z_T^*)$ provides a rough approximation of the original image, which is called the "pivot". If $w > 1$, directly using z_T^* for DDIM sampling results in trajectory $T2 = (\bar{z}_0, ..., \bar{z}_T)$, which deviates from the original trajectory $T1$ as shown in Fig.3.

To address this issue, the approach sequentially optimizes the sampling at each timestep t from $T = T$ to 1, making $T2$ as close as possible to $T1$. This ensures that the final trajectory ends near z_0, preserving the semantic and visual integrity of the original image. The optimization objective is:

$$\min \|z_{t-1}^* - z_{t-1}\|^2, \quad \text{where } z_0^* = z_0, \quad z_T^* = z_T. \tag{1}$$

Since $T2$ provides better initial values and is highly correlated with the original distribution, this optimization is efficient. Once z_{t-1} is optimized, it serves as the input for the next DDIM sampling step.

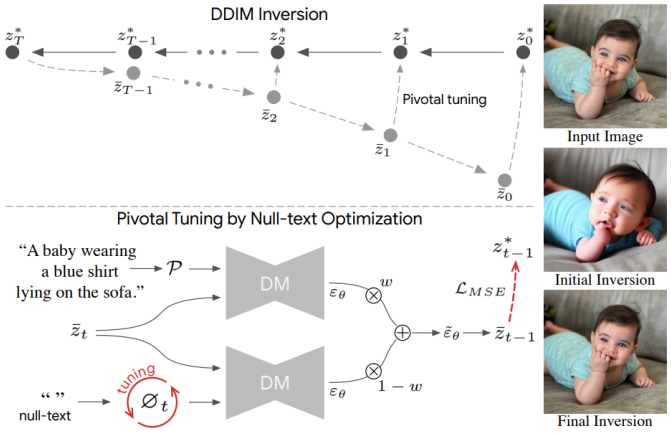


Figure 3: Null-text Inversion overview.

Considering the impact of unconditional guidance in diffusion models, the paper[MHA+23] introduce Null-text Optimization, which optimizes only the Null-text Embedding for each input image while keeping model weights and text conditions unchanged. After DDIM Inversion, the optimized null-text embedding can be used for multiple editing operations on the same input image.

Experiments reveal that, compared to using a single null-text embedding for all timesteps, optimizing different null-text embeddings per timestep significantly improves reconstruction quality. The overview of the algorithm is shown in Fig.3 Denoting the timestep-dependent embeddings as $\{\mathcal{O}_t\}_{t=1}^T$, the optimization process follows:

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

where $z_{t-1}(\bar{z}_t, \mathcal{O}_t, C)$ represents the DDIM sampling function, $\mathcal{O}_t$ is the null-text embedding, and C is the conditional text embedding. The update rule follows:

$$z_{t-1} = z_{t-1}(\bar{z}_t, \mathcal{O}_t, C). \tag{3}$$

Finally, the optimized null-text embeddings $\{\mathcal{O}_t\}_{t=1}^T$ can be used for image editing.

4.2 Evaluation and Results

To validate the effectiveness of Null-Text Inversion, we conduct multiple experiments to compare its performance with DDIM Inversion and baseline methods.

First we compare DDIM Inversion and Null-text Inversion across several examples (Fig.4). The results indicate that DDIM Inversion often introduces distortions and artifacts when modifying image content, whereas Null-text Inversion significantly enhances image preservation and editing quality. The refined null-text embeddings allow for more faithful reconstructions and smoother transitions between edited versions of the original image.

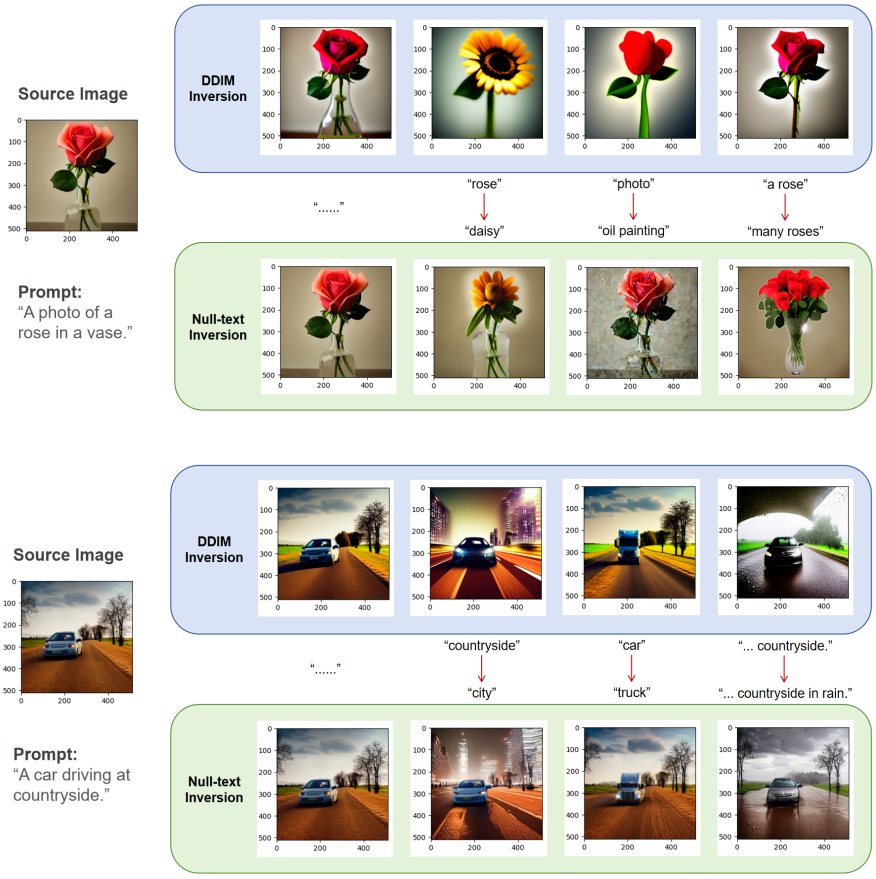
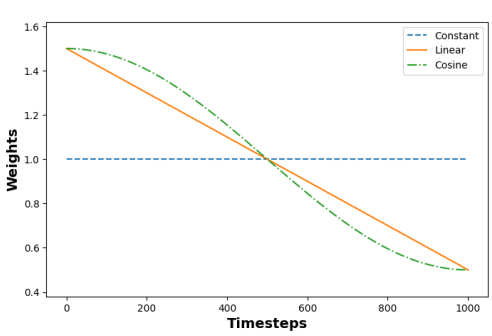


Figure 4: Examples of DDIM inversion and Null-text inversion.

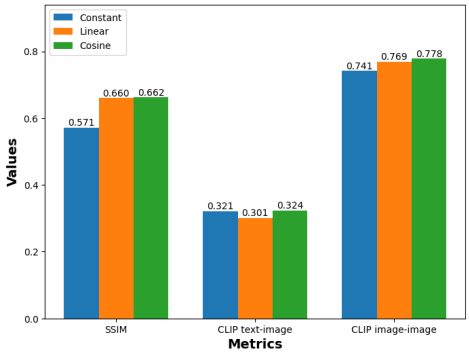
Another experiment examines the impact of different weight schedules in Classifier-free Guidance (Fig.5). We evaluate three approaches: constant, linear decay, and cosine decay. Our results show that decreasing the guidance weight over time yields better reconstructions, preventing over-enhancement of guided features. However, the difference between linear and cosine decay is marginal, indicating that both schedules are effective.

To further assess the performance, we conduct a large-scale comparison on 200 samples, evaluating three methods: the baseline model, DDIM Inversion, and Null-text Inversion (Fig.6). The evaluation metrics include SSIM, CLIP text-image similarity, and CLIP image-image similarity. Our findings indicate that: DDIM Inversion tends to distort fine details, especially under high guidance weights. Null-Text Inversion achieves the highest image fidelity, preserving structural features more accurately than other methods. However, Null-Text Inversion is computationally expensive, taking approximately 12.5 times longer than DDIM to generate a single image.

Despite the increased computational cost, Null-Text Inversion proves to be the best method for high-quality image editing and content preservation.

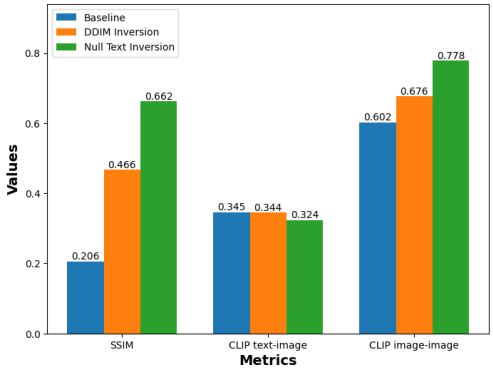


(a) Guidance weight functions.

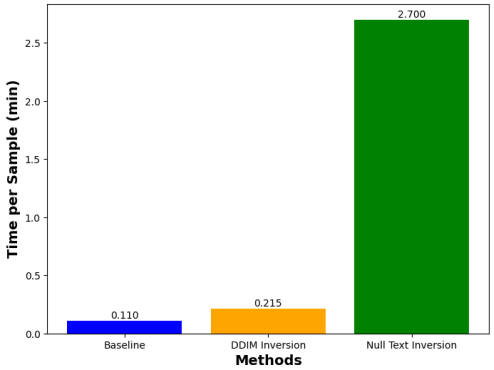


(b) Result of different functions.

Figure 5: Experiment on the guidance weight.



(a) Results on different methods.



(b) Time cost on different methods.

Figure 6: Experiment on different methods.

References

[HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[MHA+23] 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*, pages 6038–6047, 2023.

[SME20] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.