Inversion for Generative Image Editing

Konstantin Amelichev
konstantin.amelichev@polytechnique.edu

Aleksei Arzhantsev
aleksei.arzhantsev@polytechnique.edu

March 28, 2025

Abstract

In the project we research and implement methods for image editing that use diffusion models. We focus on methods that perform inversion: the search for the optimal trajectory in the latent space, that would balance between the similarity with the original image and the prompt aimed to change the image. We compare DDIM inversion and null-text inversion based on the structure kept during edition and semantic similarity to the prompt.

1. Introduction

Recent advances in the field of computer vision allow for powerful image generation methods. The most well-known techniques would be GAN's [2] and Diffusion models [4].

For the practical applications, it is crucial for users to have control over the generation results. With some methods such as classifier-free guidance [5], it is possible to account for prompting during image generation. But it is also important to be able to describe the desired result in a visual way.

The need for visual base for the generation brings us to the image editing problem, which we cover in this project. We research techniques based on diffusion models, that can take a source image and prompt as an input, and produce a resulting image that is structurally similar to the source image but also semantically matches the prompt.

1.1. Contribution

We did our best to split the workload equally. Both of us studied all of the theoretical background that we describe below. For the practical part, the baseline and DDIM inversion were initially done by Aleksei and Null-text inversion was initially done by Konstantin. However, the most time-consuming part was the debug and trial-and-error, that was required both for DDIM inversion and Null-text inversion, as well as experiments aimed to produce metrics and qualitative results: and this was mostly done by two members simultaneously in a paired programming fashion.

2. Theoretical background

In this section we describe the diffusion-based methods that can be used for image generation and image editing.

2.1. Diffusion models

Diffusion models are used for generative tasks. They are based on the diffusion process [4], that convert the image to the noise, gradually mixing the image with the Gaussian noise. Alongside with the diffusion process (forward process), the model is trained to reconstruct the noise on each step (reverse process).

Then we can generate a new image by taking a sample from the latent space and perform the denoising process using the trained model.

2.2. Classifier-free guidance

The generation technique described in 2.1 is not configurable yet, other than by adapting the dataset.

It was shown that the generation process can be tuned with prompting. We can learn two kinds of models during diffusion process: a prompted (*conditioned*) and unconditioned denoisers. Mixing their results will move the generation results towards prompt. The mixing parameter is called "guidance scale":

$$\varepsilon'_\theta(z, c) = \varepsilon_\theta(z, c) + \lambda \cdot (\varepsilon_\theta(z, c) - \varepsilon_\theta(z)) \tag{1}$$

2.3. DDIM Inversion

Denoising Diffusion Implicit Models (DDIM) [8] is a method that uses a different non-Markovian process to describe diffusion. With this method we still get the same forward process and training algorithm, but can choose between different sampling methods by varying the σ_t coefficient that is responsible for noise variance. In particular, we can set the variance to zero and get a deterministic sample process.

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

From this point on we will consider the deterministic case only.

We can also rewrite the DDIM sampling process as an ODE [1].

$$x_{t-1} - x_t = \sqrt{\alpha_{t-1}} \left(\sqrt{1/\bar{\alpha}_t} - \sqrt{1/\bar{\alpha}_{t-1}} \right) x_t + \sqrt{\alpha_{t-1}} \left(\sqrt{1/\bar{\alpha}_{t-1} - 1} - \sqrt{1/\bar{\alpha}_t - 1} \right) \epsilon_\theta(x_t) \quad (3)$$

Now, we can approximate the inverse of this ODE by replacing $t+1$ with $t-1$. In practice this should work with a small enough step between two consecutive images, or in other words with a big number of diffusion steps.

$$x_{t+1} - x_t = \sqrt{\alpha_{t+1}} \left(\sqrt{1/\bar{\alpha}_t} - \sqrt{1/\bar{\alpha}_{t+1}} \right) x_t + \sqrt{\alpha_{t+1}} \left(\sqrt{1/\bar{\alpha}_{t+1} - 1} - \sqrt{1/\bar{\alpha}_t - 1} \right) \epsilon_\theta(x_t) \quad (4)$$

This gives us an inversion algorithm to get a latent representation of any picture.

2.4. Null-text inversion

Null-text inversion is a technique that aims to improve quality of the editing methods [6].

The technique focuses on the reverse process. Having a basic classifier-free guidance formula, the denoised image after each state is a combination of the unconditional embedding and the conditional embedding. The method takes an unconditional embedding (also called as *null-text embedding*) and turns it into a learnable parameter. It is learned in a manner that will keep the source image embeddings, if conditions embeddings are taken with the source prompt.

To perform the training, MSE loss is used between the result of reverse process and the historical forward step result. The trained unconditional embeddings are saved and used later in the reverse process with the target prompt. You can see the visualisation of the training process on Fig. 1.

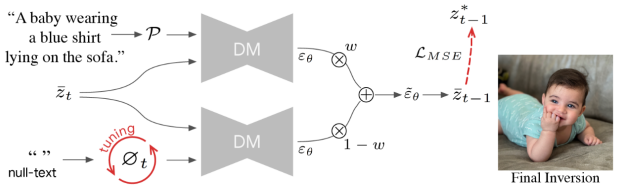

Figure 1. The training process of null-text inversion. z^*_{t-1} represent the latents obtained during DDIM inversion. Source: [6].

3. Practical work

In this section we describe our process of developing the methods for generative image editing.

Figure 2. Example of an image generation with classifier-free guidance.

3.1. Data

We use *vinesmsuic/GenAI-Bench_image_edition_processed* Huggingface dataset. It consists of source images, source prompts (image descriptions) and target prompts (desired image content after edition). All images are 720x720.

3.2. Baseline

For the baseline solution, we were generating images by prompt without any use of the source image. For consistency, this and all further experiments were run with the same StableDiffusion version (*CompVis/stable-diffusion-v1-4*), that we use as a base model. It supports classifier-free guidance out of the box, following the theoretical notation described in 2.2. In all of the experiments guidance scale of 7.5 was used.

The generation process will make image matching the prompt, but won't in any way adapt it to be close to the source image 2.

3.3. DDIM inversion

Overview of the algorithm is the following:

1. Start with the original image.
2. Perform inversion, where on each step we use the DDIM formula.
3. Take the resulting latent space vector p.
4. Perform classifier-free guidance starting from p.

The intuition behind this process is that generation process is locally-consistent in the latent spaces. This means that by starting from the inverted image, we will generate image close to the source one 3.

3.4. The issues with the DDIM inversion

DDIM inversion, while solving the editing problem, has some drawbacks. The main one is that DDIM inversion sometimes struggles to reconstruct the source image given the source prompt. The zebra image from mentioned in the previous part is good under one set of hyperparameters but

Figure 3. Zebra image, inversion of the zebra in latent space, and the generation result with source prompt, started from the corresponding inversion. Small guidance scale value of 1 is used.

Figure 4. Zebra image, inversion of the zebra in latent space, and the generation result with source prompt, started from the corresponding inversion. Small guidance scale value of 7.5 is used.

Figure 5. Sheep image, where the target prompt asks to change sheep to a black one. In the center we show the reconstruction from inversion, producing not the same image, but multiple sheeps. So the generated image with the black sheep is inconsistent with the original image and moves the sheep to another location.

is completely broken with higher value of a guidance scale **??**.

This happens because doing classifier-free guidance the process does not perfectly match reverse diffusion process: it does steps with the guidance scale. Because of this error accumulate with more steps and we get can get an absolutely different image as an output.

So we can see that the reconstruction of the original image is poor. Because of that any offset from the original image also provides weird result 5. With a big guidance scale we got some good prompt correspondence, but the image similarity was too low. And for the low guidance scale we had even worse results.

3.5. Null-text inversion

Overview of the algorithm is the following:

1. Start with the original image.
2. Perform inversion, where on each step we use the DDIM formula.

Figure 6. Sheep image, where the target prompt asks to change sheep to a black one. In the center we show the reconstruction from null-text inversion, producing almost the same image. The edited image has a black sheep in the similar place.

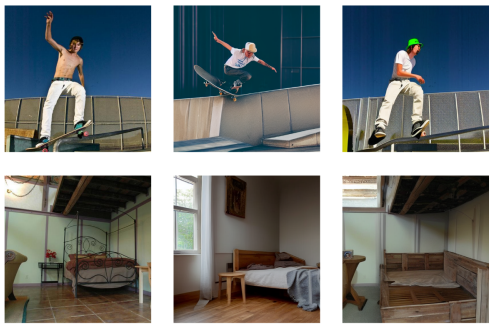

Figure 7. Editing capabilities of the methods. The left image is the source image, the middle image is the DDIM-based editing result, and the right image is the Null-text inversion editing result. The first row corresponds to the task "Add hat to the skater", and the second row corresponds to the task "Make the bed into a wooden one".

3. Take the resulting latent space vector p.
4. Fine tune unconditional embeddings $\varnothing_T$ so that result of generation starting with p and the source prompt would match the source image.
5. Perform classifier-free guidance starting from p with target prompt and finetuned unconditional embeddings $\varnothing_T$.

We tried to reconstruct source images with this method and got some good results: even for complex scenes it recovered the source image. And because of that the edited images also had better similarity 6.

4. Evaluation

We evaluated all the methods based on several metrics: CLIP similarity and SSIM. We also provide qualitative results to compare DDIM inversion and Null-text inversion 7. We noticed that hyperparameter have a high impact on the resulting images: all DDIM-inversion methods use guidance scale of 1, and reverse process for Null-text inversion editing uses guidance scale of 7.5.

Metric	Generated	DDIM Inversion	Null-text Inversion
SSIM	0.2151 ± 0.0825	0.4644 ± 0.1318	0.4894 ± 0.0971
CLIP Img2Img Target	0.7483 ± 0.0997	0.7930 ± 0.0960	0.7418 ± 0.0835
CLIP Img2Img Source	0.6991 ± 0.1061	0.7509 ± 0.1055	0.7060 ± 0.1018
CLIP Img2Text Target	0.3212 ± 0.0288	0.3181 ± 0.0318	0.3090 ± 0.0218
CLIP Img2Text Source	0.2911 ± 0.0384	0.2906 ± 0.0394	0.2643 ± 0.0374
Generation time	30 seconds	60 seconds	7 minutes

Table 1. Comparison of different inversion methods across various metrics.

4.1. CLIP Similarity

CLIP is an embedding that allows semantic matching between images and text [7]. We use it as it allows us to understand the extent to which the generated image matches the prompt.

4.2. SSIM

SSIM stands for Structure Similarity index measure [3]. It measures the difference between two images, considering them as distributions. The value is based on the mean and variance of the windows. We use it to measure the similarity between two pictures: the source image and the edited image.

4.3. Comparison

We provide the measured metrics for the methods implemented in the project 1.

As we can see, the Null-text inversion provides better similarity to the source image, while having a bit worse prompt correspondence. We explain it by the nature of the inversion process: on each step the state is denoised partially to match the prompt and partially to perform the reverse diffusion process to reconstruct the original image. That image is a source image, so it does not match the prompt itself. Comparing between null-text inversion and DDIM inversion, we get better CLIP similarity scores for the DDIM inversion as null-text inversion learns null-text embeddings to match the original image better, making a tradeoff between similarity and prompt correspondence.

One more concern would be the generation time: for the DDIM inversion the generation takes approximately twice as many steps as in simple generation. This happens because DDIM inversion with generation requires both forward and reverse processes, while basic generation only requires a reverse process. The Null-text inversion takes more time as it requires a small training loop for each step of the diffusion.

5. Conclusion

In this work, wéve researched and developed several methods for image editing: DDIM inversion and null-text inversion. We can see that Null-text inversion provides better consistency with the source image, while performing a change that matches the generation prompt.

References

[1] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 8780–8794, 2021. 2

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 1

[3] M. Hassan and C. Bhagvati. Structural similarity measure for color images. *International Journal of Computer Applications*, 43(14):7–12, 2012. 4

[4] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[5] J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 1

[6] R. Mokady, A. Hertz, K. Aberman, Y. Pritch, and D. Cohen-Or. Null-text inversion for editing real images using guided diffusion models. pages 6038–6047, 2023. 2

[7] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 4

[8] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1