ESE 6450 Final Project Report - Benchmarking P2P Modifications Against DDIM Inversion

Marika Nishi*
10501291
marikan@seas.upenn.edu

Edward Zhang*
65702495
zedward@seas.upenn.edu

Yi Fan Li*
15471810
yli1@seas.upenn.edu

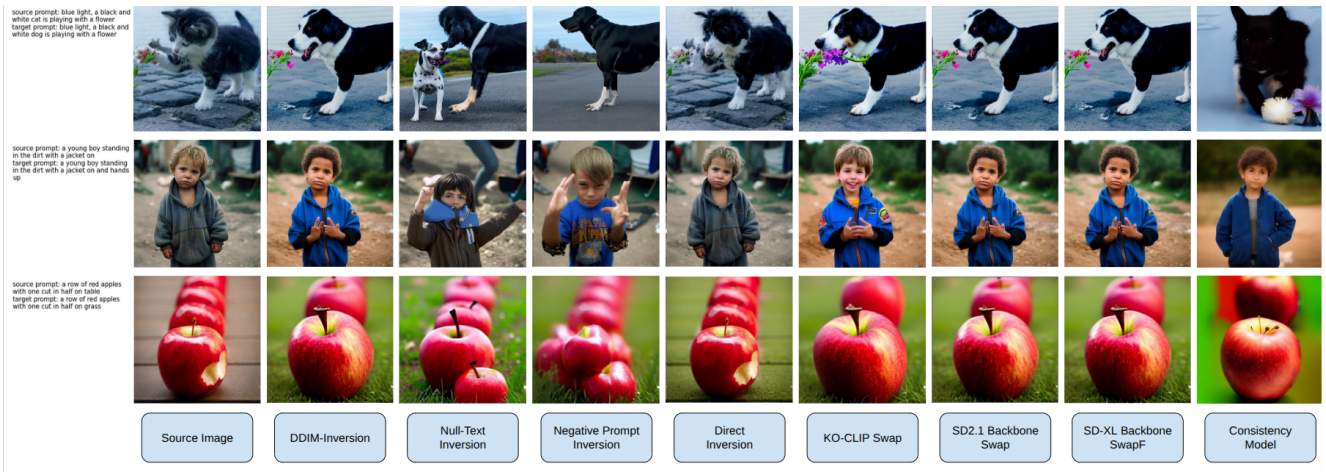

Figure 1. Comparison of different generations.

Abstract

Diffusion models enable high-quality image generation but still offer limited control when modifying an existing image, often producing new samples instead of targeted edits. Deterministic and approximately invertible sampling methods such as DDIM have recently enabled Prompt-to-Prompt (P2P) editing, which manipulates the diffusion trajectory to apply fine-grained changes while preserving the source image's structure. In this report, we benchmark several P2P editing techniques across tasks including background changes, pose adjustments, and style modifications. Using semantic and pixel-level similarity metrics, we evaluate how well each method balances edit precision with source-image fidelity. Our findings highlight the strengths and limitations of current P2P approaches and point to remaining challenges for reliable diffusion-based image editing.

*Equal contribution.

1. Introduction

The recent rise of diffusion based generative models has opened a world of possibilities for data generation, but these models off very little or coarse control over the final outputted images. Simply put, if the generated image is only within the ballpark of the desired end product, off-the-shelf diffusion models are not good at merely editing these existing images, rather often just generating entirely new images.

The rise of Implicit Diffusion Models (DDIMs), which offer a deterministic and invertible framework for diffusion based image encoding and decoding (generation), has paved the road for Prompt-to-Prompt (P2P) image editing techniques that strive to offer finegrained control to edit, alter, and modify existing images while preserving the base priors of source image. The seminal P2P technique that grew out of this was that of naive DDIM inversion, in which a DDIM is run in reverse on a source image to produce a latent representation of said image. This latent representation would then be used as an initialization for another generation, this time conditioned on a prompt that specifies how it should be edited. This approach leaves room for many possible

improvements both in the specifics of the training regime utilized as well as general plug and play improvements to the disparate parts of its architecture.

In this technical report, using DDIM Inversion as a baseline, we benchmark the performance of six modifications to this baseline P2P image editing technique on image editing tasks including but not limited to altering the background, pose, and style of a source image. We first quantitatively assess the performance of these modifications using three image similarity metrics. We then offer qualitative commentary on the generated outputs. Finally, we will engage in a higher level discussion on the lessons learned and areas for potential future work in this field.

2. Related Work

2.1. Diffusion Models

The task of image generation was gated for several decades by compute restraints and mathematical formulations such as Probabilistic Principal Component Analysis (PPCA) [15] that could not sufficiently model images at a sufficiently rich level. This changed when the deep learning revolution of the 2010s led to an explosion of deep learning based image generation methods such as Variational Autoencoders (VAEs) [6] and Generative Adversarial Networks [1]. This would come to a head when Denoising Diffusion Probabilistic Models (DDPMs) would introduce the notion of generating images by predicting marginal amounts of noise to be removed from a noisy starting point.

These diffusion models are trained by gradually corrupting an image with noise and then having a UNet learn to iteratively predict and remove that noise to recover the underlying signal. By decomposing the generation process into many small denoising steps—each responsible for only a tiny portion of the image's global and local structure—diffusion models avoid the burden of predicting all features at once, enabling far more stable training and significantly higher image quality than prior generative methods.

To achieve conditional, controllable generations, diffusion models could be coupled with a text encoder which could convert natural-language prompts into vectors usable as conditioning embeddings. This enabled the diffusion process to be guided at every denoising step through cross-attention, allowing semantic information from the prompt to influence the reconstruction trajectory. This combination of iterative denoising and prompt-based conditioning ultimately led to modern text-to-image systems such as Stable Diffusion [11], where high-level language cues consistently shape both the global composition and fine details of the generated image.

The iterative nature of diffusion models also sparked extensive research into methods for accelerating their generation time. Implicit Diffusion Models (DDIMs) [12], which reinterpret the reverse diffusion process as a deterministic ODE and enable high-quality sampling in far fewer steps. More recently, this line of work culminated in consistency models, which distill the behavior of a full diffusion model into a one- or few-step generator that directly maps noisy inputs to clean outputs while maintaining distributional fidelity [13]. Together, these approaches aim to preserve the strengths of diffusion models while mitigating their computational bottlenecks, pushing the field toward faster and more practical generative systems.

2.2. Prompt-to-Prompt Editing

Prompt-to-Prompt (P2P) editing [2] is a diffusion-based image editing technique that enables users to modify an image using only textual prompts while preserving its core structure. The key idea is to take advantage of the diffusion model's denoising trajectory: instead of generating a new image from noise, P2P partially reuses the trajectory of an existing image and alters only the semantic components influenced by the prompt. This allows fine-grained edits—such as changing style, background, or object attributes—without disrupting the identity or composition of the source image.

A central challenge in achieving this level of control is the need to invert a diffusion model: given an observed image, recover the latent representation that the model could have produced during sampling. Early diffusion models such as DDPMs are fundamentally Markovian, meaning each denoising step depends only on the previous timestep through a stochastic transition kernel. While this Markovian structure is crucial for training, it makes the reverse path nondeterministic and therefore unsuitable for accurate inversion—multiple latent trajectories could lead to the same image, and running the forward-noising process cannot reliably reconstruct a usable latent.

This limitation was addressed by DDIMs, which reinterpret the diffusion sampling process as a deterministic non-Markovian ODE. Because DDIM sampling removes stochasticity and defines a unique mapping between timesteps, it becomes approximately invertible: the same deterministic update rule used to denoise can be algebraically reversed to map an image back to a corresponding latent code. This insight is what makes P2P feasible.

Thus, DDIM inversion naturally became the foundational mechanism for performing P2P editing: instead of generating an image from random noise, the model first inverts the observed image through the deterministic DDIM update rule to recover its latent diffusion trajectory. This produces a sequence of latent states that behave as though they originated from the model's own sampling process. Once this trajectory is obtained, P2P simply resamples it while modifying the text embeddings associated with spe-

cific prompt tokens, enabling localized semantic edits without discarding the structural information encoded in the original latents.

3. Methods

DDIM inversion provides a reliable and deterministic starting point for P2P editing, making it the standard baseline for comparison. The following sections introduce several variants aimed at improving reconstruction, stability, or semantic precision, and we benchmark each against naive DDIM inversion.

3.1. Null Text Inversion

Null-Text Inversion [8] as a P2P editing technique seems to improve reconstruction quality by factoring the task of retaining information from original image along side the task of generating the new edited version. It goes about doing this by optimizing a null (i.e., empty) text embedding used during classifier-free guidance. Instead of relying on a fixed unconditional embedding, Null-Text Inversion learns an optimal value for this embedding so that, when used during DDIM inversion, the model can exactly reconstruct the input image. This optimization produces a sequence of unconditional embeddings—one per diffusion timestep—that better captures the image's structure and helps maintain fidelity during subsequent edits. In practice, this leads to more accurate inversions but comes at the cost of increased computation, since it requires a full optimization loop over all timesteps.

3.2. Negative Prompt Inversion

The problem of Null Text Inversion is it takes time because it learns a sequence of optimized embedding per diffusion timestep. To process faster, Negative Prompt Inversion just uses text prompt embedding by assuming optimized embedding is similar to the text prompt embedding. In this way, we just have to run diffusion once with fixed embeddings. It achieves image reconstruction equivalent to DDIM inversion solely through forward propagation without optimization. In the paper, Negative Prompt Inversion achieved 30 times faster than Null Text Inversion [7]. In the negative prompt, you do not have to make negative prompts, because it uses positive prompt when reconstructing an image, and source prompt when editing an image.

3.3. Direct Inversion

P2P with Direct Inversion builds upon previous methods by disentangling the source and target branch in the forward process, without using optimization or additional prompts embedding, by adding three lines of code [5]. It does this taking the difference between the latent of the inversion, and the latent of the source branch, and adds the difference back to the source branch. This method improves the reconstruction by reducing the distance between the source latent and the inversion latent to zero, enforcing essentially perfect reconstruction. The target branch then make the edits using the P2P method on its latent.

3.4. Text Encoder Swap

Another factor important in P2P editing is the text conditioning mechanism, since the quality and robustness of the text encoder directly influence how prompt modifications affect the denoising trajectory. CLIP—the text encoder used by Stable Diffusion—has well-known weaknesses, including excessive sensitivity to visible text in images, vulnerability to typographic attacks, and reliance on a small set of high-norm "register tokens" that disproportionately shape attention. These issues can introduce spurious semantic biases into the text embeddings and ultimately reduce the reliability and consistency of prompt-controlled edits.

To address this, we incorporate CLIP-KO [16], a popular fine-tuned variant of CLIP ViT-L/14 designed to improve robustness while retaining full architectural compatibility with existing diffusion pipelines. CLIP-KO applies key-projection orthogonalization, attention-head dropout, geometric parametrization, and adversarial training to mitigate CLIP's text-reading obsession and reduce unwanted attention artifacts. Importantly, CLIP-KO remains in the CLIP embedding space, meaning it can be used as a drop-in replacement for the original text encoder without requiring any retraining of the diffusion model or UNet. This makes it an attractive modification for P2P editing, where more stable and semantically reliable text embeddings can translate directly into cleaner, more controllable edits.

3.5. Diffusion Backbone Enhancement

Diffusion backbone is a model for editing images. The default implementation used stable-diffusion-v1-4. To enhance this, we employed stable diffusion v2-1-base Model [11] and SD-XL 1.0-base Model [3].

- Stable Diffusion v2-1-base Model
  This model fine-tunes stable-diffusion-2-base with 220k extra optimization updates. In this model, a classifier recognized 98 % of training images as safe, so this is more reliable.
- SD-XL 1.0-base Model
  This model uses two fixed, pretrained text encoders (OpenCLIP-ViT/G [4] and CLIP-ViT/L [10]). Compared to previous versions of Stable Diffusion, SDXL has a three times larger UNet backbone. It also uses a refeinement model that improves that visual quality of samples generated by SDXL using a post-hoc image-to-image technique.

Metric	DDIM	Direct Inversion		Negative Prompt	
		value	% change	value	% change
SSIM	0.53059	0.52788	−0.51%	0.53488	0.81%
LPIPS	0.36890	0.36944	0.14%	0.36987	0.26%
CLIP Sim.	27.05872	26.78011	−1.03%	24.79463	−8.37%

Metric	DDIM	Null Prompt		Encoder Swapped	
		value	% change	value	% change
SSIM	0.53059	0.50019	−5.73%	0.51716	−2.53%
LPIPS	0.36890	0.39413	6.84%	0.37519	1.70%
CLIP Sim.	27.05872	25.55748	−5.55%	27.07406	0.06%

Metric	DDIM	Backbone Swapped		Consistency Model	
		value	% change	value	% change
SSIM	0.53059	0.53053	−0.01%	0.40389	−23.88%
LPIPS	0.36890	0.36892	0.00%	0.55809	51.28%
CLIP Sim.	27.05872	27.05015	−0.03%	25.09279	−7.27%

Table 1. Comparison of DDIM baseline with different inversion variants. Percentage changes indicating better performance are highlighted in green and those with poor performance, red.

3.6. P2P + DDIM + Consistency Models

Naive P2P + DDIM runs take a long time to complete, with a runtime per image of around 20-30 seconds on a Nvidia 4080 gpu. We attempted to shorten the runtime by reducing the number of steps taken during the inversion and the reconstruction process. Consistency models are a relatively new type of model which can predict the output image in as few as one step, which made it a natural fit to our goal of speeding up the runtime. However, when applied to image editing with inversion, consistency models have inherent issues. Firstly, consistency models are forward models, trained to predict the clean final image, meaning they are not trained to perform the inversion operation. Not only are they not trained to do so, consistency models are not invertible as a consequence of its design. The main property of consistency models guarantees that points along a probability-flow ordinary differential equation will map to the same origin. However, the models will also match many x_t to a single x_0, which means it is impossible to invert using a consistency model [13].

As such, we used DDIM for the inversion, and then used consistency model for the forward process. For the number of steps, we had to strike a balance between fast processing and quality of the reconstruction; doing the forward process in a single step would not allow the p2p algorithm to guide the editing, while taking too many steps would go against the objective of shortening the runtime. We picked a step number of 12, which resulted in a runtime per image of around 5 seconds, 4 times faster than naive ddim, on the same Nvidia 4080 GPU.

4. Results

4.1. Metrics Used

We evaluated our models on three common metrics used in image editing: structural similarity (SSIM), Learned Perceptual Image Patch Similarity (LPIPS), and CLIP similarity. SSIM is a formula and method that is used to measure the similarities between two images [9]. LPIPS similarly also measures the similarity between images, but uses deep perception models to predict the metric [17]. CLIP similarity is used to measure the similarity between the prompt and the images. In combination, SSIM and LPIPS measures the similarity of the edited image with the source image, while CLIP similarity measures the quality of the edits made to the images. Table 1 shows the comparison of the different methods using SSIM, LPIPS, and CLIP similarity metrics. For SSIM and CLIP similarity, the higher the better, and for LPIPS, the lower the better.

4.2. Dataset Used

We evaluate all inversion and editing methods on the PIE-Bench (Prompt-driven Image Editing Benchmark), introduced in the Direct Inversion paper [5]. PIE-Bench provides a structured and diverse testbed for prompt-based image manipulation, enabling consistent comparison across editing techniques.

Each sample consists of a source image, a caption describing that image, and a modified caption specifying the desired edit. The benchmark spans ten high-level edit categories:

- Random
- Change Object
- Add Object
- Delete Object
- Change Content
- Change Pose
- Change Color
- Change Material
- Change Background
- Change Style

These categories are further organized into *Artificial* and *Natural* domains, with additional subdivisions such as *Animal*, *Human*, *Indoor*, and *Outdoor* depending on the type of edit. This hierarchical structure captures a broad spectrum of editing scenarios—from simple attribute changes to complex structural manipulations—making PIE-Bench an ideal benchmark for evaluating both reconstruction fidelity and semantic editing performance.

By providing paired captions that explicitly define the target modification, PIE-Bench allows us to quantitatively and qualitatively assess how well each inversion method preserves the source image while following the intended

edit. Within this dataset, there are roughly 700 total image-edit-caption-pairs.

4.3. Null Inversion

Across editing categories, Null-Text Inversion consistently underperforms the DDIM baseline in both SSIM and LPIPS, with average degradations of 5–10%. CLIP similarity also decreases for nearly every category, indicating weaker alignment between the edited image and the target prompt. Although a few isolated categories show slight improvements, the general trend is a decline in both reconstruction fidelity and semantic accuracy.

Qualitatively, Null-Text Inversion often exhibits *semantic confusion*. In one example, the source image contained a black-and-white (two-toned) cat, and the edit prompt requested "a black and white dog." Instead of producing a single dog with two-toned coloration, the model generated two separate dogs—one entirely black and one entirely white. Similar failures appeared in other categories, where the method misinterpreted color, object count, or composition.

These results suggest that while optimizing the unconditional embedding improves inversion consistency in theory, in practice it can destabilize the semantic grounding of the edit, causing the model to drift away from both the source image and the intended meaning of the prompt.

4.4. Negative Prompt Inversion

When we used Negative Prompt Inversion, SSIM was better than DDIM, but LPIPS and CLIP similarity was worse. The generated images had wrong/crooked points, as shown in figure 1. The dog image doesn't have flowers as pointed in the text prompt, the picture with a boy has weird hand shapes, and the apple image shows an oddly-shaped apple stack on other apples.

4.5. Direct Inversion

Direct Inversion, as the latest improvement in the domain of image editing, surprisingly does not show much better metrics when compared to base DDIM. The differences are minor, but it does show a minor degradation in performance in both SSIM and LPIPS, while also showing a minor degradation when measured by CLIP similarity. Qualitatively, the resulting images featured a much better reconstruction of the original source image. Likely as a result, the edits made were often more limited as well when compared to the other methods, to the point it does not make any edits to some images, as we can see in figure 1.

4.6. Text Encoder Swap

Replacing the default Stable Diffusion text encoder with CLIP-KO did not improve reconstruction fidelity, as both SSIM and LPIPS generally worsened across categories (Table 1). This indicates that the encoder swap makes the edited images deviate slightly more from the original source content at both structural and perceptual levels.

CLIP similarity, however, shows a different pattern: while not uniformly higher, several categories exhibit noticeable improvements. This behavior is consistent with the expected effect of CLIP-KO. Because CLIP-KO produces more robust and semantically stable text embeddings, it can strengthen alignment between the target prompt and the resulting image. In categories where semantic grounding plays a larger role than exact reconstruction—such as object or attribute modification—the model benefits from the improved text representation, leading to higher CLIP similarity scores.

Overall, the encoder swap introduces a trade-off: edits become somewhat less faithful to the source image, but in many cases more faithfully reflect the intended prompt. This suggests that CLIP-KO enhances semantic controllability at the cost of reconstruction precision.

4.7. Diffusion Backbone Swap

When we used SD2.1 for diffusion backbone of DDIM + P2P, SSIM was lower by 0.0001, LPIPS was the same, and CLIP similarity was 0.085 lower compared to the default DDIM + P2P (baseline). However, these values were almost the same. This is visible too in the generated images. The image results were almost the same as the ones generated by the default diffusion backbone, as shown in figure 1 When we used SDXL for diffusion backbone of DDIM + P2P, SSIM was lower by 0.0001, LPIPS was the same, and CLIP was lower by 0.085 compared to the default DDIM + P2P (baseline), which was the same as when we used SD2.1. Also, the generated images did not show visible difference when compared with the default diffusion backbone (SD1.4) and SD2.1. From these results, it is estimated that diffusion backbone of SD 2.1 and SDXL did not make effects in improving the diffusion.

4.8. Consistency Model

The results for the consistency model shows that they do not work well with the P2P guidance. The CLIP similarity showed the least decline, showing the models followed the target prompt during the reconstruction. However, both SSIM and LPIPS show a major performance degradation. It shows the consistency model do not preserve the source image, instead it follows its own method and essentially regenerates a new image, based on the prompt given.

4.9. Conclusion

Overall, it is worth questioning whether these metrics were sufficient to evaluate the quality of the edits on the images. We could see some images with bad scores looked actually

good and the ones with good scores looked bad. Taking reference from figure 1, direct inversion does the least amount of editing to the original image. That trend holds across the rest of the images, but direct inversion still scored slightly worse than the DDIM baseline. In contrast to that, the metric also has been successful in showing the flaws of consistency models in the forward reconstruction.

5. Lessons Learned and Discussion

The allotted scope of this project highlighted several practical limitations of working with text encoders in diffusion-based editing pipelines. Although state-of-the-art language models offer far stronger semantic representations than CLIP, integrating them into Stable Diffusion would require full-weight retraining of the generative backbone to ensure alignment between the embedding space and the UNet's cross-attention layers. Simply inserting dimensionality-matching wrappers would be conceptually incorrect: the UNet has been trained to interpret CLIP-specific embedding geometry, and forcing an unrelated embedding space through the same attention channels would break the learned correspondence between textual semantics and visual features. As a result, exploring more advanced text encoders was beyond the feasible scope of this project.

Another limitation lies in the loss functions used to train diffusion and P2P-style editing methods, which are dominated by pixel-wise noise-prediction objectives. These losses encourage photorealism and denoising fidelity but remain largely blind to higher-level semantic or structural correctness. This raises an intriguing possibility: could incorporating higher-level feature losses—for example, CLIP-space alignment, VGG-style perceptual features, or cross-attention consistency—yield edits that are both visually coherent and semantically faithful? This remains an open avenue for future work.

Finally, our experiments surfaced the persistent challenge of textual ambiguity. In one example, the prompt "make the dog black and white" was interpreted by a model as "generate one black dog and one white dog," rather than "apply black-and-white coloration to the same dog." This type of misalignment suggests that current P2P pipelines lack mechanisms to reconcile differences between the source caption and the edited caption. Strengthening cross-attention interactions or explicitly modeling relationships between original and target prompts may be necessary to resolve these ambiguities and prevent unintended edits.

Moreover, while diffusion model runtime is not the chief focus of image editing methods, faster image editing would open the possibility for more general application, if it did not come at the cost of editing quality. Consistency models appeared to be the most promising solution, given it is distilled from diffusion models and can relatively efficiently replace the diffusion backbone in the forward process. The results proved us wrong however. An inversion based on DDIM is not compatible with a forward process using consistency models, as the latter do not reconstruct the source image from the source latent. In spit of the naive method of integrating consistency models not yielding results, more fundamental reworking of the pipeline can yield results. For example, in Starodubcev's 2024 paper, [14], they trained two multistep consistency model, one for the forward and one for the backward process, and achieved good results on PieBench benchmark, though they evaluated on different metrics than us.

6. Contribution

6.1. Edward

For my technical contributions, I worked off of the existing PnP repository, tested several wrappers on state of the art text encoders like Siglip and T5 to slot into Stable Diffusion 1.4 but when that yielded complete gibberish, I ended up going with a clean swap of an existing finetuned checkpoint of CLIP-KO as an improved text encoder module to our image generation pipeline, as well as handing the baseline benchmarking of the Null text inversion approach.

For my logistical contributions, I played the role of the team leader in terms of delegating tasks, setting deadlines, and architecting the structure and story of our final presentation and accompanying technical report. I also contributed a significant portion of the compute needed for this benchmark due to the hardware constraints of my teammates.

6.2. Yifan

I edited the PnP repository to be compatible with consistency models, and reworked the code to add an additional pipeline for P2P+DDIM inversion+Consistency models. I ran the evaluation on the benchmarks for the consistency models, and Direct Inversion + P2P. For the presentation, I made the slides and presentation for Direct Inversion and consistency models. For the report, I wrote the sections related to those models, as well as the section on the metrics.

6.3. Marika

I edited the PnP repository for P2P+DDIM (SD2.1), and P2P+DDIM (SDXL), and ran P2P+DDIM. I also ran the evaluation code and organized the evaluation metrics values for P2P+DDIM, P2P+Negative Prompt Inversion, P2P+Direct Inversion, P2P+DDIM (SD2.1), and P2P+DDIM(SDXL) by making a table. For the presentation, I made the slides and presentation for Negative Prompt Inversion and Backbone Enhancement. I also organized the source and target images, and made figure 1 with Edward. For the report, I wrote the report in the parts related to these methods (Related Work, Methods, and Results).

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014. 2

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

[3] Hugging Face Model Hub. stabilityai/stable-diffusion-xl-base-1.0. https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0, 2025. Accessed: 2025-11-26. 3

[4] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. If you use this software, please cite it as below. 3

[5] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Direct inversion: Boosting diffusion-based editing with 3 lines of code, 2023. 3, 4

[6] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014. 2

[7] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. *arXiv preprint arXiv:2305.16874*, 2023. 3

[8] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models, 2022. 3

[9] Jim Nilsson and Tomas Akenine-Möller. Understanding ssim, 2020. 4

[10] OpenAI. Clip: Contrastive language–image pretraining — github repository. https://github.com/openai/CLIP, 2021. MIT License. Accessed: 2025-11-26. 3

[11] 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, 2022. 2, 3

[12] Bhomik Sharma. Mastering ddim: A simple guide to faster ai image generation. https://learnopencv.com/understanding-ddim/, 2025. LearnOpenCV, Accessed: 2025-11-26. 2

[13] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models, 2023. 2, 4

[14] Nikita Starodubcev, Mikhail Khoroshikh, Artem Babenko, and Dmitry Baranchuk. Invertible consistency distillation for text-guided image editing in around 7 steps, 2024. 6

[15] Michael E. Tipping and Christopher M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3): 611–622, 1999. 2

[16] zer0int and GPT-4.1. Clip-ko: Knocking out typographic attacks in contrastive language-image pre-training. *arXiv preprint arXiv:2504.04893*, 2025. 3

[17] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. 4