Enhanced Text Guided Image Editing Techniques and Consistency Models

Aakash Agarwal
University of Pennsylvania
Philadelphia, PA, USA
PennId: 34834428

Saaransh Pandey
University of Pennsylvania
Philadelphia, PA, USA
PennId: 42712466

Shubhan Pawar
University of Pennsylvania
Philadelphia, PA, USA
PennId: 33401078

Abstract

Text-driven image editing faces challenges in balancing structural fidelity, semantic alignment, and efficiency. We evaluate three baseline models: Edit-Friendly (EF) P2P, MasaCtrl, and Direct Inversion (DI), exploring enhancements such as richer text encoders, advanced diffusion backbones, and ControlNet integration. Additionally, we implement a Consistency Model, which outperforms in editing quality and inference time. Our evaluations highlight trade-offs across metrics, positioning the Consistency Model as a promising solution for efficient, high-quality text-driven image editing.

1. Introduction

Text-driven image editing has emerged as a significant area of research, leveraging diffusion-based generative models to achieve detailed and semantically meaningful manipulations of images based on textual inputs. Despite advancements, achieving a balance between preserving the structural fidelity of the original image and enabling diverse, precise edits remains a challenge. Existing methods, such as DI P2P, EF P2P, and mutual self-attention control (MasaCtrl), have made strides in addressing these issues but often face trade-offs in terms of flexibility, precision, and computational efficiency.

In this project, we implemented and extended three baseline methods to explore their capabilities and limitations. For the EF P2P method, we first implemented the baseline EF model to establish a foundation for comparison. Two subsequent enhancements were developed: replacing the CLIP encoder with LLaMA 3B v2 to improve the richness of text understanding, and upgrading the diffusion backbone to Stable Diffusion 2-1-base for higher fidelity and resolution. While the first enhancement showed a slight decline in performance, the second significantly improved the quality and precision of edits.

The second baseline method we implemented is the MasaCtrl method. MasaCtrl aims to be a *tuning-free* image synthesis method. It makes use of mutual self-attention to combine prompt layout and source image content, ensuring consistent and faithful image generation. In order to improve the performance of the model we examined the use of advanced stable-diffusion architecture such as SD 2.1 and also experimented with the integration of external control mechanisms like ControlNet. The findings here indicate that while MasaCtrl in comparison to the modifications, excels in its original configuration. The modifications to the model architecture and additional control layers lead to a noticeable degradation in performance, suggesting a need for more sophisticated integration methods to maintain effectiveness across varied setups.

The final baseline we implement is the Direct Inversion Method, a foundational approach for generative transformations. To improve its performance, we enhance the richness of text understanding by replacing the CLIP text encoder with the Flava Text Encoder. Flava, with its advanced multimodal capabilities, captures subtle textual nuances and fosters stronger alignment between text prompts and visual outputs. This replacement aims to achieve more accurate and coherent transformations, particularly for complex prompts requiring detailed semantic comprehension.

Finally, to overcome the challenge of long inference times caused by the inversion process in image generation, we implement the Denoising Diffusion Consistency Model (DDCM). This innovative model provides a solution to traditional inversion-based methods by enabling inversion-free image editing. By leveraging this model, we significantly reduce inference times while maintaining or even improving the semantic and visual fidelity of the generated outputs. This approach streamlines the editing process, making it more efficient and scalable for practical applications.

Through these efforts, we aim to enhance text-driven image editing by addressing critical gaps in baseline methods and demonstrating the impact of targeted enhancements. This report presents our methodologies, experimental results, and a detailed evaluation of each approach, contributing to advancements in the integration of vision-language models with generative diffusion frameworks.

2. Related Work

Diffusion models, such as DDIM and Stable Diffusion, have set the benchmark in text-driven image editing, providing tools for mapping from latent spaces to image domains, albeit often struggling with structural fidelity during semantic edits. Innovations like CycleDiffusion and the introduction of EF latent spaces aim to enhance structural consistency and editing flexibility. Concurrently, vision-language models like CLIP, and more recently OpenAI's LLaMA, have improved semantic alignment between image and text embeddings, suggesting potential for deeper semantic integration in diffusion models. This work extends the capabilities of diffusion architectures, such as the advanced Stable Diffusion 2-1-base, which offers greater resolution fidelity and robustness, paving the way for high-precision, high-fidelity image outputs. Our integration of these technologies aims to tackle the enduring challenges of structural fidelity, semantic depth, and visual precision in sophisticated text-driven image editing tasks.

3. Methods

3.1. Edit-Friendly P2P

Method: The EF method serves as one of the baselines for this work, addressing the limitations of traditional DDPM noise spaces in text-driven image editing. Unlike DDIM, which employs deterministic inversion techniques, EF leverages a stochastic inversion process to encode structural information more effectively into the latent noise space. This enables the method to preserve the input image's structure while allowing meaningful semantic modifications guided by text prompts.

The EF method achieves this by constructing a noise space specifically tailored for editing applications. Through the decoupling of dependencies across timesteps, it introduces increased variance in the noise maps, effectively embedding the image structure. This property allows EF to generate artifact-free edits, even in the context of complex semantic transformations, establishing a robust foundation for testing and further exploration.

The step-by-step procedure of the EF method is outlined in Algorithm 1 (see Figure 1). It begins with initializing Gaussian noise and employs stochastic inversion to encode the structural properties of the input image. The latent noise space is iteratively modified using the target prompt, achieving a balance between semantic alignment and structural retention. Finally, the forward diffusion process generates the edited image by progressively refining the adjusted latent space.

The effectiveness of the EF method lies in its capacity to encode both semantic and structural information into the latent space. This dual capability allows it to maintain the input image's key attributes while incorporating guided edits.

Algorithm 1 Edit-friendly DDPM inversion

Input: real image x_0
Output: $\{x_T, z_T, \ldots, z_1\}$
for $t = 1$ **to** T **do**
  $\tilde{\epsilon} \sim \mathcal{N}(0, 1)$
  $x_t \leftarrow \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\tilde{\epsilon}$
end for
for $t = T$ **to** 1 **do**
  $z_t \leftarrow (x_{t-1} - \hat{\mu}_t(x_t))/\sigma_t$
  $x_{t-1} \leftarrow \hat{\mu}_t(x_t) + \sigma_t z_t$ // to avoid error accumulation
end for
Return: $\{x_T, z_T, \ldots, z_1\}$

Figure 1. The Edit-Friendly Algorithm [2]

A key visual representation from the original paper demonstrates its ability to preserve structural details and semantic coherence, showcasing its advantages over baseline models such as DDIM.

Enhancement with LLaMA 3B v2 Encoder: The first enhancement aimed to improve the semantic richness and accuracy of text-to-image mappings by replacing the CLIP encoder with OpenAI's LLaMA 3B v2. While CLIP has been widely used in diffusion models for its robust text-image alignment, it lacks the capacity to fully capture nuanced and context-specific semantics, particularly in detailed prompts. LLaMA, being a state-of-the-art language model, offers richer embeddings and better generalization in text-driven tasks.

LLaMA 3B v2 is designed to handle long and complex prompts with greater linguistic precision, leveraging transformer-based attention mechanisms to encode semantic relationships. By integrating LLaMA, the enhanced EF method aimed to exploit these capabilities for more context-aware edits. The theoretical underpinning lies in LLaMA's ability to generate embeddings that better capture relationships between words, thereby improving alignment in the generative diffusion process.

Upgrading to Stable Diffusion 2-1-base: The second enhancement focused on improving the fidelity and resolution of the generated images by upgrading the diffusion backbone to Stable Diffusion 2-1-base. Earlier diffusion models, including those used in the baseline EF method, often struggled to retain fine details in complex editing scenarios. Stable Diffusion 2-1-base, with its advanced architecture, was selected to address these gaps.

Stable Diffusion 2-1-base builds on the latent diffusion model, which processes images in a compressed latent space rather than pixel space. This enables it to handle high-

resolution images more efficiently while preserving details. The model incorporates an improved noise scheduler and enhanced latent representations, ensuring robustness and consistency in edits.

3.2. MasaCtrl

Method: The MasaCtrl method focuses on image synthesis by integrating spatial layout information from a target prompt with visual content from a source image, leveraging a dual-input strategy to control the composition and maintain visual consistency of the generated image. The process initiates with a deterministic DDIM inversion to create an initial noise map from a source image, **applying a null source prompt when the source is a real image**. This initial noise map establishes the foundation for the denoising process, which is guided by the spatial directives provided by the target prompt.

Central to MasaCtrl's innovative approach is the implementation of a **mask-guided mutual self-attention mechanism**. This mechanism utilizes *dynamically generated masks* which is derived from averaged cross-attention maps that consider both the source prompt (P_s) and the target prompt (P). The masks, M_s and M, are **crucial for delineating** foreground and background regions, ensuring that the self-attention mechanism selectively processes only the relevant areas of the source image. This selective attention allows the queries from the target image's diffusion process to interact exclusively with **pertinent key and value vectors** from the source image, facilitating effective content transfer while adhering to the spatial layout specified by the target prompt. The equations for the mask-guided mutual self-attention mechanism are given by:

The **EDIT function** is instrumental in dynamically regulating the source image content's **impact** during the denoising process, utilizing modified query Q, key K_s, and value V_s matrices. Parameters S (starting denoising step) and L (U-Net layer) are precisely set to determine the transition point for integrating the source image content, with experimental results leading to the selection of $S = 4$ and $L = 10$ for the layers of the U-Net architecture.

The described method leads directly to the specific steps outlined in the algorithm, focusing on how the **mask-guided mutual self-attention mechanism** is operationalized to achieve seamless integration of layout and content. The algorithm's step-by-step approach ensures that each phase of the process is meticulously executed, resulting in a synthesized image that is both structurally aligned and visually coherent with the input specifications.

Adaptation to Stable Diffusion 2-1 The adaptation of MasaCtrl to the *Stable Diffusion 2-1-base* represents an attempt to leverage improved model architectures for enhanced performance. The original design of MasaCtrl was closely integrated with the architecture and training data of *Stable Diffusion v1.4*, which restricted its compatibility with new models. *Stable Diffusion 2-1-base* introduces substantial changes in the U-Net architecture and attention mechanisms, necessitating a careful recalibration of the mutual self-attention mechanism within MasaCtrl.

Algorithm 3 MasaCtrl: Tuning-Free Mutual Self-Attention Control

Input: A source prompt P_s, a modified prompt P, the source and target initial latent noise maps z_s and z_T.
Output: Latent map z_s^0, edited latent map z_0 corresponding to P_s and P.

1. For $t = T, T-1, \ldots, 1$ do
2. $\epsilon_s, \{Q_s, K_s, V_s\} \leftarrow e_\theta(z_t^s, P_s, t)$;
3. $z_{t-1}^s \leftarrow \text{Sample}(z_t^s, \epsilon_s)$;
4. $\{Q, K, V\} \leftarrow e_\theta(z_t, P, t)$;
5. $\{Q^*, K^*, V^*\} \leftarrow \text{EDIT}(\{Q, K, V\}, \{Q_s, K_s, V_s\})$;
6. $\epsilon = e_\theta(z_t, P, t; \{Q^*, K^*, V^*\})$;
7. $z_{t-1} \leftarrow \text{Sample}(z_t, \epsilon)$;
8. end for

Return z_s^0, z_0

Figure 2. The MasaCtrl Algorithm [1]

ControlNet Integration: Integrating ControlNet into MasaCtrl, specifically using the *lllyasviel/sd-controlnet-openpose* model for pose estimation, presents a strategic enhancement aimed at refining control over image transformation processes, particularly in scenarios involving human figures. While ControlNet introduces advanced capabilities for dynamic pose adjustments by leveraging OpenPose, it also adds a new layer of complexity to the image synthesis process. The integration challenges involve ensuring that the mutual self-attention mechanism of MasaCtrl effectively incorporates the detailed pose maps generated by ControlNet. This adaptation is crucial for achieving precise alignment of the same human figures in generated images as present in the source image. This requirement of identification of human figure, human posture and background context and the precision corresponding for this task, pose several challenges in the integration of external layers in MasaCtrl. The integration, requires careful calibration to prevent the disruption of the established content consistency and to manage the added computational complexity effectively.

3.3. Direct Inversion

Method: The core idea of this method is to separate the source and target branches, allowing each branch to operate at its full potential independently. In the source branch, $z_t^* - z_t^{''}$ is directly added to $z_t^{''}$, which is a straightforward approach that effectively corrects the deviation and is compatible with various editing techniques. In the target branch, leaving it unchanged ensures that the diffusion model's capacity for generating the target image is fully utilized. This simple yet efficient solution addresses three challenges in

optimization-based inversion by: (1) eliminating the need for optimization, thus minimizing additional time overhead; (2) adding $z_t^* - z_t^{''}$ which removes the noticeable gap between $z_0^{''}$ and the initial z_0 and (3) ensuring no impact on the input distribution of the diffusion model.

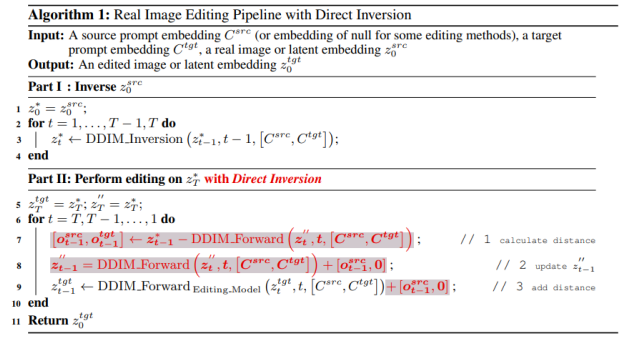

Figure 3. The Direct Inversion Algorithm cite main paper[3]

Algorithm 3 outlines the procedure for integrating **Direct Inversion** into existing diffusion-based image editing methods. The three lines of code added by **Direct Inversion** are highlighted in red with a gray background. Diffusion-based image editing typically involves two steps: an inversion process to obtain the image's representation in diffusion space, and a forward process to apply edits within that space. **Direct Inversion** can be seamlessly integrated into the forward process, gradually correcting the deviation path. Specifically, **Direct Inversion** computes the difference between z_{t-1}^* and $z_{t-1}^{''}$, then adds the difference back to $z_{t-1}^{''}$ during the DDIM forward process. Instead of setting $z_{t-1}^{''} = z_{t-1}^*$, $z_{t-1}^{''}$ is updated with using the difference of the source prompt in latent space, as described in Algorithm 3 line 8. This approach is crucial for preserving the editability of the target prompt's latent space.

Enhancement with Flava Text Encoder : The Flava encoder was incorporated to enhance the multimodal understanding required for precise edits. Flava is a single unified foundation model which can work across vision, language as well as vision-and-language multimodal tasks. Its ability to jointly process textual and visual inputs allows it to generate enriched embeddings that capture nuanced semantic relationships. The aim of this improvement is to ensure better alignment between the textual prompts and the corresponding image edits, particularly in cases involving subtle contextual changes or complex instructions.

Diffusion Latent Consistency Model: The inversion-based editing methods that we looked at until now are limited for real-time and real-world language-driven image editing application. First, most of them still depend on a time consuming inversion process to obtain the inversion branch as a set of anchors. Second, consistency remains a bottleneck given the efforts from optimization and calibration. Third, all current inversion-based methods rely on variations of diffusion sampling, which are incompatible with efficient Consistency Sampling using Latent Consistency Models.

DDCM offers an alternative to address these limitations, by introducing an inversion free paradigm for editing images. While also adopting a dual-branch paradigm, the key of this method is to directly calibrate the initial z_0^{tgt} rather than the z_t^{tgt} along the branch. Algorithm 4 shows the complete pseudcode for the algorithm.

Algorithm 2 DDCM for inversion-free image editing

Input:
Conditional Diffusion/Consistency Model $\varepsilon_\theta(\cdot,\cdot,\cdot)$
Sequence of timesteps $\tau_1 > \tau_2 > \cdots > \tau_{N-1}$
Reference initial input z_0^{src}
Source/target prompts as conditions $c^{\text{src}}, c^{\text{tgt}}$

Sample a random terminal noise $z_{\tau_1}^{\text{src}} = z_{\tau_1}^{\text{tgt}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
$\varepsilon_{\tau_1}^{\text{cons}} = (z_{\tau_1}^{\text{src}} - \sqrt{\alpha_{\tau_1}} z_0^{\text{src}})/\sqrt{1-\alpha_{\tau_1}}$
$\varepsilon_{\tau_1}^{\text{src}}, \varepsilon_{\tau_1}^{\text{tgt}} = \varepsilon_\theta(z_{\tau_1}^{\text{src}}, \tau_1, c^{\text{src}}), \varepsilon_\theta(z_{\tau_1}^{\text{tgt}}, \tau_1, c^{\text{tgt}})$
$z_0^{\text{tgt}} = f_\theta(z_{\tau_1}^{\text{tgt}}, \tau_1, \varepsilon_{\tau_1}^{\text{tgt}} - \varepsilon_{\tau_1}^{\text{src}} + \varepsilon_{\tau_1}^{\text{cons}})$
for $n = 2$ to $N-1$ **do**
Sample noise $\varepsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
① $z_{\tau_n}^{\text{src}} = \sqrt{\alpha_{\tau_n}} z_0^{\text{src}} + \sqrt{1-\alpha_{\tau_n}}\varepsilon$
① $z_{\tau_n}^{\text{tgt}} = \sqrt{\alpha_{\tau_n}} z_0^{\text{tgt}} + \sqrt{1-\alpha_{\tau_n}}\varepsilon$
② $\varepsilon_{\tau_n}^{\text{src}} = \varepsilon_\theta(z_{\tau_n}^{\text{src}}, \tau_n, c^{\text{src}})$
③ $\varepsilon_{\tau_n}^{\text{cons}} = (z_{\tau_n}^{\text{src}} - \sqrt{\alpha_{\tau_n}} z_0^{\text{src}})/\sqrt{1-\alpha_{\tau_n}}$
④* $\varepsilon_{\tau_n}^{\text{tgt}} = \varepsilon_\theta(z_{\tau_n}^{\text{tgt}}, \tau_n, c^{\text{tgt}})$
⑤ $z_0^{\text{tgt}} = f_\theta(z_{\tau_n}^{\text{tgt}}, \tau_n, \varepsilon_{\tau_n}^{\text{tgt}} - \varepsilon_{\tau_n}^{\text{src}} + \varepsilon_{\tau_n}^{\text{cons}})$
end for
Output: z_0^{tgt}
**Vanilla target noise prediction, no attention control.*

Figure 4. The DDCM Algorithm

This method overcomes several key limitations of traditional inversion-based editing techniques. First, by using DDCM sampling, we eliminate the need for the inversion branch anchors that previous methods rely on, significantly reducing computational overhead. Second, while existing dual-branch approaches progressively adjust z_t^{tgt} over time, this method directly refines the predicted initial z_0^{tgt} , avoiding the cumulative errors that typically accumulate during the sampling process. Third, our framework seamlessly integrates with Consistency Sampling via Latent Consistency Models (LCMs), enabling the efficient generation of the target image in just a few steps.

4. Experimental Details

4.1. Edit-Friendly P2P

We implemented the EF P2P method as presented by the original authors. The EF P2P method was carefully replicated following the steps outlined in the original paper. This replication, conducted on Kaggle using GPU P100, ensured

a robust starting point for evaluating subsequent enhancements. For the baseline implementation, we utilized the PIE-Bench dataset, a comprehensive benchmark designed to evaluate image editing tasks across diverse categories. Initially, the EF P2P method was applied to the complete set of 700 images in the dataset to establish its overall performance.

Building on this baseline, we investigated two key enhancements to the EF P2P method. The first enhancement involved replacing the default CLIP encoder with the LLaMA 3B v2 text encoder. This step aimed to improve semantic alignment between input prompts and generated images by leveraging the richer contextual embeddings produced by LLaMA. The second enhancement focused on upgrading the diffusion backbone to Stable Diffusion 2-1-base, targeting improvements in image fidelity and the handling of fine details during the editing process. Due to computational limitations, these enhancements were applied only to a subset of the dataset, specifically the `0_random_140` category, comprising 140 images. To enable a direct comparison, the baseline EF P2P method was also re-evaluated on this subset.

These enhancements were designed to test the method's adaptability to advanced text encoders and diffusion architectures, with results and comparisons presented in the following sections.

4.2. MasaCtrl

We implemented the MasaCtrl as presented by the original authors in the original paper. This task was performed using Google Colab T4 GPU with 15 GB RAM. The baseline experiment was performed on the complete 700 images present in the PIE-Bench dataset. **Adaptation to Stable Diffusion 2.1:** This modification focussed on the stable diffusion backbone. Here we replaced the default SD v1.4 with the SD 2.1 base. This was done with the idea to leverage from the improved noise-reduction and encoding method. This enhancement was performed on the `0_random_140` category. **Integeration of ControlNet:** In our experiment, we integrated ControlNet with MasaCtrl to preserve human features in image edits. This integration targeted images with the sub-category '*human*', using ControlNet's OpenPose to maintain key features like faces and bodies. However, the mutual self-attention mechanism of MasaCtrl often misinterpreted OpenPose's signals, leading to inconsistent outputs. This highlights MasaCtrl's sensitivity to external inputs and indicates a need for significant enhancements in future implementations to handle such complexities effectively.

4.3. Direct Inversion

We first implement the Direct Inversion algorithm as presented by the authors. We carefully replicated the algorithm presented in the paper. All the code is executed on a Kaggle environment using 16 Gb of T4 GPU. For the baseline evaluation we use complete 700 images of the PIE-Bench dataset. To explore the potential benefits of using a more advanced text encoder, we conducted experiments by replacing the default CLIP text encoder in Stable Diffusion 1.4 with the Flava model, an open-source, multi-modal encoder trained on a large corpus of text and images. The idea behind this swap was to leverage Flava's potential for richer and more nuanced text-image alignment, which could lead to improvements in the quality of generated images, particularly in scenarios involving complex or abstract textual prompts. Since we had limitation with the computational resources, we only evaluate this enhancement over a subset of the complete PIE-bench. We use the `0_random_140` category of the dataset, consisting of 140 images with editing prompts.

Diffusion Latent Consistency Model: The algorithm is implemented as presented by the authors []. We implement the algorithm on a laptop with 16GB RAM and 8 GB RTX 3070 laptop GPU. The results of baseline and the enhancement has been provided in the result section.

5. Evaluation Metrics

To evaluate our methods. we use three different Evaluation Metrics. **SSIM** (Structural Similarity Index Measure) checks how similar two images are by comparing brightness, contrast, and structure. A higher SSIM score means the images are more alike, which is great for tasks where preserving original image details is important. **LPIPS** (Learned Perceptual Image Patch Similarity) focuses on how different two images look to the human eye, especially in terms of textures and fine details. Here, a lower LPIPS score means the images are more similar. **CLIP** Similarity measures how well an image matches a given text description or another image in terms of meaning. A higher CLIP similarity score shows better alignment, which is important for generating contextually accurate images.

6. Results

In this section we present the results of the implementation of the baseline and the enhancements aplied to them.

6.1. Edit-Friendly P2P

EF P2P Baseline: The baseline EF method achieved an SSIM score of 0.72, an LPIPS score of 0.07, and a CLIP similarity of 0.21. These scores indicate a strong balance between structural fidelity and semantic alignment. The low LPIPS value suggests minimal perceptual differences between the input and edited images, while the SSIM score highlights the method's capability to preserve the structural

integrity of the input images. The moderate CLIP similarity score shows acceptable alignment between the target prompts and the generated outputs, establishing a reliable baseline for comparison.

Visual examples of the baseline method are shown in Figure 5. In the example, a cat is transformed into a dog while maintaining the surrounding context, such as the wooden chair and the background, demonstrating the structural fidelity achieved.

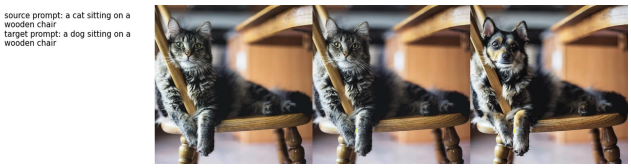

Figure 5. Transformation using EF P2P Baseline

Enhancement 1 - LLaMa 3B v2: The first enhancement, which replaced the CLIP encoder with LLaMA 3B v2, resulted in a decline in performance across all metrics. The SSIM dropped to 0.65, indicating reduced structural fidelity in the generated images. Similarly, the LPIPS increased to 0.14, suggesting that the perceptual differences between input and edited images became more pronounced. The CLIP similarity also slightly decreased to 0.19, reflecting a reduction in the alignment between the target prompts and the outputs.

Visual results in Fig. 6 further support these observations. In the transformation from "a group of pink flowers hanging from a tree" to "a group of red flowers hanging from a tree," the structural consistency is well-maintained, but the color transformation appears subtle, reflecting limited transformation accuracy. Similarly, the shift from "photograph – window of the world by Jimmy Kirk" to "painting – window of the world by Jimmy Kirk" demonstrates semantic understanding, but the style adaptation remains less pronounced. These results indicate that while LLaMA 3B V2 effectively maintains object integrity, it struggles to achieve bold semantic or stylistic edits.

Enhancement 2 - Stable Diffusion 2-1-base: The second enhancement, upgrading the diffusion backbone to Stable Diffusion 2-1-base, showed improvements in performance metrics. The SSIM increased slightly to 0.73, surpassing the baseline and indicating enhanced structural fidelity. The LPIPS value decreased to 0.06, suggesting improved perceptual similarity between input and edited images. Importantly, the CLIP similarity remained consistent with the baseline at 0.21, demonstrating stable semantic alignment.

Visual results in Fig. 7 further demonstrate the efficacy of Stable Diffusion 2-1-base in text-driven edits.

Figure 6. Transformation using EF P2P baseline with LLaMA enhancement

In the transformation of "a meerkat puppy wrapped in a blue towel" to "a lion puppy wrapped in a blue towel," the scene structure is well-preserved, and the generated lion appears realistic and coherent. Similarly, transforming "dogs running in the grass" to "rabbits running in the grass" showcases improved latent encoding capabilities, achieving semantically accurate outputs while maintaining high visual fidelity. The robust attention mechanisms ensure precise and detailed edits without introducing distortions, highlighting the enhancement's ability to preserve fine details and maintain perceptual quality during complex edits. These improvements align with the observed increases in SSIM and decreases in LPIPS scores, emphasizing the stronger semantic alignment and structural consistency achieved.

Figure 7. Transformation using EF P2P baseline with Stable Diffusion 2-1-base enhancement

6.2. MasaCtrl

Baseline Performance Analysis The Baseline Version of MasaCtrl, which is optimized for the Stable Diffusion v1.4 architecture, demonstrates robust performance with an SSIM mean of 0.68 and a standard deviation of 0.132, indicating a high structural similarity to target images. The LPIPS mean of 0.181 and a standard deviation of 0.063 reflect minor perceptual differences, and a CLIP similarity mean of 24.235 with a standard deviation of 3.842 shows

Figure 8. Comparison between MasaCtrl Baseline output and MasaCtrl with SD 2.1

Figure 9. Comparison between MasaCtrl Baseline output and MasaCtrl with ControlNet

good semantic alignment with the target image content. These metrics bring out the effectiveness of the mutual self-attention mechanism.

Impact of Model Architecture Changes Transitioning to the SD 2.1 Version results in decreased performance: the SSIM mean drops to 0.564 with increased variability (standard deviation of 0.157), meaning a reduction in structural similarity. This degradation can be attributed to the architectural differences between Stable Diffusion v1.4 and v2.1, particularly the ineffectiveness in the integration with the new improved U-Net architecture, attention mechanisms, and feature representations. The perceptual difference, as indicated by LPIPS, increases to a mean of 0.417 with a standard deviation of 0.108, reflecting this modification's decreased ability to maintain perceptual consistency. Furthermore, the CLIP similarity decreases to a mean of 22.829, with a slightly higher standard deviation of 3.893, indicating a decline in semantic alignment with the target images. This suggests that the feature representations learned by v2.1 do not align effectively with the features like those learned by v1.4, reducing the semantic coherence of generated images.

Effects of ControlNet Integration The integration of ControlNet further complicates the performance. The SSIM drastically reduces to a mean of 0.3 with a standard deviation of 0.14, indicating significant structural discrepancies. This external control led to disruption of the balance that MasaCtrl aims to maintain between layout and target. Perceptual quality, as measured by LPIPS, also worsens, showing a mean of 0.554 and a standard deviation of 0.132. Additionally, the CLIP similarity to target images decreases sharply to a mean of 17.772 with the highest variability (standard deviation of 4.43), again reflecting poor semantic coherence.

6.3. Direct Inversion

Direct Inversion Baseline We implemented the Direct Inversion Method as a baseline as presented by the authors. We achieve the following scores, 0.054, 0.85 and 0.253 , for LPIPS, SSIM and CLIP similarity, respectively. A score of 0.054 demonstrates excellent perceptual similarity, suggesting that the generated images capture low-level visual details and textures closely aligned with the target images. This highlights the method's effectiveness in preserving pixel-wise features. A score of 0.85 indicates a high level of structural similarity. This suggests that the method successfully reconstructs the overall layout and spatial details of the target images, preserving their global structure. However, some deviations from perfect similarity may arise from subtle distortions or artifacts. A score of 0.253 indicates moderate semantic alignment, suggesting that while the generated images retain some level of contextual relevance, the method struggles to fully encapsulate high-level semantic features. This highlights potential shortcomings in capturing global content or semantic essence.

Enhancement with Flava Text Encoder: With the Flava Text Encoder, we achieved scores of 0.199 (LPIPS), 0.72 (SSIM), and 0.25 (CLIP). The higher LPIPS score indicates a drop in perceptual quality, likely due to artifacts or deviations in low-level features. The lower SSIM score suggests reduced structural similarity, as fine-grained details are lost. The CLIP score remains similar to the baseline, showing minimal improvement in semantic alignment. While Flava maintains semantic coherence, it introduces changes that may favor generalization or other properties not captured by LPIPS or SSIM.

Figure 10 shows the comparison of the edit results of Direct Inversion Baseline and enhanced with Flava Text Encoder. In the first prompt, the output of the Direct Inversion aligns with the target prompt, but the generated origami birds have distortions in texture and lack clarity in shape. The background remains consistent with the original. With Flava enhancement, the origami birds have a more distinct texture and sharpness, making them closer to

the target prompt. However, some color artifacts appear in the surroundings.From the other prompts, we observe that the FLAVA enhancement introduces a more animated and stylized quality to the outputs, suggesting its potential suitability for applications such as cartoon editing or artistic transformations.

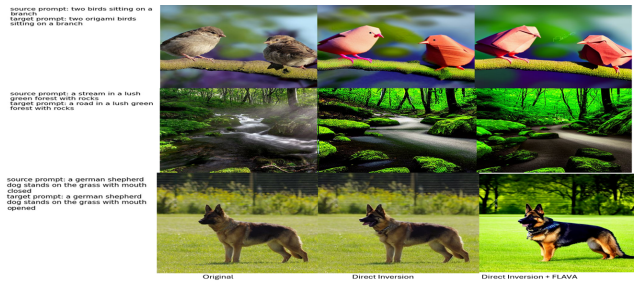

Figure 10. Edit Comparison using Direct Inversion Baseline and enhanced with Flava text encoder.

Enhancement with Diffusion Latent Consistency Model The diffusion Latent Consistency Model achieved an mean SSIM score of 0.843, LPIPS score of 0.073, and a normalized CLIP score of 0.24. The SSIM suggests that the generated images are quite similar to the reference images, with high structural fidelity. The low LPIPS means that the generated images are perceptually quite similar to the ground truth images in terms of high-level visual features, as understood by deep neural networks. Figure 11 presents a comparison between the Direct Inversion Method and the InfEdit (Diffusion Latent Consistency Model) for image editing. It is evident that the InfEdit method significantly outperforms Direct Inversion in terms of editing quality.

In panel (b), the task is to turn [smoke] into [fire]. Direct Inversion makes the edit, but the result is subtle and unclear, while InfEdit creates a vivid transformation with fire clearly surrounding the building. In panel (c), the goal is to change a [laughing] face to an [angry] face. Direct Inversion falls short, making only minor changes like a slightly redder face without capturing anger. InfEdit, however, nails the expression, delivering a clear and high-quality edit. Overall, InfEdit outperforms Direct Inversion with more accurate and effective transformations.

We also analyze the runtime of the baseline Direct Inversion Method and the Denoising Diffusion Latent Consistency Model (DDCM). The results highlight the superior efficiency of the Consistency model, which significantly outperforms the Direct Inversion method in terms of inference time. This is due to the elimination of the time-consuming inversion process in the DDCM. On a 16 GB RTX 3070 Laptop GPU, the Direct Inversion method takes approximately 20 minutes for a single inference, whereas the DDCM completes an inference in just 30 seconds. This substantial reduction in runtime underscores the efficiency and practicality of the Consistency model for real-time applications.

Figure 11. Edit Comparison using Direct Inversion and Diffusion Consistency Model.

7. Conclusion and Future Work

In this project, we implemented and evaluated three baseline models: EF, MasaCtrl, and DI, along with various enhancements to improve performance. While Stable Diffusion 2-1-base in EF method showed notable improvements in structural fidelity and perceptual quality, and MasaCtrl exhibited dependencies on specific architectures like SD1.4, the Consistency Model outperformed all others in terms of editing quality and inference time. This highlights its potential as a robust solution for efficient and high-quality text-driven image editing.

8. Contribution

Introduction and Conclusion is written equally by all the three members of the group. Individual responsibilities and work is mentioned below.

Saaransh Implemented `Edit-Friendly P2P` and its two enhancements, and wrote the related sections.

Shubhan Implemented `MasaCtrl` and its two enhancements, and wrote the related sections.

Aakash Implemented `Direct Inversion P2P`, its enhancement, and `Diffusion Latent Consistency Model (DDCM)`, and wrote the related sections.

References

[1] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing, 2023. 3

[2] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An Edit Friendly DDPM Noise Space: Inversion and Manipulations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12469–12478, 2024. 2

[3] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Direct inversion: Boosting diffusion-based editing with 3 lines of code. *arXiv preprint arXiv:2310.01506*, 2023. 4