One Prompt Diffusion

by

Vaibhav Goyal

(Under the Direction of Jin Sun)

Abstract

Diffusion models have made their mark in image synthesis by excelling in visual quality and flexibility. They use additional negative prompts with classifier-free guidance (CFG), which guides the model in generating images aligned with the user's intent. However, CFG mandates the model to run twice, making it difficult to interpret the impact of the negative prompt in the final image. My study proposes a method to generate a single prompt producing on-par quality as the two prompts/passes CFG. I have prepared a prompt-to-image dataset, used per-image optimization to find the ground truth single merged prompt for each image, and trained a neural network module to predict the embedding of that prompt. During inference time, my model generates a single prompt with a single diffusion pass, achieving up to 2x speedup and 20% memory reduction. My research contributes to developing more efficient diffusion models and deeply understanding their characteristics.

Index words: [Image Generation, Diffusion Models, Prompt Engineering]

One Prompt Diffusion

by

Vaibhav Goyal

B.E. Information Technology, University of Mumbai, India, May 2018

A Thesis Submitted to the Graduate Faculty of the

University of Georgia in Partial Fulfillment of the Requirements for the Degree.

Master of Science

Athens, Georgia

2024

One Prompt Diffusion

by

Vaibhav Goyal

Major Professor: Jin Sun

Committee: Lakshmish Ramaswamy

Wei Niu

Electronic Version Approved:

Ron Walcott

Dean of the Graduate School

The University of Georgia

August 2024

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my committee chairman, Dr Jin Sun, for his constant guidance and support, which has made this research meaningful. His persistent encouragement has helped me navigate through the challenges and I am thankful for his patience and faith in my ability to achieve the results in my study. I am particularly indebted to him for providing the tools and knowledge required to effectively conduct my study, through his courses "Advanced Topics in Computer Vision" and "Advanced Representation Learning," which sparked my interest in stable diffusion models and inspired me to pursue research in this domain. Without his unwavering support, this thesis would not have been possible.

I would also like to thank my committee members, Dr Lakshmish Ramaswamy and Dr Wei Niu, for granting me the freedom to pursue my research interests and for their trust in my ability to excel. Furthermore, I would like to thank the University of Georgia for allowing me to pursue my Master's at their esteemed institution. As an international graduate traveling outside my home country for the first time, the University provided me with the necessary resources required to become an educated and confident individual. I also appreciate the financial support provided to me by the University, which has enabled me to conduct my research with an open mind.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Synthesizing photorealistic images is fundamental and challenging in computer graphics and computer vision. Recently, diffusion-based models such as Stable Diffusion [Rombach et al., 2022b] have been dominating the visual content creation world by producing impressive, high-quality images generated from text prompts. It outperforms the prior arts, such as the Generative Adversarial Networks (GANs) [Karras et al., 2021] by visual quality, flexibility, and ease of training. Classifier Free Guidance (CFG) is a key component for better controlling the content and improving the quality: it creates an additional prompt based on a user's input. It feeds both prompts into the text-guided generation network for multiple time steps. While CFG greatly improves the generated image, having an additional prompt (thus forward pass) is undesirable due to the increased running cost, especially considering a typical diffusion model runs ≥ 50 time steps. CFG is also used as so-called *negative prompts*–objects or properties that the user does not want to exist in an image. Yet, the effects of negative prompts are hard to decode from the final image. For example, the popular approach DAAM [Tang et al., 2022] that visualizes the effective attention of prompt words, cannot work on negative prompts. Taking a step back, an intriguing question arises - whether one should have two prompts in a CFG setting for diffusion models. Is it possible to have one single prompt that can enjoy the benefits of having CFG without losing efficiency and quality?

In this work, I present *One Prompt Diffusion*. This add-on network module can transform multiple prompts into one unified prompt and be fed into the regular stable diffusion UNet structure. OPD is achieved by a learning-based approach: first constructing a prompt-to-image dataset, then performing per-image optimization to find the unified prompt, and lastly, training the OPD module to predict the unified prompt given input prompts. During inference time, stable diffusion models equipped with

OPD only need to be run once, yet they can produce similar results as CFG-enabled diffusion models in improved quality or removal of undesired objects. My research is orthogonal to approaches that speed up the original stable diffusion, such as LCM [Luo et al., 2023] and InstaFlow [Liu et al., 2023]. Thus, the efficiency of those methods can be further improved by incorporating our approach.
My contributions are two-fold:

- I propose a novel learning-based approach to replace multi-prompt CFG in diffusion models with one prompt without losing visual quality in the generated images.
- I construct a prompt-to-image dataset with complex prompts with different negative prompt settings: object removal and quality improvement. I also provide the inversion latents and embeddings for all generated images. My dataset will be useful for general research in text-based generation diffusion models.

The rest of this report is structured as follows: Chapter 2 reviews the literature survey, highlighting some of the advancements and limitations of existing models. Chapter 3 briefly introduces the relevant concepts, such as CFG and negative prompts. Chapter 4 describes the proposed method OPD, including the architecture of the learnable module and the detailed process for creating the optimized prompts. Chapter 5 and 6 details our experimental setup, results, and comparative analysis with standard CFG results. Finally, I conclude with the importance of our research findings and directions for future work.

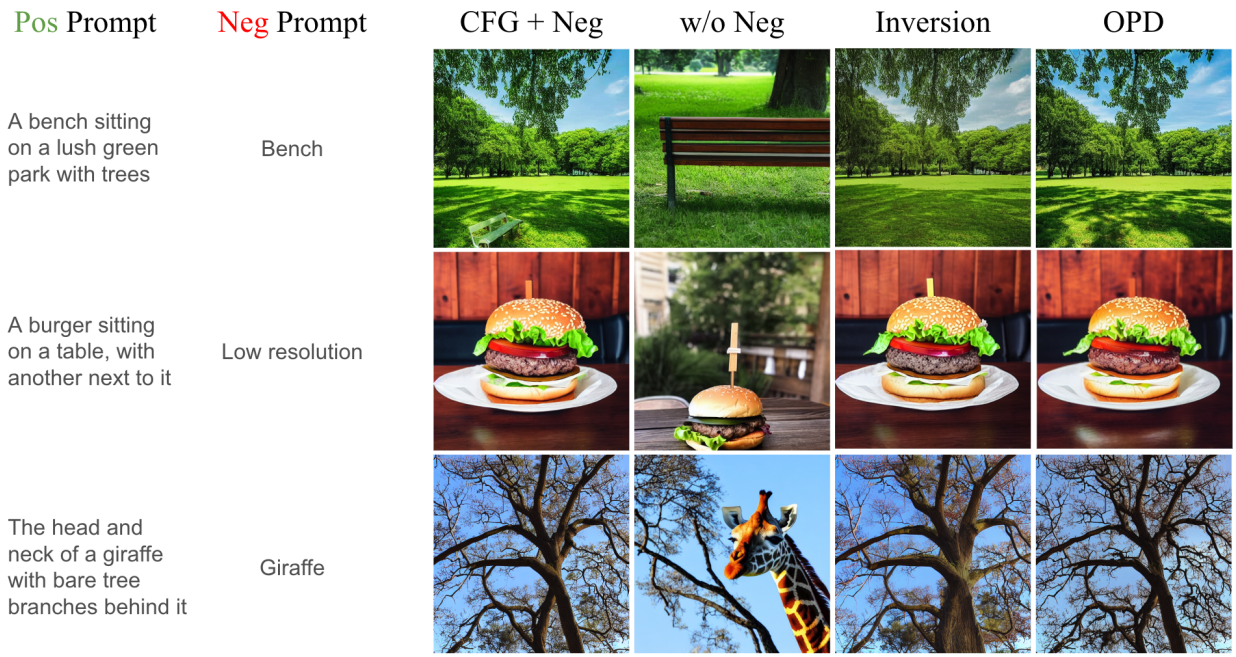


Figure 1.1: Stable diffusion models utilize Classifier-Free-Guidance (CFG) and negative prompts to improve image quality. However, those techniques introduce additional overhead and complexity to the process. I propose *One Prompt Diffusion*: by learning from the *Inversion* of the original *CFG+Neg*ative prompt result, my *OPD* result achieves high image quality (unlike *w/o Neg*) and greatly reduce the computational cost.

CHAPTER 2

LITERATURE SURVEY

2.1 Text-to-image diffusion models

My work builds on the recent foundational work in high-quality text-to-image synthesis through diffusion models [Ho et al., 2020; Ramesh et al., 2022; Rombach et al., 2022b; Saharia et al., 2022; J. Song et al., 2020]. These studies established the groundwork for high-fidelity image synthesis from textual prompts. While Denoising Diffusion Probabilistic Models (DDPM) laid the foundation by introducing a noise-removal process [Ho et al., 2020], Denoising Diffusion Implicit Models (DDIM) built on it by streamlining the sampling process, making it faster and more efficient [J. Song et al., 2020]. The latent diffusion models reduced complexity without compromising quality by working in latent space [Saharia et al., 2022]. Furthermore, by combining text and image latents, overall accuracy and versatility improved for generated images [Ramesh et al., 2022]. My methodology seeks to further improve these models by simplifying the prompt guidance procedure, thereby enhancing the synthesis process's efficiency and accessibility.

2.2 Efficient control of image content with prompts

The ability to control the content of generated images using textual prompts has been explored in various works [Gal et al., 2022; Li et al., 2024; Oppenlaender, 2023]. Prompt taxonomy presents a comprehensive taxonomy study to understand prompt engineering by introducing six types of prompt modifiers directly related to controlling image generation via modifier input [Oppenlaender, 2023]. While insightful, it primarily serves as a guideline and does not provide direct methodologies for practical implementation. The unwanted content can be suppressed by adjusting text embeddings using soft-weighted regularization,

addressing the challenges of specific content control in text-to-image diffusion models [Li et al., 2024]. Despite its effectiveness, this method can be difficult to implement due to fine-tuning text embeddings for each specific instance. Moreover, textual inversion customizes text-to-image models to generate novel scenes of user-provided concepts that are guided by natural language [Gal et al., 2022].

Several notable works in the field of image editing have directly manipulated text embeddings or prompts for style transfer without changing the semantic setting [Kawar et al., 2023; Kim et al., 2022; Meiri et al., 2023; Mokady et al., 2023; Sun et al., 2024; Wallace et al., 2023; Wen et al., 2024; Wu et al., 2023]. One crucial aspect of quality image editing is the accurate inversion or reconstruction of the original image. Null-text Inversion [Mokady et al., 2023] is a text-guided image editing method using diffusion models. It optimizes unconditional embeddings for near-identical reconstructions, which is then used for text-based meaningful modifications without model tuning. While this method achieves high accuracy in image edits, the optimization process for image inversion is very resource and time-intensive, which is not ideal for practical applications. It may also not generate the accurate reconstruction of the input image, resulting in faulty edits at later stages. Similarly, the fixed-point equation introduces a method for efficient and accurate inversion of text-to-image diffusion models. It uses fixed point iterations to solve the inversion problem, significantly reducing computational time while maintaining high fidelity in image reconstruction [Meiri et al., 2023]. Building on this, EDICT [Wallace et al., 2023] is an inversion method, influenced by affine coupling layers, by maintaining two coupled noise vectors that invert each other, alternatively achieving identical inversion between real and model-generated images.

Further enhancing the control over image generation, hard prompts [Wen et al., 2024] introduce a gradient-based discrete optimization approach to generate a sequence of interpretable and discrete tokens to guide the behavior of generative models for tasks across different models. This method enhances reusability for image and language tasks across different models. However, the optimization process is computationally intensive and may not always achieve the desired level of content control without significant manual intervention. On the other hand, Imagic [Kawar et al., 2023] is used for complex text-based semantic edits on an input image, leveraging a pre-trained model and optimizing text embedding that

aligns with both the input and target image. This approach is effective for intricate modification but requires extensive resource optimization and fine-tuning to achieve quality edits. Furthermore, some research has explored the capability of diffusion models to disentangle attributes by partially modifying text embeddings, enabling style changes without altering the image's semantic content [Wu et al., 2023]. It also proposes an optimization framework that combines text embeddings to control the level of disentanglement for style matching and content preservation. Although effective, partial modification can sometimes produce unexpected results and be difficult to manage, requiring careful consideration to avoid unintended alterations in the image. [Sun et al., 2024] explores incorporating spatial information to control the image generation process. It introduces spatial-aware latent initialization to enhance layout controllability. This method remarkably improves spatial adherence by leveraging spatial-aware initialization noise during denoising. However, it does not address broader thematic and content control aspects.

While these methods address the need for content specificity and image editing, they often involve complex interactions with the model and extensive optimization processes, which make them difficult to use. Our work simplifies the image generation process by employing a prediction network, eliminating the need for constant optimization to generate high-quality images.

2.3 Time and memory efficiency in diffusion models

Improving the computational efficiency of image editing and generation processes has been a significant focus, and some strides have been made in achieving this vision [Liu et al., 2022, 2023; Luo et al., 2023; Miyake et al., 2023]. Inspired by null-text inversion, negative-prompt inversion [Miyake et al., 2023] accelerates image editing by bypassing the time and resource-intensive optimization loop required in null-text inversion. Although their approach speeds up the image editing workflow, it primarily focuses on the editing aspect rather than the initial generation process.

Some recent works [Luo et al., 2023; Meng et al., 2023; Salimans and Ho, 2022; Y. Song et al., 2023] have shown that we can generate impressive images in fewer steps by using knowledge distillation on pre-trained diffusion models. Notably, LCM uses one-stage guided distillation, which distills pre-trained

LDMs, such as stable diffusion, to generate high-quality images in just a few steps. However, these don't perform well with one-step generation [Luo et al., 2023].

Meanwhile, Rectified Flow [Liu et al., 2022] learns an ordinary differential equation (ODE) to connect two distributions in a straight path, generating images in a few steps. Moreover, Insta-Flow uses the Rectified Flow [Liu et al., 2023] technique to generate high-quality images in a single step, overcoming the shortcomings of knowledge distillation methods. This significantly accelerates the image synthesis process by achieving high fidelity in just a few steps and reduces computational costs as well.

Since these works use a standard stable diffusion pipeline, they still add overhead for computing resources and time due to CFG. Our approach builds upon this premise by introducing a one-prompt strategy, simplifying the process and potentially enhancing user experience. This method streamlines the prompt generation process and achieves high-quality image synthesis with improved speed and memory usage efficiency with CFG.

CHAPTER 3

BACKGROUND

3.1 Generative Models

Generative models are a family of machine learning models that can generate new data samples from various modalities such as text, image, and video. They are essential in many applications, including text generation and image synthesis. The most common generative models are Generative Adversarial Networks (GANs) [Goodfellow et al., 2014], Variational Autoencoders (VAEs) [Kingma and Welling, 2013], and Diffusion Models [Ho et al., 2020; Rombach et al., 2022a].

GANs consist of two networks, a generator, and a discriminator, competing with each other to produce realistic data samples. VAEs encode input data into a latent space and decode it to generate new data samples. Unlike GANs and VAEs, Diffusion models gradually transform random noise into data samples through denoising steps, making them very effective in producing high-quality images.

3.2 Diffusion models

Diffusion models as a class of generative models, have established a new frontier in visual content creation by learning a reverse diffusion process. There are several types of diffusion models developed since its inception [Weng, 2021]. Denoising Diffusion Probalistic Models (DDPMs)[Ho et al., 2020] slowly adds Gaussian noise to data and then train a neural network to denoise it. Score-Based Generative Models [Y. Song et al., 2020] estimate the gradient score of the log data density with Stochastic Differential Equations and generate samples by following this score. Latent Diffusion Models [LDMs] [Rombach et al., 2022a] generate high-fidelity images from noise by repeatedly removing predicted noise from user-provided text prompts.

For my research, I use the LDMs for their ability to perform diffusion in latent space, significantly reducing the computational burden. Unlike other diffusion models which process data in its original high-dimensional form, LDMs work with lower-dimensional latent space. This helps in generating images with lower computational cost and memory usage, without impacting the quality of the images.

3.3 Stable Diffusion Model

The stable diffusion model is a latent text-to-image diffusion model generating photo-realistic images with textual descriptions. Text prompts are encoded in tokens (often with an encoder model like CLIP [Radford et al., 2021]) and interact with image features in a UNet structure via cross-attention. A high-quality diffusion model typically requires the usage of two separate model passes, each with a different prompt: one pass given the input encoded text and another pass given noise, i.e., conditional & unconditional passes. This CFG is a measure of robustness as image synthesis with the conditional pass alone produces inconsistent and visual artifact-laden images [Ho and Salimans, 2022]. The interplay of the two prompts is controlled by the guidance scale w:

$$\tilde{\epsilon}(Z, x_{\text{inp}}; w) = \epsilon(Z, x_{\text{emp}}) + w(\epsilon(Z, x_{\text{inp}}) - \epsilon(Z, x_{\text{emp}})), \tag{3.1}$$

where Z is the latent vector, w is guidance-scale, x_{inp} is the embedding of the input prompt, x_{emp} is the embedding of an empty string prompt, ϵ is the putout from the diffusion UNet, and $\tilde{\epsilon}$ is the transformed representation of the UNet output created by CFG with the two prompts [1]. We aim to produce results highly similar to $\tilde{\epsilon}$ but without CFG.

3.4 Negative prompts

Negative prompts, which we refer to as x_{neg}, are additional inputs in text-to-image tasks when the user explicitly outlines things that should not appear in the generated image. A common use is to guide the model to produce higher-quality images by using words like “distorted”, “blurry”, and “fuzzy” in the

[1] Ho and Salimans CFG equation does not include x explicitly. We show it here to describe negative prompting, similar to better [Rombach et al., 2022a].

negative prompt. We call these *quality attributes*. Another way of using it is to prevent a certain object in the generation. For example, if the input prompt is "a plate full of food" and the negative prompt is "fork", the generated image should contain a plate with food but no forks.

CFG can be used to incorporate negative prompts in a similar way:

$$\tilde{\epsilon}(Z, x_{\text{inp}}; w) = \epsilon(Z, x_{\text{neg}}) + w(\epsilon(Z, x_{\text{inp}}) - \epsilon(Z, x_{\text{neg}})), \tag{3.2}$$

This turns the guidance scale into a weight between the representation of the negative prompt and the difference between the negative and conditional, referred to as *positive prompt*. Negative prompts could have unexpected results. For instance, a negative prompt can fail to remove obvious objects like in Fig:3.1, or it removes the most semantically meaningful part of the object. We study the effect of negative prompts in building our dataset in Section 4.1.

CFG is crucial not only for removing unwanted content guided by negative prompts but also for generating relevant images without negative prompts. For example, an image generated only with a positive prompt and without CFG lacks coherence with its textual description, as shown in Fig: 3.2. However, the quality significantly improves with CFG. When a negative prompt is not provided, the stable diffusion pipeline automatically generates an empty set of unconditional embeddings to guide the model in creating an appropriate image using CFG.

3.5 COCO Dataset

Common Objects in Context (COCO) dataset is a large-scale dataset widely used in computer vision research. It contains over 330k images with more than 200k labeled instances and 80 object categories. The dataset includes annotations for each image for object detection, segmentation, and scene understanding. The images are diverse and represent objects in complex scenes, providing a rich resource for training and evaluating models.

In the context of my research, I use COCO dataset to generate crucial datasets to train and evaluate my MLP models. Especially, I use the captions and object segmentation's to get a set of positive and negative

prompts which are then fed into the stable diffusion models to generate new samples of images to use for further training.

3.6 Null-Text Inversion

Null-text inversion [Mokady et al., 2023] is a technique for text-based image editing using a Stable diffusion model. The distinction with other image editing techniques is that it does not modify the input text embedding, but rather optimizes the unconditional text embedding used in classifier-free guidance. This facilitates text-based editing without changing model weights or conditional embedding. The optimized embedding is used in image inversion in latent space which is then used for direct image editing.

My research takes inspiration from this approach to optimize one single merged embedding which can generate a similar quality image without using classifier-free guidance. This technique is especially beneficial for its use of coco images for optimization and the stable diffusion model for reconstruction. The modified algorithm is discussed in detail in Sec. 4.2.

Figure 3.1: Original stable diffusion (CFG + negative prompt) could fail to generate the right image. Positive prompt: "black and white photo of a bus and big ben". Negative prompt: "bus". Left: CFG w/o negative prompt, Right: CFG w/ negative prompt. Both still contain the bus.

Figure 3.2: Original stable diffusion (Positive Prompt - CFG) does not generate a meaningful image. Positive prompt: “People riding bicycles down the road approaching a bird”. Negative prompt: “”. Left: w/o CFG and negative prompt, Right: CFG and w/o negative prompt. Quality is significantly improved in the right image.

CHAPTER 4

ONE PROMPT DIFFUSION

In this chapter, I describe in detail the design of One Prompt Diffusion: the construction of the prompt-to-image dataset featuring negative prompt designs, the inversion of the generated images, and the learning formulation of the OPD module.

4.1 Prompt-to-image Dataset

I utilize the COCO dataset [Lin et al., 2014] as the source of rich text prompts. The dataset is a vast collection of images and associated captions. For this research study, I use the Coco2017 version of the dataset which consists of around 121k images including training and validation datasets. In particular, I use the COCO captions associated with each image as the basis for the positive prompts.
There are several processing steps involved:

4.1.1 Identify Dominating Object

I use ground truth segmentation masks in COCO to find the largest object covering more than 75 pixels of an image and assign it as the dominating object. If no object mask meets this criterion, I discard the corresponding COCO caption. This step ensures that I consider only the focus objects, which are essential for analyzing image alterations with or without the use of CFG (Classifier-Free Guidance), for our research. This step is necessary for all attribute types as highlighted in Table: 1.

4.1.2 Quality-attribute negative prompt

I use keywords that are widely used in the digital art community, including blurry, overexposed, distorted, and fuzzy. These are undesirable attributes that users do not want the final images to have. Other negative

prompts, such as "bad anatomy," are specific to generating humans. I have not used them as the focus is on a more general setting.

4.1.3 Object-removal negative prompt

I carefully design the protocols to obtain negative prompts for object-removal effects. One thing worth noting is that if we do not contain the object in the positive prompt, the negative prompt might not be necessary. For example, an image generated by the positive prompt of "a plate full of food" might not contain any forks, even without the negative prompt being "fork". Therefore, to study the real effect of negative prompt in object removal, we use the object's name in *both* positive and negative prompts.

I use the original COCO caption (containing the object name) as the positive prompt and the dominant object as the negative prompt. To ensure that the dominant object is referenced in the positive prompt, I filter out any positive prompts that do not include the negative prompt word.

4.1.4 Image synthesis

For each positive prompt, I prepare positive and negative prompts as mentioned in previous steps and use Stable Diffusion 2.1 model[1] with consistent random seeds to generate different versions of images. Using CFG, I generate two images for each candidate's positive and negative prompt pair, including the (prompt-only) and the (prompt, negative prompt) pair. I use the default settings, such as the number of time steps provided by the Huggingface documentation, to generate the images.

4.1.5 Postprocessing filtering

As shown in Figure 3.1, the generation of ground truth stable diffusion and CFG results could fail to respect the negative prompts. We need additional postprocessing steps to ensure the quality of the learning dataset and reduce false positives.

I filter the dataset to retain images with desirable attributes. I start with checking if the image generated with just the positive prompt has the negative prompt object in it with a minimum pixel area of 15% of

[1] https://huggingface.co/stabilityai/stable-diffusion-2-1

the total area of the image. Then, I check if this object is absent in the image generated with positive and negative prompt pair. Table 1 shows the statistics of images after each filtering step in building our dataset.

My data and the software tools developed will be released to the public to promote further research, as there is no existing dataset featuring the design of negative prompts.

Table 1: The number of instances passed each dataset filtering level.

Filtering Step	**Instances Remaining**		
	Object	**Quality**	**Empty**
Total Images	69000	54000	4000
Dominate objects	67972	53557	3976
Object Presence	22685	N/A	N/A
Object Removal	13162	N/A	N/A
MTO Optimization	8701	11640	1397

4.2 Merged-Text Optimization (MTO)

Looking at Equations 3.1 and 3.2, we should be able to reduce the need for multiple prompts (thus multiple forward passes) if we can obtain another text embedding x_{merged} so that $\epsilon(Z, x_{\text{merged}})$ is identical to that obtained from CFG. Formally, we are looking for the optimal merged embedding for each Z and x_{inp}:

$$x_{\text{merged}} =_x \|\tilde{\epsilon}(Z, x_{\text{inp}}; w) - \epsilon(Z, x)\|. \tag{4.1}$$

Once we obtain the merged embedding x_{merged}, it can be used to generate an identical image containing the full information of the positive and negative prompts in the original setting. One of the challenges is to optimize for x_{merged}, which is a well-studied problem known as the *text inversion* task in generative models.

I leverage the widely-used Null-Text Optimization (NTO) algorithm [Mokady et al., 2023] as the inversion method for our purpose. NTO uses techniques like Denoising Diffusion Implicit Models (DDIM) [J. Song et al., 2020] to generate step-latent for all time steps iteratively. Starting with initial "null-text" embeddings and the initial step-latent obtained from DDIM, NTO optimizes the embeddings in multiple iterations by minimizing loss between the previous step latent from DDIP loop and previous

step latent obtained from the predicted noise generated using Equation 3.1. Please find the full details of the original NTO algorithm in [Mokady et al., 2023].

I have designed a modified version of NTO called Merged-Text Optimization (MTO) to obtain x_{merged}. MTO starts with randomly initialized x_{merged} and the initial step-latent obtained from the DDIM loop. It iteratively optimizes the embeddings by minimizing loss between ground truth, previous step-latent, and previous latent obtained from predicted noises. Instead of using a unique x_{merged} for each time step as in the original NTO, I found that the information stored in the last x_{merged} is sufficient to generate a close-enough image without using CFG. The optimized embedding x_{merged} along with the initial latent are then used to generate the inverted image with a single diffusion model forward pass for verification. I use LPIPS [Zhang et al., 2018] as the metric to measure visual similarity between the inverted image and the original one generated by CFG. I use a threshold of 0.25 to only keep inverted images below it. Table 1 summarizes the data I have obtained using MTO.

Algorithm 1 Merged-Text Optimization (MTO)

Require: Prompt embedding x_{inp}, x_{neg}, and generated image I.
Ensure: Optimized embeddings $\{x_t\}_{t=1}^{T}$ and the latent vector Z_T.

Compute the intermediate latents $z_T^*, \ldots, z_0^*$ using DDIM inversion over I;
Initialize $\tilde{Z}_T \leftarrow Z_T^*, x_T \leftarrow x_{\text{neg}}$;

for $t = T, T-1, \ldots, 1$ **do**
    for $j = 0, \ldots, N-1$ **do**
        $x_t \leftarrow x_t - \nabla_x \left\| Z_t^* - Z_{t-1}(\tilde{Z}_t, t, x_t) \right\|_2^2$
    end for
    Set $\tilde{Z}_{t-1} \leftarrow Z_{t-1}(\tilde{z}_t, x_t, x_{\text{inp}}), x_{t-1} \leftarrow x_t$
end for
return $\{x_t\}_{t=1}^{T}, \tilde{Z}_T$

4.3 Merged-Text Prediction (MTP)

MTO performs well in achieving high-fidelity images as the original CFG ones, but the optimization process takes ∼150 seconds and a large chunk of GPU memory to optimize one image, which is a significant amount of overhead preventing any practical use. To tackle the challenge, I propose a learnable module,

Merged-Text *Prediction*, that is trained to approximate the output from MTO but with fast feedforward inference time performance. As later presented in the experiment section, this straightforward approach is effective.

The learnable module is an MLP consisting of four linear layers and non-linear activation in between Fig. 4.1. The simplicity of the model is designed to have efficiency in the design. The model takes the original positive prompt embedding x_{inp} as well as the additional prompt embedding x_{emp} or x_{neg} together as the input and outputs the one embedding. The learning is formulated as a standard regression problem so that the model's output is close to x_{merged} obtained by MTO measured by typical regression loss (MSE). With a trained MTP model incorporated in a standard stable diffusion pipeline between the text encoder and input to the UNet, we can achieve *One Prompt Diffusion*.

The MLP model used in this study consists of multiple linear layers intertwined with GELU activation layers, comprising nearly 2 million trainable parameters as seen in Fig. 4.2. The architecture includes layers with 512 and 768 output features, resulting in a total model size of approximately 9.48 MB, demonstrating its simplicity and efficiency. The optimization process employs the Adam optimizer with an initial learning rate 1e-2, which decreases proportionally with epoch progression. The model's efficacy in training is evidenced by its convergence after 7 epochs of non-improving mean validation loss.

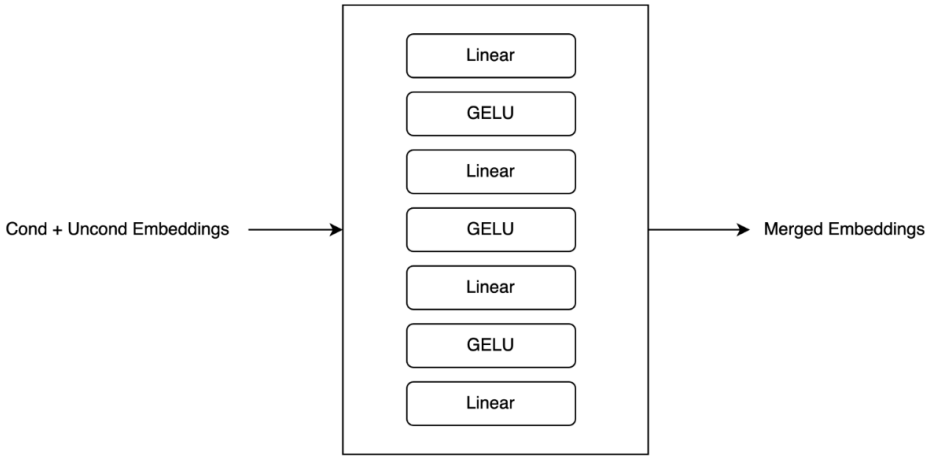


Figure 4.1: Merged-Text Prediction Model - MLP model consisting of stacked linear and activation layers. The model accepts two sets of embeddings passing through multiple layers and producing one single merge embedding as output

```
==========================================================================
Layer (type:depth-idx)          Output Shape              Param #
==========================================================================
Linear: 1-1                    [-1, 1, 77, 512]          786,944
GELU  : 1-2                    [-1, 1, 77, 512]            --
Linear: 1-3                    [-1, 1, 77, 768]          393,984
GELU  : 1-4                    [-1, 1, 77, 768]            --
Linear: 1-5                    [-1, 1, 77, 512]          393,728
GELU  : 1-6                    [-1, 1, 77, 512]            --
Linear: 1-7                    [-1, 1, 77, 768]          393,984
==========================================================================
Total params: 1,968,640
Trainable params: 1,968,640
Non-trainable params: 0
Total mult-adds (M): 1.97
==========================================================================
Input size (MB): 0.45
Forward/backward pass size (MB): 1.50
Params size (MB): 7.51
Estimated Total Size (MB): 9.46
==========================================================================
Optimizer: torch.optim.Adam
    Initial Learning Rate: 1e-2
    scheduler: LambdaLR (1 - epoch# / 100)

Convergence: 7 epochs of non-improving mean validation loss
```

Figure 4.2: MTP Model Summary

CHAPTER 5

EXPERIMENTS

In this chapter, I evaluate the proposed OPD and compare it against the standard stable diffusion plus CFG in both efficiency and visual quality. I train and test the MTP model on our dataset described in Section 4.1.

5.1 Experimental Setup

The experimental setup for this research is conducted on a server with a 10th Gen Intel(R) Core(TM) i9-10920X processor having 12-core, 24-thread configuration with a base clock speed of 4.8 GHz, and a maximum turbo frequency of 4800 MHz. The server has a total of 257.4 GB of system RAM, allowing for efficient handling of large dataset files and extensive computations.

The server is also equipped with two NVIDIA GeForce RTX 3090 GPUs, each with 24.5 GB of dedicated GPU memory. The optimization process and the models are loaded on the Cuda GPU servers. The dataset files are loaded on the system RAM to facilitate multi-tasking and faster processing of deep neural networks. This robust hardware configuration ensures optimal performance for the complex computations for training and evaluation of models.

5.2 MTPs

I train the model with the Adam optimizer and an initial learning rate of 1×10^{-2} adjusted to $1 - \frac{1}{epoch}$ over subsequent epochs. This progressive reduction in the learning rate throughout successive epochs thereby encourages model stabilization and refinement during training. Models are trained in the 50-100 epochs range, over 22,000 unique prompt pairs as highlighted in Table 1. To fully understand the characteristics of OPD, I have constructed four different versions of MTP, as outlined below.

5.2.1 Empty MTP

The Empty MTP model is trained to generate a positive plus empty prompt image using a single merged embedding. This is the most standard use of CFG as in Equation 3.1.The model takes in a combined input from the positive prompt and an empty string prompt, producing a merged embedding to be fed into the UNet. The model is trained with 1,075 samples over 100 epochs, achieving a average validation MSE loss of 0.087 as seen in loss curve Fig:5.1.

5.2.2 Object MTP

The Object MTP model is trained to remove an object from the positive prompt in the generated image. It combines input from the positive prompt (containing the target object) and a negative prompt with the target object alone, producing a merged embedding for the UNet. The model is trained with 7,123 prompt pairs over 50 epochs, achieving a average validation MSE loss of 0.092 as seen in loss curve Fig:5.2.

5.2.3 Quality MTP

The Quality MTP model is trained to improve the visual quality of the generated image. It takes in a combined input from the positive and negative prompt with undesirable quality attributes, producing a merged embedding for the UNet. The model is trained with 7,660 prompt pairs over 50 epochs, achieving a average validation MSE loss of 0.091 as seen in loss curve Fig:5.3.

5.2.4 Combined MTP

The Combined MTP is trained on the combined set of training data from Object MTP and Quality MTP. I want to see if I can train a one-solution-for-all MTP for the most typical use of CFG in diffusion models. During training, data is randomly selected between object-based negative prompt and quality-based negative prompt. The model itself is identical in structure compared to both cases discussed above. The model is trained for 100 epochs and it achieves an average validation MSE loss of 0.091 as seen in loss curve Fig:5.4.

5.3 Metrics

5.3.1 Efficiency

Since the original stable diffusion pipeline with CFG uses multiple forward passes, it requires increased running time and memory consumption. We employ the CUDA memory profiler to measure Peak GPU Memory Usage (PMU) and the time library in Python to record the total processing time per prompt set, Averaged over the Total number of Prompts (ATP). Notably, we run the experiments in both the batch and instance mode: batch mode denotes multiple prompt pairs processed simultaneously, while instance mode denotes individual prompts processed in separate passes. This is to demonstrate the common use cases by stable diffusion users.

5.3.2 Image quality

I use popular image similarity metrics to validate the quality and similarity of the images generated by OPD versus the original results from CFG. In particular, I use LPIPS [Zhang et al., 2018] for image similarity and FID for image quality. Smaller is better for both metrics.

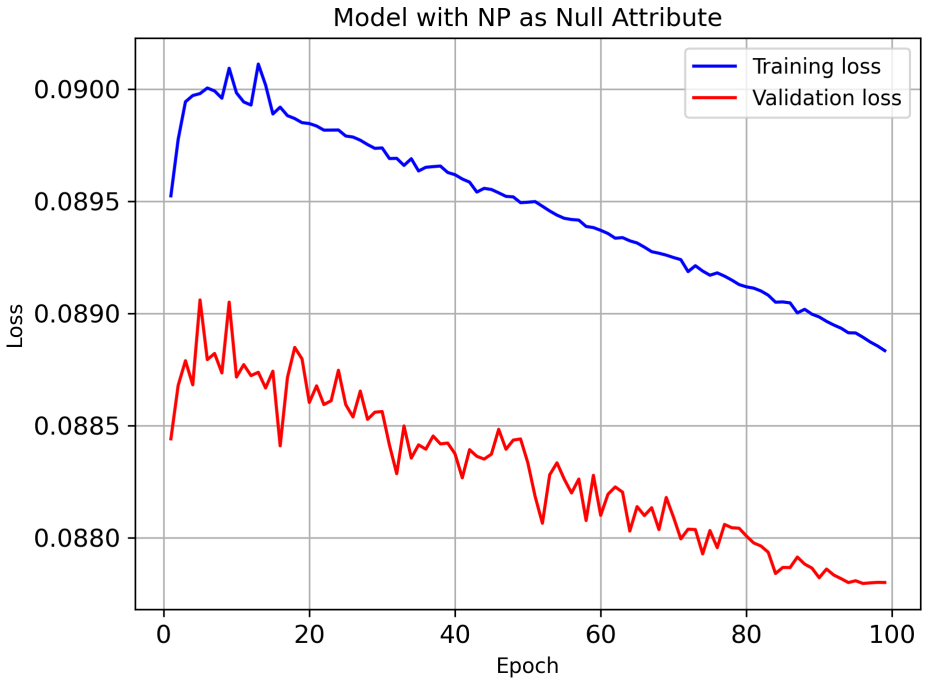


Figure 5.1: MTO Empty - Training Vs Validation Loss Curve

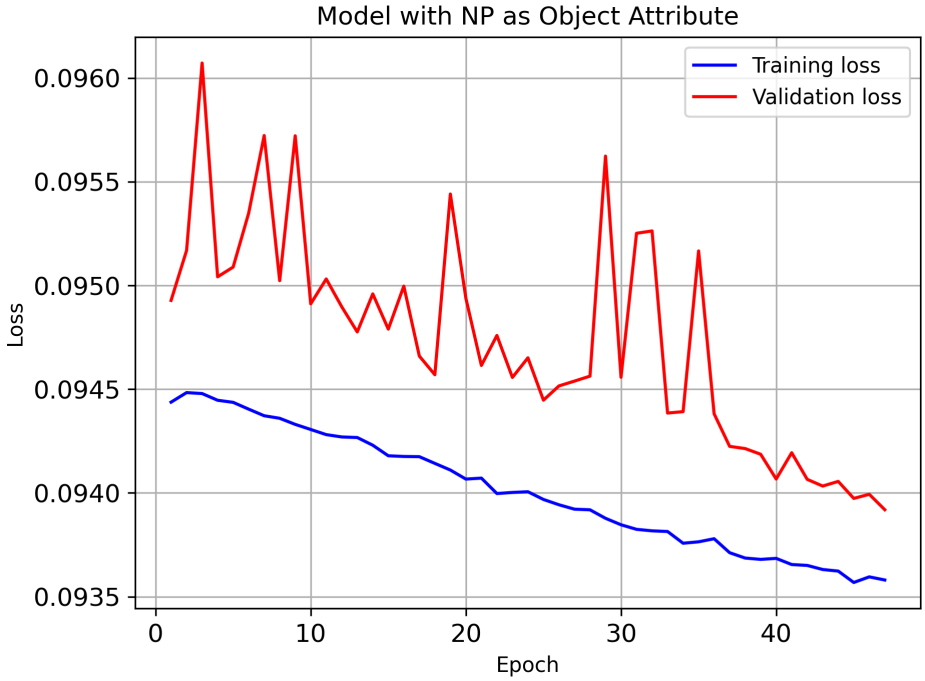


Figure 5.2: MTO Object - Training Vs Validation Loss Curve

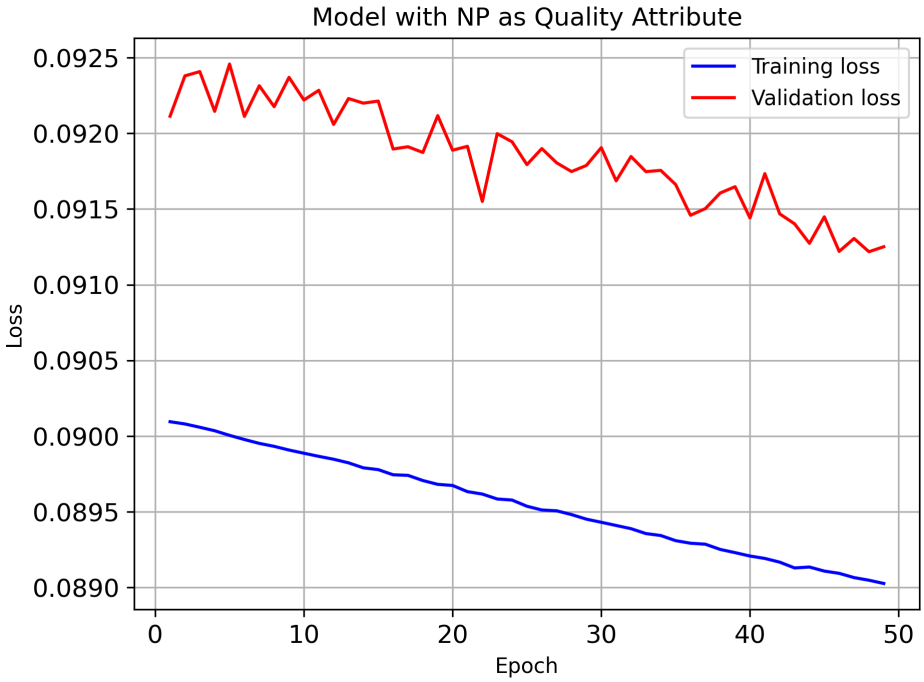


Figure 5.3: MTO Quality - Training Vs Validation Loss Curve

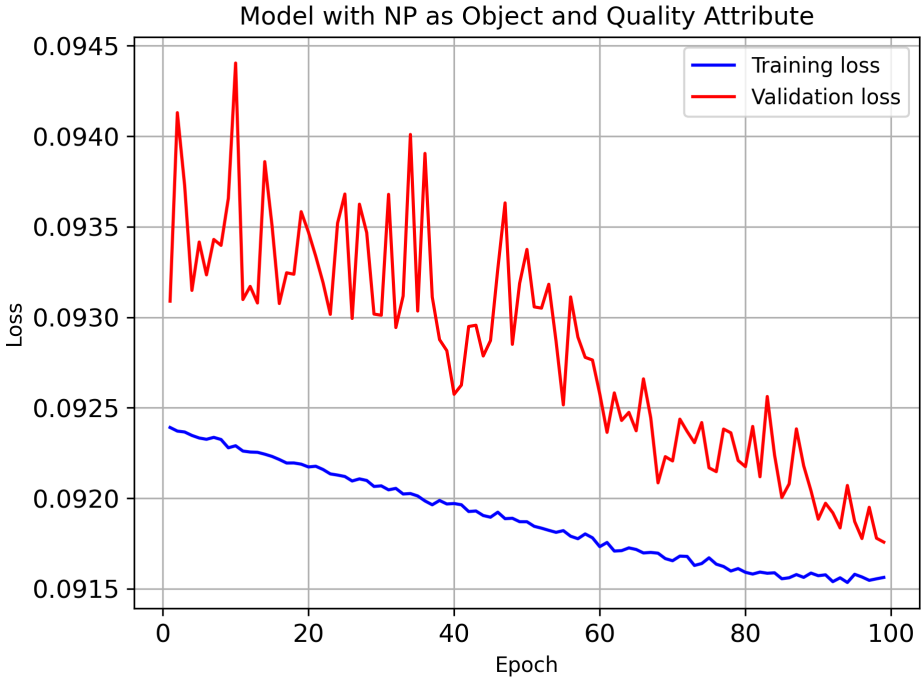


Figure 5.4: MTO Combined - Training Vs Validation Loss Curve

CHAPTER 6

RESULTS

6.1 Time and memory efficiency

A comparative analysis is in Table 2. My model is up to 2x faster with batch processing of size 10 and provides at least 1.72x speedup in other cases compared with the stable diffusion + CFG model. OPD also consumes 18% less memory.

Additionally, in a typical casual user's hardware setup (such as RTX3090 with 24G GPU memory) stable diffusion plus CFG model runs out of memory when processing a batch size of 50 prompts, while OPD can successfully finish the generation process with the batch. My model performs better regarding memory consumption while maintaining a stable processing time for large batches.

Table 2: Batch and instance mode efficiency results between stable diffusion + CFG and MTP.

Batch	**Pos + Neg**			**MTP**			**Reduction**		
	PMU	AMU	ATP	PMU	AMU	ATP	PMU	AMU	ATP
5	11.8G	11.8G	5.27s	11.2G	11.2G	2.8s	5%	5%	36%
10	13.1G	13.1G	5.28s	11.8G	11.8G	2.6s	9.9%	9.9%	38%
20	15.8G	15.8G	5.04s	13.1G	13.1G	**2.6s**	**17%**	17%	**38%**
50	-	-	-	17.0G	17.0G	2.5s	-	-	-

Instance	**Pos + Neg**			**MTP**			**Reduction**		
	PMU	AMU	ATP	PMU	AMU	ATP	PMU	AMU	ATP
1	8.4G	8.4G	11s	6.9G	6.9G	6.0s	17.8%	17.8%	45.4%
50	8.4G	8.4G	10.5s	6.9G	6.9G	6.1s	17.8%	17.8%	41.9%
200	8.4G	8.4G	10.5s	6.9G	6.9G	**6.1s**	**17.8%**	**17.8%**	**41.9%**

PMU: Peak Memory Usage AMU: Average Memory Usage
ATP: Average Time Per Prompt

6.2 Image quality

Table 3 shows image quality and similarity results from the four MTPs in the OPD setting compared against the original stable diffusion plus CFG images on our test set prompts. All of the MTP models achieve good performance on average in LPIPS: they are all below 0.25 which is a good indication of the visual similarity. **Figures 6.1, 6.2, 6.3, 6.4** contain example prompts and OPD outputs. **Tables 4, 5, 6, 7** contain LPIPS and FID comparison between ground truth image and OPD outputs.

Notably, Object MTP successfully removes the target object from the final image and produces high-quality results at the same level as the stable diffusion plus CFG. Quality MTP also works well in enhancing the overall quality of the generation by preventing undesirable attributes from the final image. Interestingly, the model has been trained with a specific set of quality attributes, yet it demonstrates the ability to generalize to other similar quality attributes. For example, the Quality MTP model accurately addresses "low resolution", a quality attribute that is not included in the training set. Despite not being explicitly informed about the type of the input negative prompt being about object or quality, the Combined MTP demonstrates remarkable performance in both object removal and image quality enhancement. I also notice that in many cases, the MTPs outperform MTO, generating images that more closely resemble the ground truth images. I suspect this is due to the learning-based approach being more "smooth" than single-image optimization.

To summarize, all versions of the trained MTP models achieve good visual quality and high efficiency compared with the standard stable diffusion plus CFG models, validating the success of the proposed One Prompt Diffusion pipeline. More results are available in Appendices A.

Table 3: OPD generated image quality and similarity with LPIPS and FID on the test set.

MTP	**Images**	**LPIPS**			**FID**		
		Min	Max	Avg	Min	Max	Avg
Empty	322	0.04	**0.60**	0.21	11.40	**643.6**	**155.4**
Object	1578	0.03	0.68	0.24	9.72	722.9	225.2
Quality	3980	0.02	0.69	**0.18**	9.28	742.8	201.5
Combined	5055	**0.02**	0.69	0.239	**4.5**	876.5	191.1

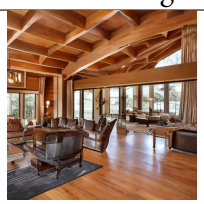
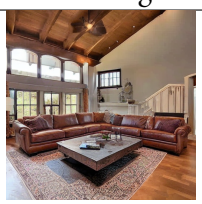
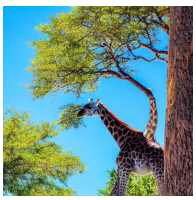
Sr	Positive	Negative	CFG + Neg	w/o Neg	Inversion	OPD
A	A large living room with leather couches and wood flooring	couch				
B	A pizza covered in lots of greens on top of a table	pizza				
C	A busy city street lined with stores, parked cars and people	car				
D	A giraffe standing next to a tree with blue sky in the background	giraffe				
E	A bench sitting on a lush green park with trees	bench				

Figure 6.1: Object MTP results. For a given positive prompt and negative prompt to remove objects, OPD achieves similar results as *CFG+Neg*.

Table 4: Object Loss Metrics

Sr	**LPIPS [GT, Inv]**	**LPIPS [GT, Merged]**	**LPIPS DIFF**	**FID [GT, Inv]**	**FID [GT, Merged]**	**FID DIFF**
A	0.06	0.14	0.08	40.85	82.76	41.9
B	0.12	0.13	0.01	132.58	238.55	105.97
C	0.09	0.12	0.03	76.55	94.19	17.63
D	0.08	0.09	0.01	40.59	59.82	19.23
E	0.13	0.09	**0.04**	101.58	102.38	0.80

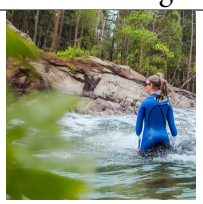
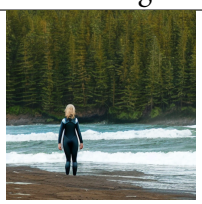
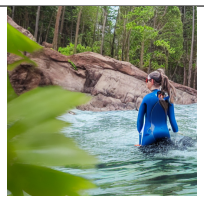
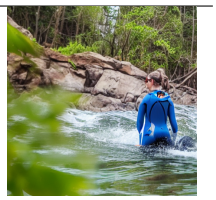
Sr	Positive	Negative	CFG + Neg	w/o Neg	Inversion	OPD
A	Woman in wet suit considers waves near forested coastline	low resolution				
B	A cat wearing a hat while resting its paws on top of a chair	noise				
C	An older pickup truck with decorative murals painted on	distorted				
D	A burger sitting on a table, with another next to it	low resolution				
E	A doll with a bonnet and a book on a pillow	low resolution				

Figure 6.2: Quality MTP results. For a given positive prompt and negative prompt for undesirable quality attributes, OPD achieves similar results as *CFG+Neg*.

Table 5: Quality Loss Metrics

Sr	LPIPS [GT, Inv]	LPIPS [GT, Merged]	LPIPS DIFF	FID [GT, Inv]	FID [GT, Merged]	FID DIFF
A	0.09	**0.07**	0.02	74.37	**52.34**	22.03
B	0.06	0.12	0.06	50.92	124.03	78.12
C	0.05	0.08	0.03	11.25	12.50	1.26
D	0.06	0.07	0.01	24.41	28.81	4.40
E	0.18	**0.12**	0.06	99.72	72.42	**27.29**

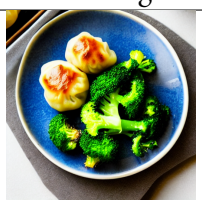
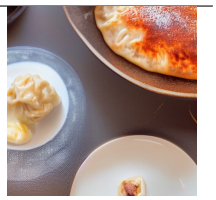
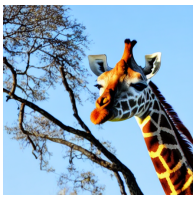
Sr	Positive	Negative	CFG + Neg	w/o Neg	Inversion	OPD
A	A white plate topped with broccoli and a dumpling	broccoli				
B	A small kitten inside of a laptop on the floor	grainy				
C	Ten porcelain pieces with floral patterns painted on them	clipped				
D	A large living room with leather couches and wood flooring	inappro -priate				
E	The head and neck of a giraffe with bare tree branches behind it	giraffe				

Figure 6.3: Combined MTP results. It can handle both object and quality negative prompts and achieve similar results as *CFG+Neg*.

Table 6: Combined Loss Metrics

Sr	LPIPS [GT, Inv]	LPIPS [GT, Merged]	LPIPS DIFF	FID [GT, Inv]	FID [GT, Merged]	FID DIFF
A	0.08	0.12	0.04	52.5	53.11	0.60
B	0.07	0.09	0.02	26.5	133.75	107.25
C	0.07	0.08	0.01	28.06	96.69	68.63
D	0.10	0.11	0.01	67.05	79.46	12.4
E	0.16	0.09	**0.07**	97.52	40.28	**57.54**

Sr	Positive	Negative	CFG	w/o CFG	Inversion	OPD
A	People riding bicycles down the road approaching a bird	“”				
B	A skier in the midst of a jump	“”				
C	A white cake with chocolate being poured over the top of it	“”				
D	A blue bench covered in different style bags	“”				
E	A view of the mountains is shown from an airplane window	“”				

Figure 6.4: Empty MTP results. The negative prompt here is an empty string. Our model can achieve similar quality output as the original *CFG+Neg*.

Table 7: Positive Prompt Only Loss Metrics

Sr	LPIPS [GT, Inv]	LPIPS [GT, Merged]	LPIPS DIFF	FID [GT, Inv]	FID [GT, Merged]	FID DIFF
A	0.04	0.07	0.03	21.39	23.73	2.33
B	0.10	0.11	0.01	24.5	30.74	6.24
C	0.12	0.12	**0.0**	81.57	**38.83**	42.74
D	0.11	0.09	**0.02**	63.26	**19.46**	43.8
E	0.11	0.06	**0.04**	77.15	**57.92**	19.23

CHAPTER 7

LIMITATIONS

Through the evaluation of my method, I have observed the following limitations.

7.1 Inconsistent results

Sometimes OPD might not produce the expected results. Fig 7.1 highlights some examples where the predicted embedding is not consistent with the originals. I observed that although the MTO embeddings encode the combined information of both prompts, the trained OPD model might not learn well for certain prompts. Furthermore, Fig 7.2 shows a few examples where my optimization algorithm MTO cannot optimize embeddings to produce identical images with the ground truth. Several reasons for this can be variations in the training dataset or inefficient training loops, which could be resolved by generating more training data and fine-tuning the training parameters.

7.2 Inconsistent negative prompt handling

While my method improves the existing process of handling negative prompts, the effectiveness of negative prompts in removing target objects is not always consistent. However, this is expected. As displayed in Figure 3.1 stable diffusion is not always consistent with negative prompt handling. My performance on those images are capped by the limitation from the stable diffusion model.

7.3 Dataset limitations

My prompt-to-image dataset used for training and evaluation, derived from the COCO dataset, may not cover all possible variations in positive prompts and negative prompts. With the rapid development of

prompt engineering approaches in stable diffusion models, my method might be limited by the specific negative prompt set that exists in the training set.

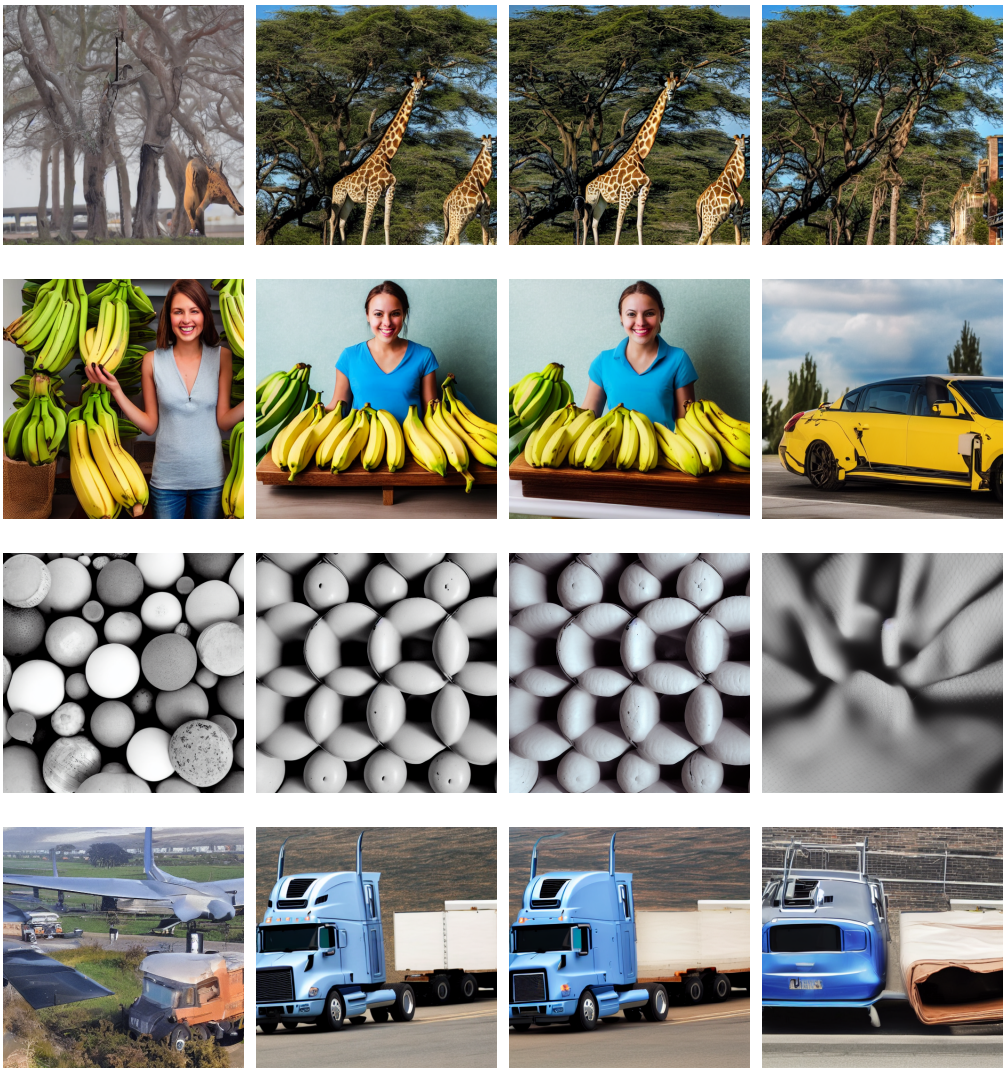
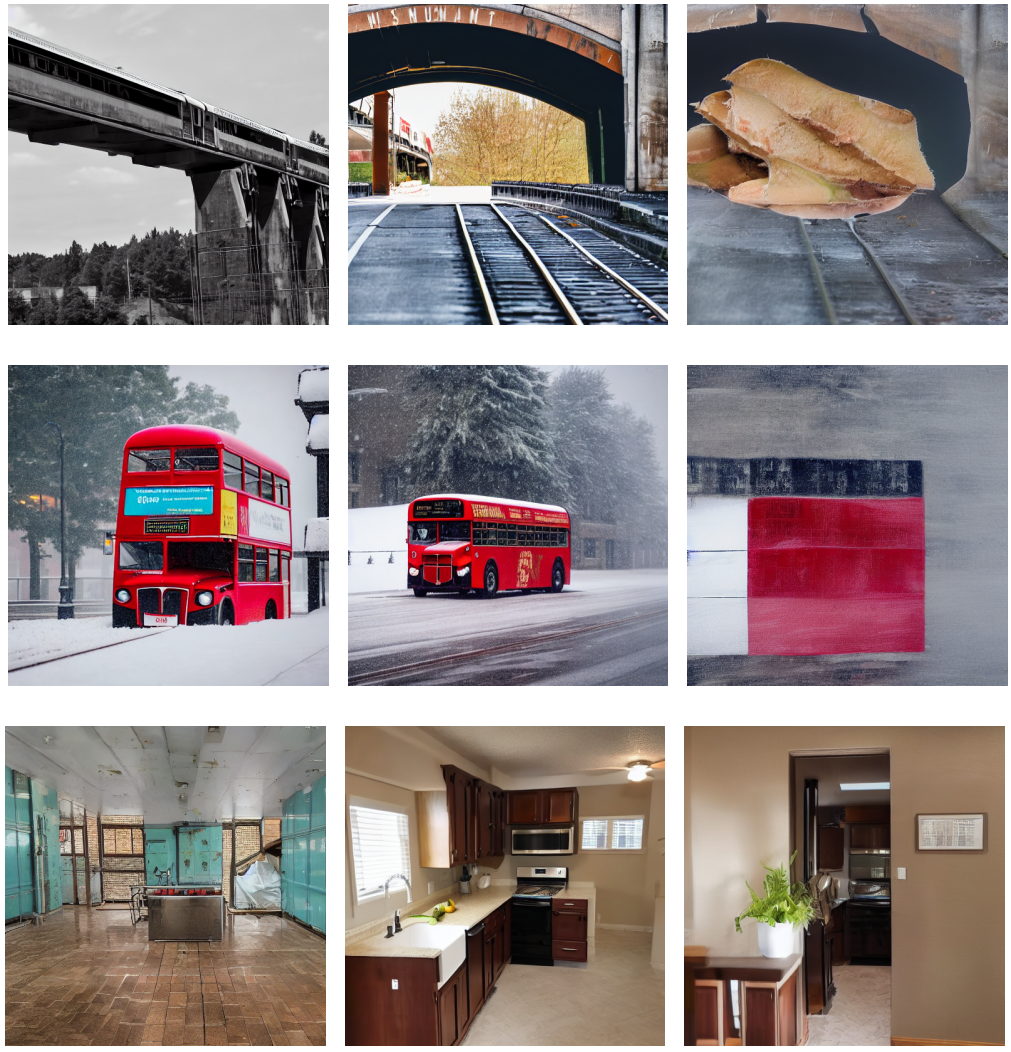


Figure 7.1: Examples of MTP failure cases. **(A)** Images generated using the positive prompt only with CFG. **(B)** Images generated using both positive and negative prompts with CFG. **(C)** Images generated using optimized merged embeddings without CFG. **(D)** Images generated using predicted merged embeddings without CFG. The models and prompts pairs used are: "Object MTP: A man watches a giraffe eating from a tree, negative: person", "Quality MTP: Smiling lady standing by two bunches of bananas on a table, negative: clipping", "Combined MPT: A black and white image of a lot of round objects, negative: grainy", and "Empty MTP: A large truck drives up to an airplane".



A B C

Figure 7.2: Examples of MTO failure cases. **(A)** Images generated using the positive prompt only with CFG. **(B)** Images generated using both positive and negative prompts with CFG. **(C)** Images generated using optimized merged embeddings without CFG. The models and prompts pairs used are: "Object MTP: A train traveling under a bridge past a train station, negative: train", "Quality MTP: A double decker bus during a snowy day, negative: clipping", and "Empty MTP: A kitchen area is being built that contains a deep sink, a dishwasher, and cabinets"

CHAPTER 8

CONCLUSION

In this work, I have presented a method called *One Prompt Diffusion (OPD)* to eliminate the usage of computationally expensive Classifier-Free-Guidance in diffusion models. To this end, I design an MLP model that predicts a merged embedding equivalent to the positive and negative prompt embeddings generated for CFG in the standard stable diffusion setup. OPD effectively reduces the computational resource requirement while achieving a 2x speedup and a $\sim 20\%$ reduction in memory consumption during batch processing. My method can generate high-quality realistic images by removing specific target objects mentioned in negative prompts or alleviating the undesirable quality attributes in the generated images. I propose a novel approach to greatly improve the efficiency of stable diffusion while keeping the generation quality. My method can be combined with other efficiency-driven approaches for diffusion models, and I carefully design a prompt-to-image dataset that features the design of complex negative prompts. It can be used by a broader group of researchers who want to study the effects of complex prompts in diffusion models.

In future work, I plan to address the limitations outlined in Chapter 7. In particular, I am interested in designing more flexible negative prompts and developing better verification steps to handle inconsistent results. In addition, I am very interested in understanding better the predicted merged embeddings. Currently, the embeddings cannot be decoded back to words and I plan to explore the explainability aspect of OPD.

BIBLIOGRAPHY

Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A. H., Chechik, G., & Cohen-Or, D. (2022). An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial networks. *Advances in neural information processing systems*, *27*.

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models.

Ho, J., & Salimans, T. (2022). Classifier-free diffusion guidance.

Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., & Aila, T. (2021). Alias-free generative adversarial networks. *Proc. NeurIPS*.

Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., & Irani, M. (2023). Imagic: Text-based real image editing with diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6007–6017.

Kim, G., Kwon, T., & Ye, J. C. (2022). Diffusionclip: Text-guided diffusion models for robust image manipulation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2426–2435.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Li, S., van de Weijer, J., Hu, T., Khan, F. S., Hou, Q., Wang, Y., & Yang, J. (2024). Get what you want, not what you don't: Image content suppression for text-to-image diffusion models. *arXiv preprint arXiv:2402.05375*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755.

Liu, X., Gong, C., & Liu, Q. (2022). Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*.

Liu, X., Zhang, X., Ma, J., Peng, J., et al. (2023). Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. *The Twelfth International Conference on Learning Representations*.

Luo, S., Tan, Y., Huang, L., Li, J., & Zhao, H. (2023). Latent consistency models: Synthesizing high-resolution images with few-step inference.

Meiri, B., Samuel, D., Darshan, N., Chechik, G., Avidan, S., & Ben-Ari, R. (2023). Fixed-point inversion for text-to-image diffusion models. *arXiv preprint arXiv:2312.12540*.

Meng, C., Rombach, R., Gao, R., Kingma, D., Ermon, S., Ho, J., & Salimans, T. (2023). On distillation of guided diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14297–14306.

Miyake, D., Iohara, A., Saito, Y., & Tanaka, T. (2023). Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. *arXiv preprint arXiv:2305.16807*.

Mokady, R., Hertz, A., Aberman, K., Pritch, Y., & Cohen-Or, D. (2023). Null-text inversion for editing real images using guided diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6038–6047.

Oppenlaender, J. (2023). A taxonomy of prompt modifiers for text-to-image generation. *Behaviour & Information Technology*, 1–14.

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

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. arxiv 2022. *arXiv preprint arXiv:2204.06125*.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022a). High-resolution image synthesis with latent diffusion models.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022b). High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. (2022). Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, *35*, 36479–36494.

Salimans, T., & Ho, J. (2022). Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*.

Song, J., Meng, C., & Ermon, S. (2020). Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y., Dhariwal, P., Chen, M., & Sutskever, I. (2023). Consistency models. *arXiv preprint arXiv:2303.01469*.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2020). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Sun, W., Li, T., Lin, Z., & Zhang, J. (2024). Spatial-aware latent initialization for controllable image generation. *arXiv preprint arXiv:2401.16157*.

Tang, R., Liu, L., Pandey, A., Jiang, Z., Yang, G., Kumar, K., Stenetorp, P., Lin, J., & Ture, F. (2022). What the daam: Interpreting stable diffusion using cross attention.

Wallace, B., Gokul, A., & Naik, N. (2023). Edict: Exact diffusion inversion via coupled transformations. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22532–22541.

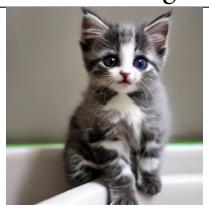
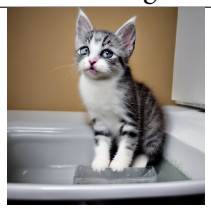
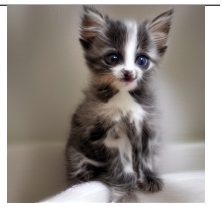
Wen, Y., Jain, N., Kirchenbauer, J., Goldblum, M., Geiping, J., & Goldstein, T. (2024). Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *Advances in Neural Information Processing Systems*, *36*.

Weng, L. (2021). What are diffusion models? *lilianweng.github.io*. https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

Wu, Q., Liu, Y., Zhao, H., Kale, A., Bui, T., Yu, T., Lin, Z., Zhang, Y., & Chang, S. (2023). Uncovering the disentanglement capability in text-to-image diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1900–1910.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. *CVPR*.

APPENDIX A

Positive	Negative	CFG + Neg	w/o Neg	Inversion	OPD
a gray and white kitten sitting in a bathroom sink.	sink				
a group of people stand on a side walk near a city bus	bus				
A man parking on a road with his motorcycle.	motor-cycle				
A white toilet sitting next to a large window.	toilet				
An intersection with antique cars and a bus at it.	bus				

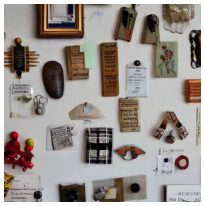
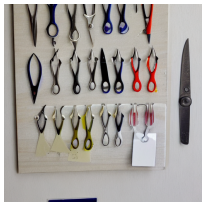
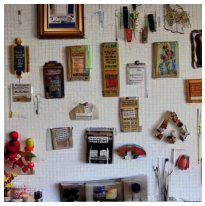
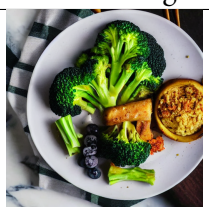
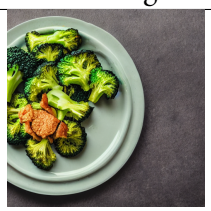
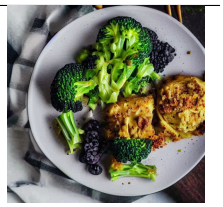
Positive	Negative	CFG + Neg	w/o Neg	Inversion	OPD
A blue motorcycle parked next to a van with a painting on its side	motor-cycle				
There is a wooden bench in front of a window	bench				
A little boy and little girl in a train car.	train				
Two giraffes standing next to each other by a door.	giraffe				
A dog rests his head on the edge of a boat at sea.	boat				
A collection of small scissors labeled and pinned on a board.	scissors				

Figure A.1: Object MTP results. For a given positive prompt and negative prompt to remove objects, OPD achieves similar results as *CFG+Neg*.

Positive	Negative	CFG + Neg	w/o Neg	Inversion	OPD
Plate of food with broccoli on glass plate with table cloth	clipping				
A man riding on the back of a motorcycle.	unnatural				
An airplane just landed on the runway	inappropriate				
A lady hold a game controller, pointed towards a laptop.	unnatural				
A yellow food truck parked close to a car	inappropriate				
a double deckered bus on a city street	inappropriate				
A sandwich with nachos and a salad on a plate.	low resolution				

Positive	Negative	CFG + Neg	w/o Neg	Inversion	OPD
A white bus parked next to a sidewalk near a fence.	low resolution				
Several cars parked along the side of a street next to a street sign.	low resolution				
Kitchen area with counter top space and dining room table.	low resolution				
A brown bird perched on top of a metal fence.	low resolution				
A plate of broccoli and meat on a table.	low resolution				
a double deckered bus on a city street	inappropriate				
a close up of a motorcycle parked near a building	noise				

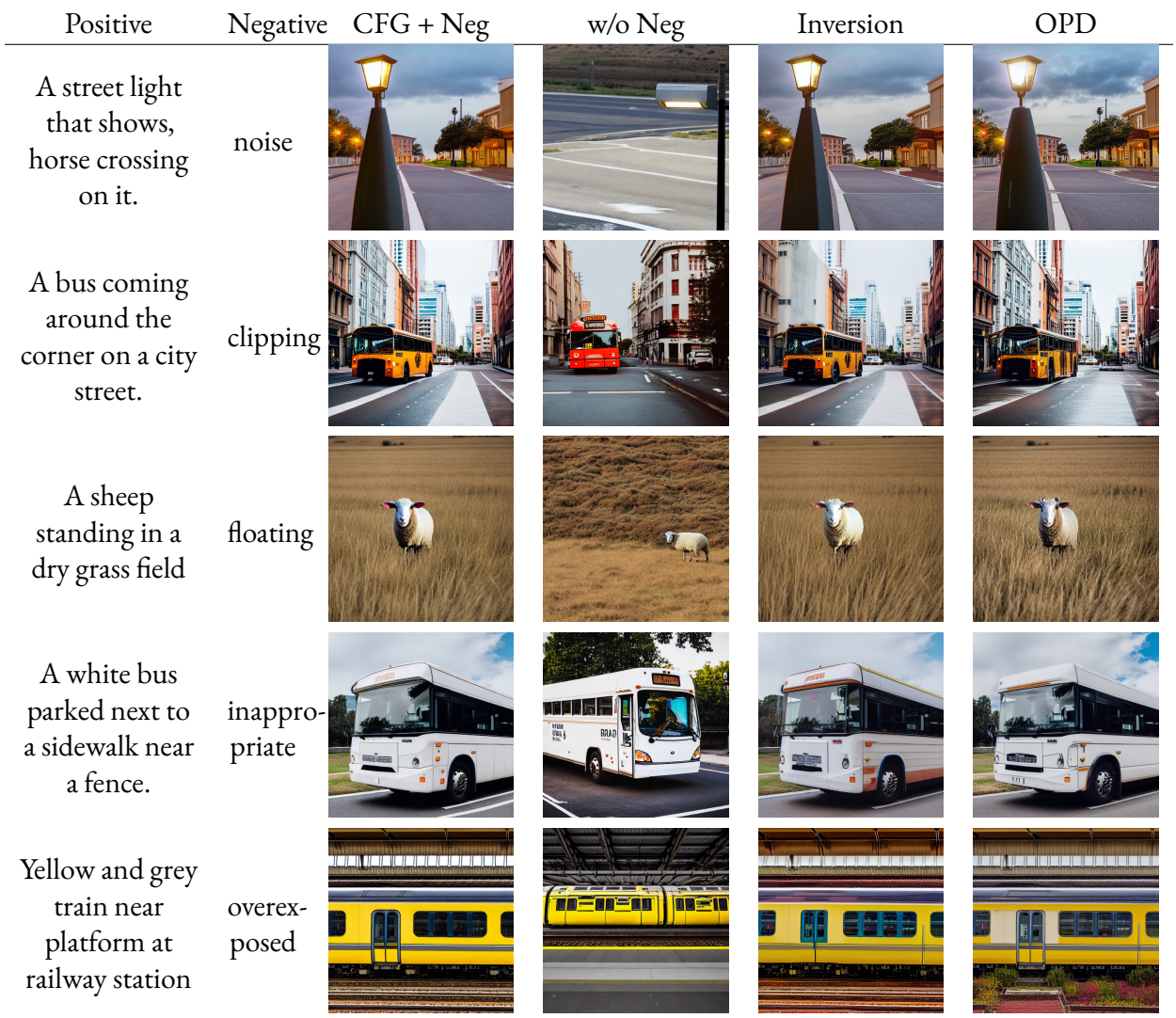


Figure A.2: Quality MTP results. For a given positive prompt and negative prompt for undesirable quality attributes, OPD achieves similar results as *CFG+Neg*.

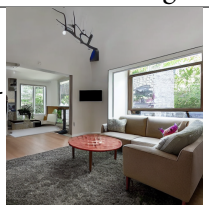
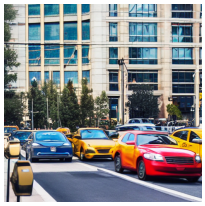
Positive	Negative	CFG + Neg	w/o Neg	Inversion	OPD
A disorderly living area is free from decorative elements.	inappropriate				
A fire hydrant with a tire underneath it.	fire hydrant				
A yellow bus is on a street near a black guard rail	grainy				
a few cars waiting at a traffic light	car				
A plate holds a good size portion of a cooked, mixed dish that includes broccoli and pasta.	floating				
A majestic bear looks out across a grass plain.	bear				
A truck that is sitting in the street.	overexposed				

Figure A.3: Combined MTP results. It can handle both object and quality negative prompts and achieve similar results as *CFG+Neg*.

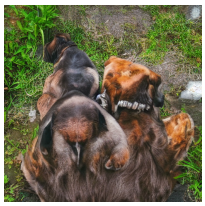
Positive	Negative	CFG	w/o CFG	Inversion	OPD
A blue bench covered in different style bags	“”				
A wooden kitchen table and bench on wooden floor	“”				
a dog, sits on a rug, looking at a television	“”				
A kitchen with wooden counter tops and a stove top oven	“”				
Pedestrians walking down a sidewalk next to a small street	“”				
Two dogs are looking up while they stand near the toilet in the bathroom	“”				
A man is riding a bike and talking on his cell phone	“”				

Figure A.4: Empty MTP results. The negative prompt here is an empty string. My model can achieve the similar quality output as the original *CFG+Neg*.