Leipzig University
Institute of Computer Science
Data Science, M.Sc.

Manipulating Embeddings of Stable Diffusion Prompts

Master's Thesis

Julia Peters)

1. Referee: Prof. Dr. Martin Potthast
2. Referee: Dr. Harrisen Scells

Submission date: December 7, 2023

Declaration

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work.

Leipzig, December 7, 2023

. . . . . .
Julia Peters

Abstract

Generative text-to-image models have shown remarkable progress in recent years, transforming textual prompts into intricate, high-quality images. However, the prevailing method of influencing image output, known as prompt engineering, often results in an unpredictable and time-consuming trial-and-error process of refining textual prompts to derive desired images. This thesis addresses the challenges associated with prompt engineering by developing and evaluating three automated methods for manipulating prompt embeddings. The first method, Metric Based Optimization, employs gradient optimization techniques to automatically refine the text prompt embeddings towards a specific metric, eliminating the need for manual prompt modification. While effective in generalizing across various random seeds, it does face the limitation of over-optimization for certain prompts. For the second approach, Iterative User Interaction, the intention is to assist users in their creative endeavors by enabling them to explore the image space. Rather than adjusting the textual prompt directly, users navigate along selected directions of close prompt embeddings, providing a more intuitive and guided interaction. This method was found to be user-friendly in a conducted study, although the need for greater control mechanisms was expressed. Lastly, Seed-Invariant Optimization aspires for consistent image outputs across various random seeds, even if some variation persists. Results showed that while all methods offer advancements in user interaction with text-to-image models, certain limitations exist, necessitating further refinement. Overall, this thesis contributes to enhancing the user experience in text-to-image generation by providing more reliable and intuitive control mechanisms, reducing the complexities associated with prompt engineering.

Contents

Chapter 1

Introduction

In the rapidly evolving landscape of machine learning and artificial intelligence, text-to-image generative models have emerged. These generative models, such as DALL-E 2 [Ramesh et al., 2022b], Imagen [Saharia et al., 2022] or Stable Diffusion [Rombach et al., 2022], bring forth the capability to transform textual descriptions, termed as prompts, into intricately detailed images of high quality, opening up new creative possibilities. However, despite these advancements, affecting the image generation process continues to be a significant challenge, which is largely based on the discipline of prompt engineering. This concept implies an iterative refinement of the prompt to achieve desired results. Users primarily rely on trial-and-error methods, adjusting prompts until a satisfactory output is found. This process can be time-consuming, hindering the users to efficiently express their creative vision. Therefore, best practices and design guidelines for the production of better text to image outcomes have emerged [Liu and Chilton, 2022, Oppenlaender, 2022]. Even this way by far not enough control can be provided. Despite these efforts, associated with prompt engineering, users have to face further considerable obstacles. The model's internal logic and understanding of prompts often appear incalculable, leading to unforeseen deviations between what the user intended and the generated visual. With the inability to control specific granular details and the unpredictability introduced by varying random seeds, users find themselves evolving an approximate orientation but without actual control [Deckers et al., 2023].

These challenges reflect an increasing demand to develop more flexible and user-friendly approaches for image generation. The objective is to design systems and frameworks not only simplifying the image generation process but also providing users a higher level of control without the sophistication of prompt engineering. Hence, this thesis' aim is to address this gap by propos-

ing and investigating flexible approaches for user-driven image generation.

Rather than rephrasing textual description, the introduced techniques facilitate the refinement of a prompt's embedding in targeted manners. Within the workflow of text-to-image models, a prompt undergoes translation by a text encoder into an embedding, which is a high-dimensional numerical representation that the generative model can interpret. Subsequently, an image can be generated based on this representation. The premise of these methods rests on the principle that arbitrary subtle modifications to the embedding can lead to arbitrary nuanced changes in the generated image. By adjusting the embedding, the information encapsulated within the original prompt can be refined. This approach alleviates the need for users to rephrase their intentions verbally and to construct descriptions that the model interprets correctly, thereby enhancing user satisfaction with the image generation process.

This research will focus on the following three strategies updating the text embedding without revision of the text prompt:

1. **Metric Based Optimization**: To tackle the circumstances, where a generated image aligns at a broader level with users expectations, while missing specific nuances or a certain level of aesthetics, the thesis explores a metric-oriented technique. This technique facilitates precise optimization in the embedding space of the prompt by identifying prompts that yield images complying with the chosen detail metrics. Simultaneously, the original intent remains aligned.

2. **Iterative User Interaction**: For instances where users do not have an exact visual image in mind, helping them find a satisfactory visual representation becomes imperative. For that purpose an interactive image generation mechanism is introduced. Beginning with images complying with a base prompt, the approach refines the embedding of the prompt through user interactions to converge on visuals aligning more closely with user feedback.

3. **Seed-Invariant Optimization**: In order to address the inconsistency introduced by varying seeds, which can often lead to divergent outputs for a well-crafted prompt, a method is introduced to design seed independent prompt embeddings. This ensures that the derived images remain consistently, irrespective of the seed chosen. The reduction of randomness in image outcomes intends to provide users with a precise image editing tool.

For the implementation of these methods the model under consideration is Stable Diffusion. However, it is important to emphasise that the methods and results presented in this thesis are also applicable and relevant to other text-to-image models.

The subsequent sections delve into a comprehensive evaluation of these techniques, assessing their effectiveness through experimental analyses and a user study. The structure of this thesis is as follows:

- **Chapter 2 (Technical Background)**: Before continuing with the core methodologies, it is crucial to have a foundational understanding of the technical mechanisms driving image generation. This chapter clarifies the underlying principles, thereby equipping the reader with the knowledge necessary to comprehend the details of the proposed image guidance techniques.
- **Chapter 3 (Related Work)**: This chapter showcases studies and works with a similar research focus. Specifically, it illustrates attempts in the domain of influencing image generation and automated prompt engineering, providing context for the methodologies presented in this thesis.
- **Chapters 4-6 (Methods)**: These chapters elaborate on the three implemented methods, evaluating their effectiveness through both experimental analysis and a user study.
- **Chapter 7 (Conclusion)**: The conclusion offers a comprehensive summary of the results and insights gained throughout this research.
- **Chapter 8 (Future Work)**: Lastly, the outlook chapter concludes this thesis by outlining potential avenues for future research based on the methods and findings presented.

Chapter 2

Background

Stable Diffusion [Rombach et al., 2022], introduced in August 2022 [stability.ai, 2022] as an open-source image synthesis model on GitHub, holds significant importance, as highlighted by this thesis' title. Due to its broad scientific recognition and open availability it has been employed in this research. In the image generation system of Stable Diffusion, an embedding generated from a text prompt forms a primary element. The developed image guidance methods modify these embeddings to different extends, leading to a closer alignment of user preferences with the visual output. To fully comprehend the adopted methodologies, it is vital to enter in depth into the image generation technique of Stable Diffusion itself, particularly corresponding to the type of Latent Diffusion Models (LDMs) [Rombach et al., 2022].

Since these models fall under the broader category of Diffusion Models (DMs) [Sohl-Dickstein et al., 2015], Variational Autoencoders (VAEs) [Kingma and Welling, 2014] are initially covered as the foundational framework in this chapter. Subsequently, follows the elaboration of DMs, including LDMs, laying the groundwork for understanding Stable Diffusion's image synthesis technique. A comprehensive grasp of DMs, containing their components as well as their training objective, is crucial for understanding the sampling process and the gradient based approaches in this research. Accordingly, these key concepts are addressed in this chapter with a concluding explanation of the image generation process itself providing the knowledge base for the realized image guidance methods.

2.1 Variational Autoencoder

VAEs are a type of generative models that employ a probabilistic approach to encode data into a compressed form within a latent space, a domain where complex data patterns are represented in a simplified manner. Subsequently, they enable the reconstruction or generation of new instances that bear similarity to the original input. The architecture of a VAE typically comprises two main components: an encoder that condenses the input into a compact representation, and a decoder that reconstructs data from this compact form [Kingma and Welling, 2014].

A DM can be understood as a type of Markovian Hierarchical VAE [Luo, 2022]. Within the context of continuous time, researchers such as Song et al. [2021b], Huang et al. [2021], and Kingma et al. [2021] have shown that the primary training goal can be closely aligned with the Evidence Lower Bound of a deeply structured VAE. This implies that optimizing a DM is akin to training a very deep hierarchical VAE. Accordingly, this section serves as a thorough explanation of VAEs, focusing on their training process. This includes the challenges of backpropagation in a directed probabilistic environment that deals with continuous variables and complex posterior distributions. Finally, a brief overview of hierarchical VAEs [Sønderby et al., 2016] is also provided.

2.1.1 Evidence Lower Bound

In the following the fundamental concepts underlying the training objective of VAEs are concisely summarized, as detailed by Kingma and Welling [2014] and Luo [2022]. This way a foundational understanding necessary for deriving the corresponding loss function for DMs can be provided.

VAEs encode data into a latent space, resulting in latent variables z. The data x, observable in the regular, high-dimensional space, is thus linked to z through the joint probability distribution $p(x, z)$. This relationship is fundamental to the VAE's ability to not only reconstruct the original data from the latent representation but also to generate new data that resembles the original dataset [Kingma and Welling, 2014, Luo, 2022].

In order to optimize a VAE, a common approach is to maximize the likelihood of the observed data $p(x)$. However, a direct computation is complex.

Either a simultaneous integration of all latent variables z is required in the following equation [Luo, 2022]:

$$p(x) = \int p(x, z)\, dz$$

Alternatively, access to $p(z|x)$ is demanded to calculate [Luo, 2022]:

$$p(x) = \frac{p(x, z)}{p(z|x)}$$

The Evidence Lower Bound (ELBO) is introduced as a solution, referring to a lower bound of the evidence, as the name suggests. The evidence corresponds to the log likelihood of the observed data. By maximizing the ELBO, the exact evidence or at least an approximation can be derived.

The relationship can be expressed in formal terms [Kingma and Welling, 2014]:

$$\log p(x) \geq \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{p(x, z)}{q_\phi(z|x)} \right] = \text{ELBO}$$

The optimization of ELBO aims to approximate the true latent posterior $p(z|x)$ with a variational distribution $q_\phi(z|x)$. Here, ϕ refers to the parameters of this approximation. The ELBO can be deciphired into two terms [Luo, 2022]:

$$\begin{aligned}
\mathbb{E}_{q_\phi(z|x)} \left[\frac{\log p(x, z)}{q_\phi(z|x)} \right] &= \mathbb{E}_{q_\phi(z|x)} \left[\frac{\log p_\theta(x|z) p(z)}{q_\phi(z|x)} \right] \\
&= \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)] + \mathbb{E}_{q_\phi(z|x)} \left[\frac{\log p(z)}{q_\phi(z|x)} \right] \\
&= \underbrace{\mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)]}_{\text{reconstruction term}} - \underbrace{D_{KL}(q_\phi(z|x) || p(z))}_{\text{prior matching term}}
\end{aligned}$$

The reconstruction term evaluates how accurately the variational distribution can regenerate original data from learned latents. Conversely, the prior matching term equates to the KL divergence between $q_\phi(z|x)$ and $p(z|x)$. It measures similarity between the variational distribution and prior latent belief. Hence, by maximizing the ELBO, the KL divergence between $q_\phi(z|x)$ and $p(z|x)$ is minimized, even when the true posterior is intractable and simultaniously the reconsruction term is maximized.

As displayed in Figure 2.1, a VAE contains an encoder and a decoder part. The function $p(x|z)$ acts as a decoder that translates latent variables z to the data space, while $q_\phi(z|x)$ serves as an encoder that translates data to the latent variable space. Thus, maximizing the ELBO serves as a reliable approximation for inference and learning in complex probabilistic models, encapsulating both the data reconstruction term and a term that encourages the approximate posterior to align with the prior [Kingma and Welling, 2014, Luo, 2022].

Typically, the encoder of VAE models a multivariate Gaussian with diagonal covariance and the prior is a standard multivariate Gaussian [Luo, 2022]. Given these distributions, the reconstruction term is computed by employing the Monte Carlo estimate, while the KL divergence in the ELBO can be analytically calculated by reparameterization, as described in Section 2.1.2 [Luo, 2022].

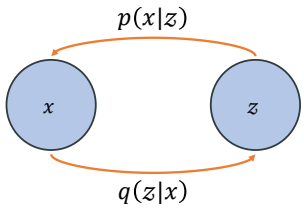


Figure 2.1: A VAE, with encoder $q(z|x)$ over latent variables z for observation x and decoder $p(x|z)$, adapted from Luo [2022].

2.1.2 Reparameterization

In order to perform an efficient learning in directed probabilistic models involving continuous variables with intractable posterior distributions, Kingma and Welling [2014] introduced the method of reparameterization, which is illustrated in Figure 2.2. It facilitates the application of backpropagation within a stochastic process by expressing a random variable with a deterministic function of a noise variable. Essentially, samples from an arbitrary Gaussian distribution, $x \sim N(x; \mu, \sigma^2)$, can be recast as $x = \mu + \sigma\epsilon$, where $\epsilon \sim \mathcal{N}(\epsilon; 0, I)$. This means that any Gaussian is essentially a standard Gaussian that has been scaled by its variance and shifted by its mean.

In the context of VAEs, the latent variable z is determined by the input x and noise ϵ: $z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon$, where $\odot$ signifies element-wise multiplication. With this reparameterized format, it becomes possible to compute gradients with respect to ϕ, aiding in the optimization of the model's encoder

means μ_ϕ and deviations σ_ϕ. Therefore, VAEs leverage both the reparameterization trick and Monte Carlo estimates to optimize the ELBO over both ϕ and θ.

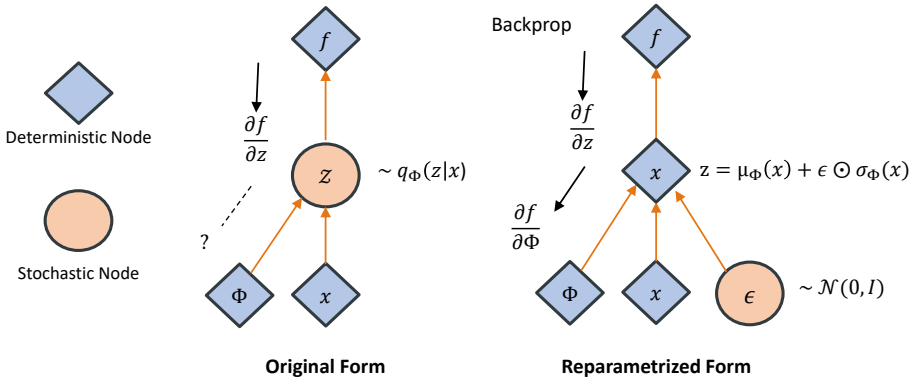


Figure 2.2: Reparametrization, adapted from Sharma [2021].

2.1.3 Hierarchical Variational Autoencoder

A Hierarchical VAE (HVAE) [Sønderby et al., 2016] is displayed in Figure 2.3. It is an extension of a standard VAE that has multiple layers of latent variables instead of merely a single one. Each layer's latent variables are influenced by the layer above it. This hierarchy allows for more complex and nuanced representations of data. The training objective of an HVAE is similar to a standard VAE. Comparably, an ELBO can be derived.

In a special version called Markovian HVAE [Luo, 2022], each layer depends only on the layer directly above it, simplifying the model. Diffusion processes can be viewed as similar to the hierarchical layers in HVAEs, allowing the two models to be related in their objectives and techniques for data representation and generation.

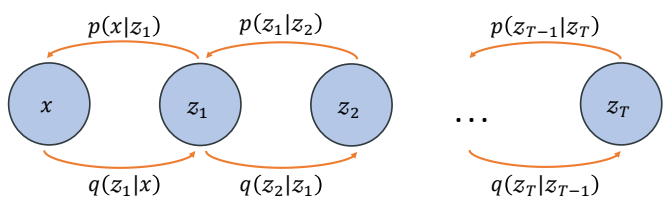


Figure 2.3: A HVAE using Markov Chain with T hierarchical latents. Each latent z_t is derived solely from its preceding latent z_{t+1}, $t \in \{1, \ldots, T\}$. Adapted from Luo [2022].

2.2 Diffusion Models

This section illuminates the technical intricacies of Diffusion Models (DMs), laying the groundwork for understanding Stable Diffusion and, by extension, the image-guiding methodologies explored in a subsequent section. Drawing from foundational principles of non-equilibrium thermodynamics, Diffusion Probabilistic Models, as proposed by Sohl-Dickstein et al. [2015], have been fundamental for the development of Stable Diffusion.

DMs constitute a category of generative models that employ two Markov chains to transform an initial data distribution, like Gaussian noise, into a target distribution. This transformation involves a forward chain that progressively adds noise to the data and a reverse chain which is a learned denoising mechanism that recovers the original data. Through this bidirectional process, DMs generate data samples closely approximating the target distribution.

In the following subsections, an in-depth exploration of the mechanics of DMs is provided. This includes examination of both the forward and reverse chains and an explanation of the training objectives of DMs. A comparison to VAEs will also be drawn to highlight the shared aspects of these models.

Subsequently, the concept of conditional image generation is discussed, establishing the framework for integrating external variables like text into image generation algorithms. This is followed by a focus on key components vital for the functionality of DMs, namely the denoising U-Net [Ronneberger et al., 2015] and the CLIP text encoder [Radford et al., 2021].

Attention will then be directed towards the computational challenges intrinsic to DMs due to their operations in pixel space. To address these challenges, the architecture of LDMs, a modified variant designed for computational efficiency, is introduced.

Upon concluding this section, the reader will have a comprehensive technical understanding of DMs, laying the foundation for the exploration of Stable Diffusion and its image generation capabilities.

2.2.1 Forward Process

In order to transform Gaussian noise into images, a DM initially disrupts the original data incrementally with noise within the forward chain. The reconstruction can be learned in the reverse chain. In the following the forward

process is reflected based on the work of Sohl-Dickstein et al. [2015] and Ho et al. [2020].

From a data distribution, denoted as $x_0 \sim q(x_0)$, a sequence of random variables x_1, x_2, ..., x_T can be generated by the forward Markov process. The generation of each x_t in the sequence is dependent on its immediate predecessor x_{t-1}, as depicted by $q(x_t|x_{t-1})$.

Employing the chain rule of probability along with the Markov property, symbolized as $q(x_1, ..., x_T|x_0)$ or $q(x_{1:T}|x_0)$, the joint distribution of x_1, x_2 ... x_T conditioned on x_0, can be broken down into [Ho et al., 2020, Sohl-Dickstein et al., 2015]:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$$
$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

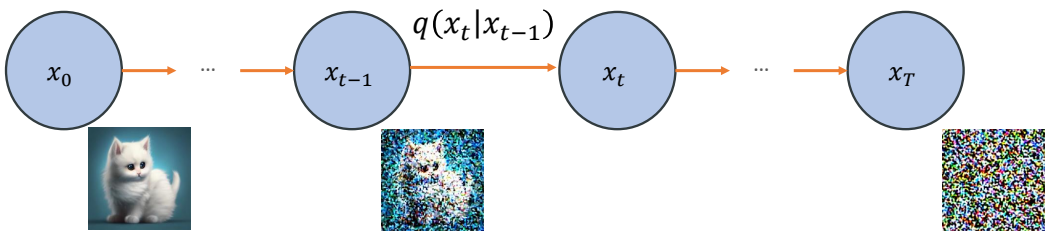


Figure 2.4: Forward diffusion process, adapted from Ho et al. [2020].

Let $\alpha_t = 1 - \beta_t$, $\bar{\alpha} = \prod_{i=1}^{t} \alpha_i$ and $\epsilon_0, ..., \epsilon_{t-2}, \epsilon_{t-1} \sim \mathcal{N}(0, I)$. Application of reparameterization (Section 2.1.2) results in [Ho et al., 2020, Luo, 2022]:

$$\begin{aligned} x_t &= \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1} \\ &= \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} \\ &= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\epsilon_{t-2} \\ &= ... \\ &= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_0 \end{aligned}$$

This allows to derive the following property of the forward process.

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$$

This property is instrumental in deriving an efficient training objective, as will be detailed in Section 2.2.3.

2.2.2 Reverse Process

After the forward process, the reverse chain is employed to sample images from Gaussian noise [Sohl-Dickstein et al., 2015]. Henceforth, this procedure is summarized as detailed by Sohl-Dickstein et al. [2015] and Ho et al. [2020].

In order to reverse the forward process $q(x_{t-1}|x_t)$, $x_T \sim N(0, I)$ sampled from the Gaussian distribution $q(x_{t-1}|x_t)$ is not known and therefore intractable. Accordingly, a model p_θ must be trained to approximate the appropriate conditional probabilities. This way a reversal of the diffusion process can be obtained.

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$
$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

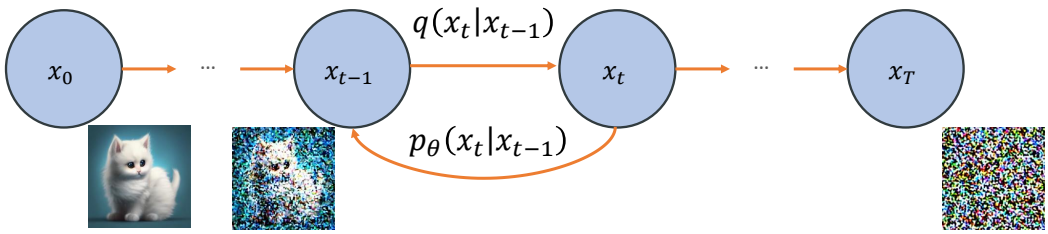


Figure 2.5: Reverse diffusion process, adapted from Ho et al. [2020].

During training the mean $\mu_\theta(x_t, t)$ as well as the covariance matrix $\Sigma_\theta(x_t, t)$ are approximated for each timestep t.

Notably, conditioning in addition on x_0, the reverse process becomes tractable.

$$p_\theta(x_{t-1}|x_t, x_0) = N(x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t I)$$

2.2.3 Training objective

To gain a comprehensive understanding of the forward process and the image generation, it is beneficial to grasp the DM's loss function which is defined for the purpose of reconstructing the data within the forward chain. The approach of establishing an objective function to learn the respective parameters is based on the consideration that the integration of q and p_θ mimics the structure of a VAE. Hence, as described in Section 2.1.1, the ELBO can be implemented to approximate the log-likelihood concerning the reference data sample x_0 [Luo, 2022, Sohl-Dickstein et al., 2015]:

$$\log p_\theta(x_0) \geq \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right] = \text{ELBO}$$

By multiplying with -1, the maximizing objective is transferred into a minimizing objective [Ho et al., 2020, Luo, 2022]:

$$\begin{aligned} -\log p_\theta(x_0) &\leq \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right] \\ &= \mathbb{E}_{q(x_{1:T}|x_0)}\left[-\log p(x_T) - \sum_{t\geq 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right] \end{aligned}$$

The objective can be further rewritten as follows [Ho et al., 2020, Luo, 2022]:

$$\begin{aligned} L = \mathbb{E}_{q(x_{1:T}|x_0)}[&\underbrace{D_{\mathrm{KL}}(q(x_T|x_0)||p(x_T))}_{\text{prior matching term}} + \sum_{t=2}^{T} \underbrace{D_{\mathrm{KL}}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))}_{\text{denoising matching term}} \\ &- \underbrace{\log p_\theta(x_0|x_1)}_{\text{reconstruction term}} \ ] \end{aligned}$$

The derived form of the ELBO can be interpreted in terms of its individual components [Ho et al., 2020, Luo, 2022]:

- $\mathbb{E}_{q(x_T|x_0)}[D_{\mathrm{KL}}(q(x_T|x_0)||p(x_T))]$ acts as the prior matching term. It reaches its lowest when the last latent distribution is in sync with the Gaussian prior. Since this term does not have parameters that can be trained and the assumption is that T is large enough for the ending distribution to be Gaussian, its value practically reduces to zero.

- $\mathbb{E}_{q(x_t|x_0)}[D_{\mathrm{KL}}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))]$ equates to the denoising matching term. It aims to establish consistency in the distribution at x_t from both the preceding and the succeeding steps. This means that a denoising action on a more distorted image should correspond to the noise introduction action on a clearer image at every intermediate stage. This relationship is mathematically represented by the KL Divergence. Minimization occurs when $p_\theta(x_t|x_{t+1})$ aligns with the gaussian distribution $q(x_t|x_{t-1})$.
- $\mathbb{E}_{q(x_1|x_0)}[\log p_\theta(x_0|x_1)]$ serves as a reconstruction term. It predicts the log likelihood of the initial data sample based on the initial latent step. This is a familiar term also seen in a standard VAE and its training process is comparable. Ho et al. [2020] employ a distinct discrete decoder stemming from $N(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$ to compute this component of the loss.

In order to parameterize the training loss of the denoising matching term, the following steps must be traversed. During training, a neural network has to approximate a conditional probability distribution.

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

Utilizing Bayes as well as the property derived in Section 2.2.1, the mean and the variance can be expressed as follows [Ho et al., 2020]:

$$\begin{aligned}
\mu_\theta(x_t, t) &= \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\right)\epsilon_\theta(x_t, t) \\
\sigma^2 = \tilde{\beta}_t &= \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t
\end{aligned}$$

Consequently, by assigning $\Sigma_\theta(x_t, t) = \sigma_t^2 I$ the loss for the denoising matching term can be formulated, accordingly [Ho et al., 2020]:

$$\begin{aligned}
L_t &= \mathbb{E}_{x_0,\epsilon}\left[\frac{1}{2\sigma_t^2}\left\|\tilde{\mu}(x_t, x_0) - \mu_\theta(x_t, t)\right\|^2\right] \\
&= \mathbb{E}_{x_0,\epsilon}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t\sqrt{1-\overline{\alpha}_t}}\left\|\epsilon_t - \epsilon_\theta(\sqrt{\overline{\alpha}_t}x_0 + \sqrt{1-\overline{\alpha}_t}\epsilon_0, t)\right\|^2\right] \\
&= \mathbb{E}_{x_0,\epsilon}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t\sqrt{1-\overline{\alpha}_t}}\left\|\epsilon_t - \epsilon_\theta(x_t, t)\right\|^2\right]
\end{aligned}$$

Ho et al. [2020] have shown that the simplified version of this term is more efficient as detailed below:

$$L_t = \mathbb{E}_{x_0,\epsilon}\left[\left\|\epsilon - \epsilon_\theta(x_t, t)\right\|^2\right]$$

In this context $\epsilon_\theta(x_t, t)$ is a neural network that aims to forecast the foundational noise ϵ_0 drawn from a standard normal distribution, which influences the transformation from x_0 to x_t. This reveals that a reconstruction of the initial image x_0 is analogous to train a DM to estimate the noise.

Notably, this loss function was derived from the method of Denoising Diffusion Probabilistic Models (DDPM), as introduced by Ho et al. [2020]. Drawing from Diffusion Probabilistic Models [Sohl-Dickstein et al., 2015], Kingma et al. [2021] analyze the DMs' ELBO. In continuous-time settings, they present an invariance of the generative model and its ELBO concerning the diffusion process. They demonstrate that multiple DMs from various studies are essentially the same, differing only in a time-dependent data rescaling. Utilizing this knowledge, an equivalence between several models, previously introduced in academic literature, can be proven including DDPM. Respectively, this expression of the loss can be considered as being universal across various DMs, under the condition that they comply with the following constraints [Luo, 2022]. The DM has the same dimensions for data and latent space, uses a pre-defined linear Gaussian model for its latent encoder at each timestep and the Gaussian parameters evolve such that the latent distribution becomes a standard Gaussian at the final timestep.

The underlying principle implies, while different formulations might be used, their behavior in continuous-time will lead to equivalent results, as they converge towards the same generative distribution. Hence, understanding the details of one model can provide insights applicable to a broader array of DMs. This not only simplifies the field of diffusion-based generative models but also provides a unified perspective [Kingma et al., 2021].

2.2.4 Conditional Image Generation

The methodology in this work involves guided image generation anchored by an underlying textual prompt. Guided DMs aim to condition the sampling process, enabling the generation of specialized sample types, often predicated on an additional input C, such as class labels or textual embeddings. In its mathematical essence, this conditional transformation evolves an unconditional DM $p_\theta(x)$ into its conditional counterpart $p_\theta(x|C)$ by incorporating the conditioning information C throughout every diffusion step [Luo, 2022]:

$$p_\theta(x_{0:T}|C) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t, C)$$

The classifier-guided [Dhariwal and Nichol, 2021] approach leverages an auxiliary model, typically referred to as a classifier. This classifier serves the purpose of providing gradients which steer the diffusion trajectory. This method requires the additional step of training an independent classifier, increasing computational requirements.

Classifier-Free Guidance [Ho and Salimans, 2022] is an approach designed to improve conditional DMs by overcoming some of the limitations inherent in Classifier Guidance methods. Unlike Classifier Guidance, which requires the simultaneous learning of a DM and a separate classifier, Classifier-Free Guidance leverages both a conditional $\epsilon_\theta(x_t|C)$ and an unconditional $\epsilon_\theta(x_t|\varnothing)$ denoising model. For this purpose it is sufficient to train a singular entity. This is achieved by replacing the conditioning information with constant values, effectively performing random dropout on the conditioners.

During sampling, the following linear combination is used:

$$\hat{\epsilon}_\theta(x_t, t, C) = s \cdot \epsilon_\theta(x_t, t|C) + (1 - s) \cdot \epsilon_\theta(x_t, t|\varnothing)$$

Here, the term s determines how the learned model accounts for the conditioning information C during the diffusion mechanism.

- When $s = 0$: The model disregards the conditioning information, behaving like an unconditional DM.
- When $s = 1$: The model fully internalizes the conditioning.
- When $s > 1$: The model gives precedence to the conditional data while also deviating from the unconditional function. This yields samples more congruent with the conditioning while simultaneously decreasing the sample variety.

To summarize, while Classifier Guidance leverages auxiliary models to focus diffusion towards specific objectives, Classifier-Free Guidance elegantly accomplishes this within a unified architecture. Given that this research aims to generate images conditioned on specific textual prompts, the Classifier-Free Guidance technique is the utilized approach. Therefore, the training objective derived in Section 2.2.3 can be specified, including the conditional information:

$$L_t = \mathbb{E}_{x_0,\epsilon}\left[\|\epsilon - \hat{\epsilon}_\theta(x_t, t, C)\|^2\right]$$

2.2.5 Denoising Model Architecture

After covering the foundational functionalities of DMs in order to perform image sampling, DMs require a neural architecture that can accept noised images at specific time steps and output the corresponding predicted noise. Particularly, the predicted noise is a tensor that has the same size and resolution as the input image. For that reason both the input and output must share the same spatial dimensions over various timesteps. For this purpose Ho et al. [2020] employed an architecture proposed by Ronneberger et al. [2015], namely the U-Net.

In its design, the U-Net (Figure 2.6) embodies principles similar to those of an autoencoder. As described in section 2.1, an autoencoder, traditionally used in introductory deep learning contexts, compresses input data into a smaller 'bottleneck' hidden representation and then decodes it back to its original shape. The compression forces the model to retain only the most salient information. Similarly, the U-Net downsamples the input image, reducing its spatial resolution and then upsamples it to restore its original dimensions. What sets the U-Net apart from traditional autoencoders is its use of skip or residual connections between encoder and decoder layers of matching feature dimensions. These connections enhance the gradient flow and prevent the U-Net from losing important information while downsampling, a design inspired by the ResNet model [He et al., 2016]. In order to incorporate the diffusion timestep t a positional embedding is added into each residual block [Ho et al., 2020].

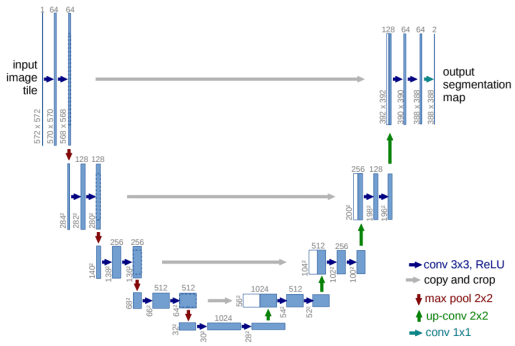


Figure 2.6: The U-Net architecture. Created by Ronneberger et al. [2015].

At this point, it is relevant to introduce the concept of schedulers. Schedulers are algorithms operating alongside the U-Net component to regulate the denoising process [Patil et al., 2022]. These algorithms manage the denoising process by adjusting the noise levels at each time step. This allows for a bal-

anced approach between speed and quality in the image generation process. Schedulers, also known as Samplers, are integral to shaping the characteristics and aesthetics of the generated images.

Two common scheduler techniques are DDPM (Denoising Diffusion Probabilistic Models) introduced by Ho et al. [2020] and DDIM (Denoising Diffusion Implicit Models) by Song et al. [2021a]. Notably, DDIMs operate 10 to 50 times faster than DDPMs. Furthermore, the scheduling algorithm proposed by Karras et al. [2022] significantly enhances training convergence speed and produces images of superior quality compared to earlier methods. In this research, the latter technique is employed for implementing the image guidance methods.

2.2.6 CLIP Text Encoder

To facilitate image generation that is conditioned on textual descriptions, the conversion of text into a suitable embedding space is imperative. Within the framework of this research, the CLIP text encoder serves this role. Unlike generic embeddings, CLIP (Contrastive Language-Image Pretraining) [Radford et al., 2021] offers a nuanced encoding that captures both the content and style nuances articulated in the textual prompt.

CLIP simultaneously trains an image encoder and a text encoder, creating a shared embedding space that dynamically links visual and linguistic data. Trained on a vast dataset of around 400 million image-text pairs, CLIP's objective is to maximize the alignment of authentic image-text pair embeddings while minimizing it for non-matching pairs. Consequently, the models text encoder becomes a pivotal component in generating contextually-aware images.

Deciphering CLIP Embeddings for Image Generation

In the text-to-image process, a prompt is converted by a text encoder into an embedding. Since Stable Diffusion is deployed, this research specifically focuses on the CLIP text encoder for this purpose. By strategically modifying the embedding, a more desirable image can be produced. Consequently, it is crucial to gain insight into the structure of CLIP embeddings before developing the image guidance techniques.

Prior to the implementation of the proposed methods, an empirical analysis was conducted to comprehend the embeddings generated by CLIP. This analysis utilized the `large_random_1k` subset of prompts from the dataset

described in Section 4.3. This investigation reveals that the tokenizer of the CLIP model begins processing the input text by assigning a unique starting token, identified by the number 49406, to the text input. This starting token is pivotal as it signals the beginning of a text sequence to the model. Every input sequence is transformed into an embedding with a uniform size of 77×768 dimensions. It is important to note that the embedding for this starting token remains invariant across different instances, ensuring that it consistently conveys the initiation of a sequence. The embedding for the end-of-text token, numbered as 49407, fills the role of a padding mechanism.

The sequential embeddings in the CLIP model are cumulative in nature. Specifically, the second entry in the embedding sequence contains information regarding its corresponding and the first token. Each successive embedding then incorporates the context from all previous tokens in addition to its own. This cumulative embedding process ensures that each entry within the 77-dimensional embedding array carries forward the contextual information of all preceding tokens.

For the seed-invariant optimization technique introduced in Section 6, this cumulative property is pivotal. It allows for a significant reduction in the parameter space. The embedding of the last token in the conditional sequence, by virtue of encapsulating the context of all preceding tokens, is replicated 76 times following the start token. This creates a conditional embedding, which can be utilized during the optimization process. The robustness of this method is evident in the consistency of image outcomes. Images generated from embeddings of the last token in such a repetitive fashion remain contextually aligned to those produced from the initial embedding.

The constancy of the start token's embedding is crucial. Throughout the optimization procedure, despite multiple updates, the embedding of the start token must remain unchanged. This immutability is essential as any substantial modification to this embedding vector can introduce increasing noise levels within the generated images. Such noise increase could culminate in the production of images that are entirely noise. To prevent this, the embedding vector of the start token is not subjected to modification during the update process. This approach ensures the preservation of image integrity and allows for the optimization of the upfollowing embedding's segments.

2.2.7 Computational Constraints of Diffusion Models

DMs have emerged as a powerful class of generative models, achieving state-of-the-art results in image synthesis, but their operations directly in pixel space have resulted in substantial computational resource demands. The training process for these models often consumes hundreds of GPU days [Dhariwal and Nichol, 2021] due to the high capacity allocated for capturing even imperceptible details in the data [Salimans et al., 2017]. Moreover, inference is also resource-intensive. Generating a moderate number of samples can take up to five days on advanced GPUs [Dhariwal and Nichol, 2021]. This not only restricts the usage of DMs to those with access to extensive computational resources, but also raises environmental concerns due to the high energy consumption [Patterson et al., 2021, Strubell et al., 2020]. In addition, the inherently sequential nature of these models, requiring multiple iterative steps for both training and inference, further compounds the computational demands.

2.2.8 Latent Diffusion Models

Addressing the challenges with the computational constraints of DMs, Rombach et al. [2022] proposed the approach of Latent Diffusion Models (LDMs) to which Stable Diffusion adheres. LDMs introduce a shift by moving the training and operational processes from the pixel space to the more manageable lower-dimensional latent space of pretrained autoencoders. Due to the fact that most of the image's bits contribute to perceptual details, the semantic and conceptual composition remains intact even after significant compression. LDMs effectively separate perceptual compression and semantic compression through generative modeling learning. It begins by eliminating pixel-level redundancy using an autoencoder and subsequently generates semantic concepts through a diffusion process applied to the learned latent space.

Utilizing an encoder $\mathcal{E}$, the input image $x \in \mathbb{R}^{H \times W \times 3}$ is transformed into a compact latent representation $z = \mathcal{E}(x) \in \mathbb{R}^{h \times w \times c}$ with a compression rate defined as $f = H/h = W/w = 2^m$, where $m \in \mathbb{N}$. Reconstruction of the image is facilitated by the decoder $\mathcal{D}$ translating the latent representation back into the image space, yielding $\tilde{x} = \mathcal{D}(z) \in \mathbb{R}^{H \times W \times 3}$.

The loss computation for LDMs mirrors that of DMs but operates within the more compact latent domain:

$$L_{LDM} = \mathbb{E}_{\mathcal{E}(x),t,\varepsilon} \left[\|\varepsilon - \hat{\varepsilon}_\theta(x_t, t, C)\|^2 \right]$$

Augmenting the architecture, the denoising U-Net now incorporates cross-attention layers. This enables LDMs to adept at processing a range of conditioning inputs, spanning from text descriptors to bounding boxes. This consolidation enables them to undertake sophisticated operations like high-resolution synthesis.

2.3 Image Generation with Stable Diffusion

After laying the conceptual and technical foundations of DMs, including their variant LDMs, the attention can be shifted to the image generation process facilitated by Stable Diffusion. This process is paramount to the image guidance methodologies presented in this thesis. Grounded in the principles of LDMs, Stable Diffusion integrates three key components: the CLIP text encoder, the denoising U-Net and a VAE. The forthcoming approach on image generation is depicted in Figure 2.7.

Given a set seed in the system, initial latents corresponding to a standard normally distributed tensor of size 64×64 can be sampled. Simultaneously, the prompt is transformed into a text embedding via a pretrained text encoder, which is CLIP in this context. Both the Gaussian-distributed seed latent and the text embedding function as inputs for Stable Diffusion. Using the latent seed, an image representation is constructed in the latent space while retaining the latent seed's dimensions.

As previously explained, the responsibility of denoising these latent image representations falls to the U-Net architecture. It conducts this task iteratively, conditioned on the text embeddings derived earlier by the CLIP text encoder. The output from the U-Net is equivalent to the noise residual, which aids in computing a less noisy image latent representation using scheduling algorithms, including those described in Section 2.2.5.

After this denoising phase the latent representation is passed to the VAE decoder, culminating in the final image rendition.

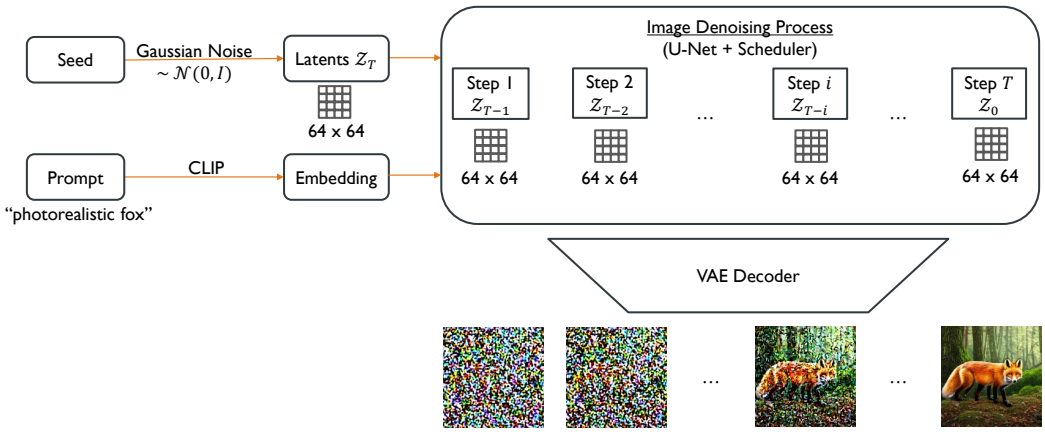


Figure 2.7: Image generation process with Stable Diffusion.

2.4 Interpolation

Finally, the concept of interpolation is thematized as a fundamental technique for the methodologies realized in this thesis. Essentially, it serves as a mechanism to estimate values within a known range. In the context of the implemented image guidance techniques, it is employed for prompt embeddings and latents initialized from different seeds. This is followed by an exploration of its application, specifically in the context of text embeddings and latent variables.

The role of interpolation techniques in this study becomes evident as they are integrated into different approaches of the research. Their applications include assuring a consistent continuity in embeddings for gradient-based optimizations in the context of metric-based image optimization (Section 4), facilitating expansive optimization steps as part of the user interaction method (detailed in Section 5) and concluding with the seed invariant prompt embeddings (Section 6). In the latter, interpolation of the initial seed latents introduces a measured degree of randomness, optimizing the process to produce seed-independent images.

2.4.1 Utilized Methods

Interpolation is a mathematical technique that allows the estimation of values situated between two given points. Throughout the proposed image guidance methods, three interpolation techniques are considered. These are: Linear Interpolation (LERP), Normalized Linear Interpolation (NLERP) and the Spherical Linear Interpolation (SLERP) [Shoemake, 1985]. For a visual under-

standing and comparison of these methods, Figure 2.8 graphically illustrates the interpolation approaches covered in this section.

LERP determines intermediary values by tracing the direct path connecting two endpoint vectors. Given two vectors v_i and v_j, representing the start and end points and an interpolation parameter $s \in [0, 1]$, the corresponding LERP formula can be expressed as:

$$\mathrm{LERP}(v_i, v_j; s) = (1 - s)v_i + sv_j.$$

For situations involving standardized data, NLERP comes into play. This method begins by conducting a LERP operation. Afterwards, a normalization of the resulting vector is performed, maintaining a uniform length. A distinctive characteristic of NLERP is that, due to the consistent lengths of the vectors, the interpolated points are not evenly spaced. When increasing the interpolation parameter in constant step sizes, the interpolated vector is moving more rapidly as it approaches the midpoint, given the increased distances it must traverse compared to the vector's extremities [Kremer, 2008].

An alternative to the previous interpolation methods is SLERP. Unlike its counterparts, SLERP interpolates by executing a rotational movement along the shortest path on a unit sphere connecting two endpoints. The mathematical representation for SLERP, using the vectors v_i and v_j, is:

$$\mathrm{SLERP}(v_i, v_j; s) = \frac{\sin((1 - s)\Omega)}{\sin \Omega} v_i + \frac{\sin(s\Omega)}{\sin \Omega} v_j.$$

While SLERP might demand more computational resources compared to NLERP, it is enabling a smoother interpolation [Kremer, 2008].

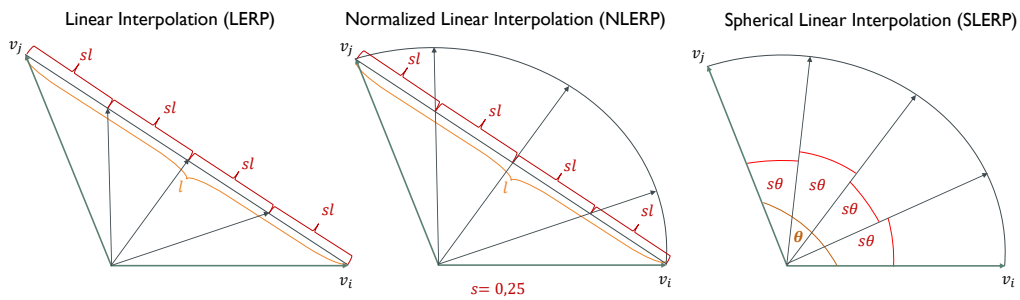


Figure 2.8: Considered interpolation methods.

2.4.2 Embedding Interpolation

In the context of image generation through Stable Diffusion, CLIP embeddings serve as indispensable tools for fine-tuning of textual embeddings. To accomplish notable modifications in a single step, interpolation techniques are applied between two distinct prompt embeddings.

While prior research has utilized LERP [Tevet et al., 2022] and SLERP [Ramesh et al., 2022a] for smooth interpolation, LERP presents challenges when applied to CLIP embeddings. Specifically, due to their standardized distribution, LERP can generate embeddings with norms that deviate from a initial range, potentially leading Stable Diffusion to produce images containing undesirable artifacts. This issue becomes particularly significant when the output of LERP serves as an input for additional interpolation processes. A viable workaround to this limitation is the use of NLERP to adjust the norms of interpolated embeddings. The usage of SLERP is also a prevalent option and acknowledged as an effective interpolation technique for prompt embeddings [Han et al., 2023].

Figure 2.9 illustrates an interpolation example using SLERP between embeddings. The resultant images generated via Stable Diffusion convincingly interpolate both style and content, underlining the efficacy and robustness of both CLIP embeddings and Stable Diffusion.

Further, the Stable Diffusion model defines a continuous mapping from the prompt embedding space to the image space. This continuity implies that minor alterations in the embedding space yield correspondingly small changes in the generated images, both in terms of pixel values and perceived aesthetic qualities. To leverage this property, the proposed methods perform incremental adjustments to prompt embeddings. Marginal adjustments are executed through gradient descent. For more substantial changes, direct SLERP interpolation between two prompt embeddings is employed. This offers fine-grained control over modifications in the prompt embedding space.

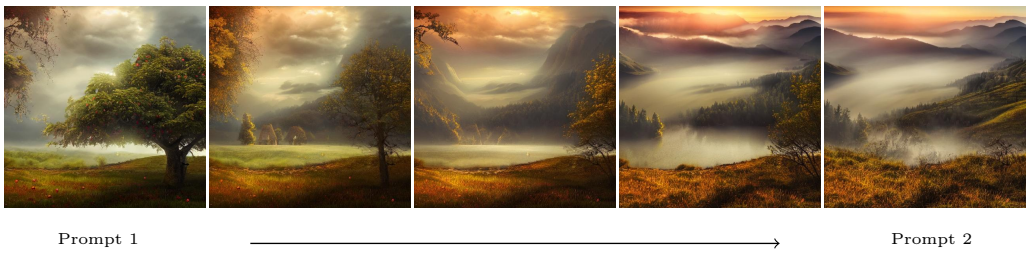


Figure 2.9: Interpolation between two prompt embeddings. The prompts are contained in Appedix A.

2.4.3 Latent Interpolation

In the context of computing seed invariant embeddings, interpolation within a normally distributed latent space is conducted. As shown by White [2016] utilizing LERP is inappropriate in such a high-dimensional normally distributed environment. This is due to the consistent variances and equal lengths of the two endpoints. Employing LERP would cause a reduction in vector length as the interpolation approaches the center, thereby leading to a decrease in variance. As a result the images become blurry as shown in Figure 2.10. This image is generated based on linearly interpolated seed latents, leveraging Prompt 3 as presented in Appendix A including the corresponding seeds. In alignment with Section 2.4.2, NLERP offers an alternative by adjusting the vectors' length and simultaneously the variances. SLERP is advocated in this context as the more effective technique, facilitating a smoother interpolation process.

Figure 2.10: Resulting image after performing LERP on standard normally distributed initial seed latents.

Chapter 3

Related Work

In recent years, the automation of prompt engineering has seen widespread application not only in image generation but also in the enhancement of techniques across various tasks, for instance in the field of natural language processing (NLP), including text generation and classification. Therefore, this chapter covers two main areas. The first part covers the methods of automated prompt engineering in NLP, highlighting it as an additional field, where the challenges of prompt engineering are addressed. The second part explores strategies that steer the image guidance process of generative text-to-image models.

3.1 Automated Prompt Search for NLP Applications

A distinction within the area of automated prompts generally lies in their separation into two categories [Liu et al., 2023]. Discrete prompts correspond to an actual text string, while continuous prompts refer to prompts that are directly represented within the embedding space of the present language model.

In the area of discrete prompts, Wallace et al. [2019] introduced a method employing gradient-based search over actual tokens to identify short sequences that effectively guide pre-trained language models toward generating desired target predictions. This iterative search process involves navigating through tokens in the prompt, leading to the discovery of sequences that trigger specific model outputs. Building upon this foundation, Shin et al. [2020] advanced the concept of automated prompt engineering by proposing AutoPrompt. This technique automatically generates natural language prompts for a variety of NLP tasks, such as sentiment analysis, natural language inference, fact re-

trieval and relation extraction. It exploits the knowledge embedded in pretrained masked language models (MLMs) by finding a sequence of discrete trigger tokens that can induce the MLM to produce the desired output. These trigger tokens are selected by a gradient-guided search that measures the sensitivity of the MLM output to each input token. AutoPrompt shows that MLMs have an innate ability to perform diverse tasks without requiring additional parameters or fine-tuning.

Another work that optimizes discrete text prompts is RLPROMPT [Deng et al., 2022], which uses reinforcement learning (RL) to explore the prompt space. RLPROMPT employs a parameter-efficient policy network that generates the optimal discrete prompt after training with a reward signal. RLPROMPT evaluates its performance on both classification and generation tasks, such as few-shot text classification and unsupervised text style transfer. RLPROMPT differs from AutoPrompt in that it uses RL and does not rely on label tokens to guide the prompt optimization.

Expanding on continuous prompts, Zhong et al. [2021] proposed a two-step method that first uses a discrete search approach, such as AutoPrompt. This step involves the establishment of virtual tokens based on the identified discrete prompt, subsequently followed by the fine-tuning of embeddings to increase task accuracy. This method leverages the effectiveness of manually crafted templates, but may not be able to capture the context information of the input.

Tang et al. [2022] proposed an approach called context-tuning, which aims to fine-tune pretrained language models (PLM) for natural language generation. Unlike previous methods that use fixed or independent prompts, context-tuning generates contextualized prompts that are derived from the input text and encode its context information. These prompts can obtain useful knowledge from PLMs for generation and adapt to different inputs dynamically. The authors showed that context-tuning outperformed existing prompting methods on several text generation tasks, such as summarization, dialogue and story generation.

While these discrete and continuous approaches for automated prompt engineering have been mainly applied to natural language generation tasks, this thesis focuses on text-to-image guidance, which poses different challenges and opportunities. In particular, the prompt is updated in a continuous manner to manipulate the embeddings of Stable Diffusion prompts, enabling a more user friendly experience, when using generative text-to-image models like Stable Diffusion without the need for prompt based editing.

3.2 Image Generation Guidance

In order to provide control in generative text-to-image models without relying on prompt engineering, the diffusion process itself offers some control by inpainting. This technique operates by reconstructing missing or damaged areas in images. Its applications range from image restoration and object elimination to artistic creation. Image inpainting usually involves a generative model that learns to fill in the missing regions with realistic and coherent content, based on the surrounding pixels or some guidance signals.

Nevertheless, inpainting alone may not be sufficient when tasked with capturing novel concepts or replicating specific appearances based on user input. Therefore, in the recent years some techniques have been proposed to enhance the guidance and personalization of the text-to-image process by using additional signals, such as Textual Inversion [Gal et al., 2022]. Leveraging this concept, personalization is carried out by learning new words in the embedding space of the text encoder from a few example images. To accomplish this, a set of images displaying a target concept is inverted using a DM to obtain a diffusion trajectory. Afterwards, it is reconstructed by optimizing the loss function of the utilized DM, simultaneously the weights of both the denoising model and the text encoder are preserved. Inspired by this approach, two inversion techniques have emerged, enabling a precise editablity of images.

Similar to Textual Inversion, Null-text inversion [Mokady et al., 2023] reconstructs the image through a DM to derive a diffusion path. Subsequently, only the unconditional textual embedding is used as a classifier-free guidance signal for optimization. The unconditional embedding is learned by employing the exact same method as Textual Inversion. Finally, the inverted image can be modified through a prompt based editing technique proposed by Hertz et al. [2023], which will be outlined in the following.

Prompt Tuning Inversion [Dong et al., 2023] operates in two stages. In the reconstruction stage the image is also inverted by applying a DM. This time the conditional embedding is optimized during the reconstruction of the image. In a subsequent editing stage the prompt embedding is modified by interpolation with the target text embedding, obtained from the user defined target prompt.

In response to the limitations of these methods, including the generation of poor results in selected areas and unexpected modifications in untouched regions as well as the need to provide detailed prompts, Li et al. [2023] introduced StyleDiffusion as an enhanced concept, internalizing adventages of the previous techniques. Despite its advancements, even StyleDiffusion does not completely eliminate the need for prompt editing.

An alternative way to achieve personalization is fine-tuning. DreamBooth [Ruiz et al., 2023] fine-tunes text-to-image DMs for subject-driven generation from a concise amount of reference images.

Since fine-tuning of DMs can lead to overfitting when handling several subjects, Han et al. [2023] proposed SVDiff. By fine-tuning a compact amount of a DM's parameters they reduce the risk of overfitting and language drifting, providing a multi-subject personalization technique. For this purpose they perform a singular value decomposition on the weight matrices of the DM.

Black et al. [2023] adapt the DM's weights by employing RL. The reward function is defined based on an aesthetic metric, obtained by the usage of human generated image rating pairs, assessing the aesthetic score of an image.

Likewise, Wu et al. [2023] presented a method utilizing an aesthetic metric. Their technique aims to align the resulting images with human preferences. Therefore, they fine-tuned CLIP leveraging a data set of human choices from generated images by Stable Diffusion and determined an aesthetic metric based on it.

The latter has a direct effect on the prompt embeddings but does not allow for individual adjustments of single prompts. Furthermore, the outcome of fine-tuning is a static model without the flexibility to leverage other metrics.

In the realm of interactive methods incorporating user feedback, Hertz et al. [2023] introduced Prompt-to-Prompt. This makes use of a cross-attention mechanism that allows the model to control the relation between the spatial layout of the image to each word in the prompt. They created a precise image editing framework by performing modifications on the prompt only. This way most of the original image can be preserved when performing small changes in a prompt.

Brooks et al. [2023] pursue the DM InstructPix2Pix, which is able to update images by following instructions. These are expressed independently of the prompt. This can be accomplished by generating a large training set of image editing examples utilizing Prompt-to-Prompt in combination with a language model and a text-to-image model.

Another interactive approach is FABRIC [von Rütte et al., 2023]. This technique is a training-free method aimed at improving the output of diffusion-based text-to-image models by incorporating iterative human feedback. It utilizes the self-attention layers common in many DMs to condition the image generation process based on feedback images. This technique allows the model to focus on certain features or details from a reference image during the denoising process, leading to more tailored and refined outputs.

In order to guide DMs based flexibly on different input modalities, Zhang and Agrawala [2023] proposed ControlNet, a neural network architecture designed to add spatial conditioning controls to large, pretrained text-to-image DMs. This essentially enables users to influence or guide the output of these DMs using some conditions like edge, depth, segmentation or human pose. To incorporate these additional conditional inputs often a reference image is required, which can be considered as a drawback of this method.

Huang et al. [2023] persued a similar objective by proposing Collaborative Diffusion, a method for generating faces from multiple additional input modalities.

Lastly, the scope of influencing DMs has dramatically expanded, even including the use of neural signals as inputs rather than traditional text prompts, as demonstrated by Takagi and Nishimoto [2023].

Chapter 4

Metric Based Optimization

Generating images that precisely align with envisioned concepts has always posed challenges, particularly when striving for intricate details and specific nuances. This divergence underlines the importance of making targeted adjustments, especially in scenarios where desired refinements are not easy to express verbally, such as aiming for a specific aesthetic. As a consequence the users add descriptors to the prompt like `4k high resolution` to find a satisfying image. However, the effect of these descriptors does not provide enough control. With appended phrases, not always a satisfying visual outcome can be created. To address this, the proposed approach does not merely rely on ad-hoc textual changes. Instead, it optimizes the underlying embedding of the textual prompt based on a predefined differentiable metric computed in the image space. If a user's desired style can be programmatically described by this metric, this methodology refines the embedded representation of the given prompt to generate better images.

Therefore, this chapter delves into the approach of metric based image generation to overcome such challenges, aiming to refine the textual prompts for more accurate results. The subsequent sections elaborate on the methodological approach and the implementation specifics, followed by the setup of the experiments and concluding with the results derived from the experiments.

4.1 Methodological Approach

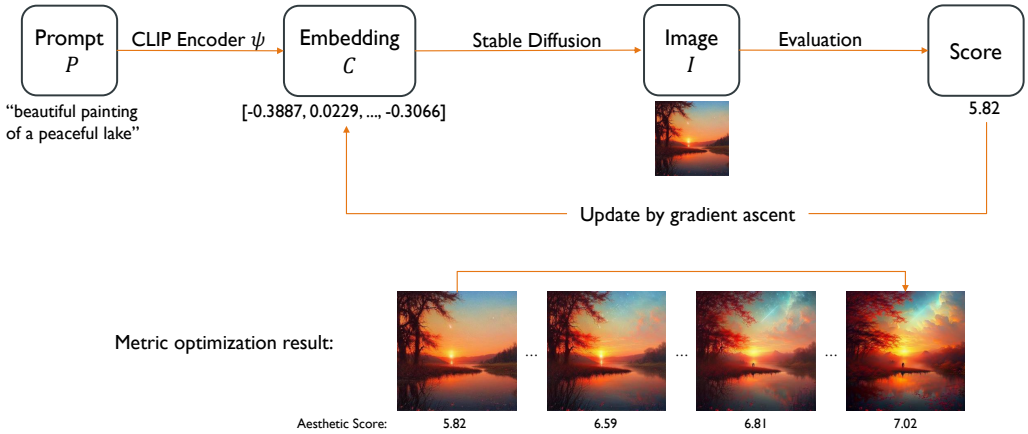


Figure 4.1: Iterative improvement through a metric based optimization process. The aesthetic scores beneath each image serve as examples, showcasing the incremental enhancements in image quality as perceived by a computational evaluation metric.

Traditionally, an image represented as I is derived from a prompt P by embedding the prompt and subsequently applying the Latent Diffusion Model:

$$I = \mathrm{LDM}(\psi(P))$$

Leveraging a given metric m, the prompt embedding is optimized by employing gradient descent.

$$C^* = \arg\min_C m(\mathrm{LDM}(C))$$

Alternatively, gradient ascent is utilized if the metric optimization indicates an improvement with rising values.

$$C^* = \arg\max_C m(\mathrm{LDM}(C))$$

Here, the initial embedded prompt is $C = \psi(P)$. The resulting optimized image is represented as $I^* = \mathrm{LDM}(C^*)$.

It is crucial to highlight, that during this optimization the model's weights remain constant, preventing comprehensive model fine-tuning. The advantage of this approach lies in its ability to make minimal adjustments to the prompt embedding, ensuring that most characteristics of the resultant image remain consistent, while still precisely meeting the predefined metric. For visualization of the results' reliability, a fixed seed is maintained throughout the procedure.

4.2 Utilized Metrics

The metric based image optimization approach has been implemented using the following metrics:

- **Blurriness and Sharpness:** The blurriness or sharpness of an image is calculated by turning it to grayscale and applying the extended Laplacian kernel, utilized to detect edges in images. After the convolution, the variance of the resulting filtered image is computed. In this context, a high variance indicates a sharp image with pronounced edges, while a low variance suggests a blurrier image with fewer clear edges. Accordingly, by performing gradient ascent the sharpness of an image increases, while gradient descent leads to a more blurry result.
- **Deep-learning based Aesthetic Metric:** The LAION aesthetic predictor [Schuhmann, 2022a,b] is a multi-layer perceptron, trained on 176.000 image rating pairs. The ratings range from 1 to 10. An image's aesthetic score is computed by embedding the image with the CLIP model and afterwards feeding it to the linear model trained on human ratings.

For all implemented metrics their gradients can be computed, making them appropriate candidates for this optimization technique.

4.3 Dataset

The DiffusionDB [Wang et al., 2023], with particular emphasis on subsets `large_random_100k` and `large_random_1k`, provides the data foundation for this assessment. DiffusionDB stands as the first large-scale dataset for text-to-image prompts, encompassing 14 million images. These images are produced by Stable Diffusion, guided by prompts and hyperparameters determined by actual users. The images showcased in the figures are obtained based on the initial prompts sourced from Lexica[1], a repository known for its well designed prompts and corresponding images.

[1] https://lexica.art

4.4 Implementation

This section outlines the practical aspects involved in implementing the method described in Section 4.1. While the theoretical approach serves as the foundation, several adaptations were necessary during the transition to actual implementation. These adaptations were driven by computational constraints and empirical findings. Subsequently, this section includes the choice of software frameworks utilized for all three approaches in this research, modifications made based on empirical data and the findings that influenced the final implementation.

For the implementation of this method, as well as the two methods presented in the subsequent chapters, Stable Diffusion 1.4 was deployed along with the K-LMS [Karras et al., 2022] scheduling algorithm, recognized as one of the standard schedulers mentioned on the Huggingface blog related to Stable Diffusion [Patil et al., 2022].

For simplification purposes, the methodological description of the metric-based optimization approach (Section 4.1) exclusively considers the conditional part of the embedding for an update through the gradient. This approach was tested using the aesthetic metric on a sample of the first 15 prompts from the dataset derived in Section 4.3, containing 150 prompts in total. This procedure was executed for 500 iterations for each prompt. While the aesthetic score for the complete embedding increased on average by 1.05 points, a significantly smaller increase of 0.29 was observed when only the conditional part of the embedding was used. According to this observation, the complete embedding, including the conditional and unconditional, was leveraged for this procedure.

It is essential to note that, due to hardware constraints, the gradient was computed only for the first denoising step. An initial hypothesis had been established that the gradient should ideally be calculated for the final denoising step, the 70th in the present implementation to be precise. The underlying assumption was that computing the gradient for this last step would exert a more pronounced influence on image optimization, which would in turn yield a higher average aesthetic score compared to calculations made for the initial denoising step. This was tested employing the same 15 prompts as described above. Contrary to this supposition, the results were not as anticipated. The average score was considerably lower, corresponding to 0.32. Consequently, for evaluating this method, the gradient was formed exclusively for the very first denoising step.

As described in Section 2.2.6 the first vector of the CLIP embedding must remain consistent. This characteristic should be retained throughout the updates made by the gradient. If the encoded start token undergoes increasing changes, the images generated start showing increasing levels of noise, eventually culminating in purely noisy images. Therefore, this part of both the conditional and unconditional sections of the embedding will never be updated.

4.5 Evaluation

The primary objective of this evaluation is to investigate the efficacy and practicality of the proposed metric-based approach in optimizing prompt embeddings. Figure 4.2 showcases a series of images generated from these optimized prompts, demonstrating the transformative effect of the method on the initial images. The utilized metric is the aesthetic score. On the leftmost side, images derived from the initial prompts are displayed. Moving rightward, there is a noticeable evolution in the images due to the increased optimization of the prompt embeddings towards the aesthetic scores.

To provide a comprehensive understanding of the different aspects of metric based optimization, Figure 4.3 details the changes for further applied metrics, more accurately for blurriness and sharpness metrics.

While the initial observations suggest promising outcomes, a comprehensive assessment is essential to validate the robustness of the method. This evaluation commences with a foundational examination of the metric-based approach, utilizing the DiffusionDB. Subsequently, a notable concern is addressed: the potential for over-optimization, which could lead to a divergence from the desired outcomes. Finally, the evaluation will consider the method's ability to generalize across various seed values.

Initial Prompt → Optimized Aesthetic Score

Figure 4.2: Aesthetics metric refinement utilizing selected prompts. The corresponding Prompts are located in Appendix B (Prompt 1 - Prompt 3).

Initial Prompt → Optimized Bluriness Score

Initial Prompt → Optimized Sharpness Score

Figure 4.3: Sharpness and blurriness metric refinement using a selected prompt (Prompt 4, Appendix B).

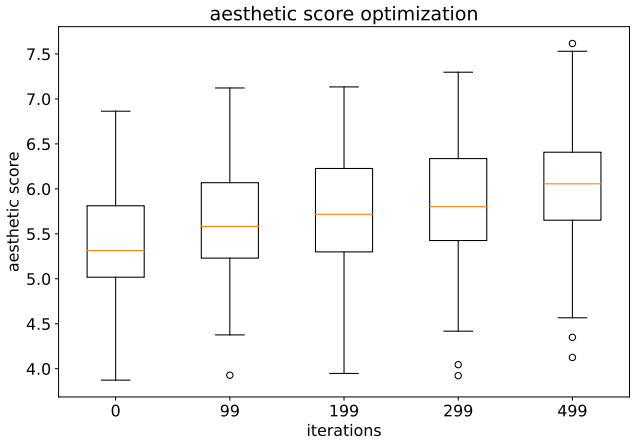
4.5.1 Empirical Assessment

150 prompts are randomly chosen from the referring dataset, described in Section 4.3. The iterative process, essential for the method's evaluation, was modulated based on the metric in focus. For instance, 500 iterations are dedicated to optimizing the aesthetic score, while the sharpness and blurriness metrics are addressed over 50 iterations each.

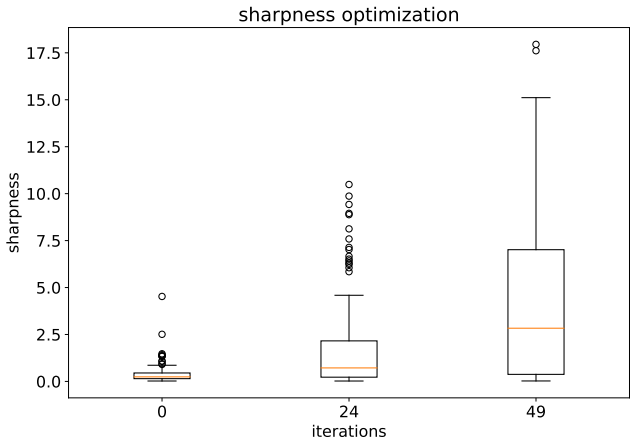
Figure 4.4 provides an overview of the optimization process for the three implemented metrics and traces the progression of the metric scores. As the evaluation unfolds, the observations align with expectations. The aesthetic score and the sharpness metric values exhibit an upward trend, whereas the blurriness metric registers a decline.

Initially, the median aesthetic score starts at 5.3 and ascends to 6.3 following 500 iterations, implying an increase of 18.9%. In contrast, the sharpness metric, despite its fewer iterations, witnesses a stronger increase compared to the aesthetic score. The median begins at 0.24 and post-optimization, ascents to 2.8 — an increment over ten times its original value. Moreover, the sharpness metric exhibits a significant growth in variance, evident in the increasing size of its boxplots. Lastly, the blurriness metric is under consideration. For visualization reasons, two outliers in a range from 2.5 to 5.0 were removed from Figure 4.4c to prevent an excessive reduction in the size of the boxplots and thus ensure a clear identification of the trend. The blurriness metric presents a converse trend to the sharpness metric. The median recedes from 0.23 to 0.11 and is half the size of the initial value. Furthermore, the variance decreases throughout the iterations, as illustrated by the shrinking boxes.

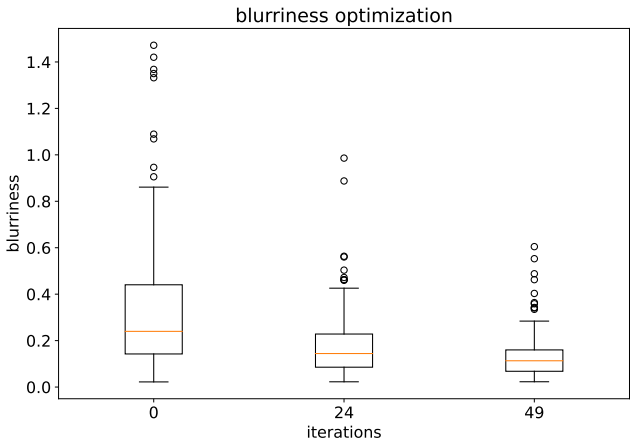
These significant changes in variance across the iterations of the optimization process indicate that both blurriness and sharpness metrics are acutely responsive to the unique features of each prompt. This implies that a consistent specification to the number of iterations may be ineffective, necessitating individualized optimization strategies. Moreover, the potential influence of a prompt's distinctiveness on the aesthetic score cannot be disregarded. A thorough visual analysis of the generated images is essential to fully grasp these effects. This phenomenon is examined in greater detail in the subsequent section.



(a) Aesthetic Score



(b) Sharpness



(c) Blurriness

Figure 4.4: Optimization progress for the implemented metrics.

4.5.2 Metric Over-Optimization

During this extensive process of examining the generated images, it was evident that certain prompts were prone to being over-optimized. This over-optimization manifests in varying extents across different prompts. To offer a visual understanding of this phenomenon, Figure 4.5 showcases some examples of over-fitted prompts towards the aesthetic score. In this figure, the first image in every row represents the visualization based on the initial prompt. The second image corresponds to the representation with the highest aesthetic score among still congruent images with the original prompt's conceptual framework. The concluding is most optimal with respect to the aesthetic score, often deviating from the original intent.

A close inspection of the optimization process corresponding to the first row reveals a trend of divergence from certain details initially defined by the prompts. The description initially included "chrome and gold" directly at the beginning. Throughout the optimization, it was observed that these elements gradually lost their prominence, until they disappeared altogether.

The second set of images presents an even more pronounced case. Not only did the motif undergo significant changes but the color scheme of the image also shifted dramatically. The blue component in the image is now more dominant than it initially was. Surprisingly, this trend of increased blue tones was observed in other sets as well, even when the motif remained closer to the provided prompt, the increase is clearly visible.

This increasing appearance of blue tones raises questions about the nature of the training data employed for the aesthetic prediction model. It potentially points towards a low diversity of particularly high-ranking images within the dataset of the aesthetic predicting model. The training set might lack a sufficient representation of such high-ranking images. Such an imbalance and low diversity might have potentially biased the model towards generating images with blue components.

The investigation continued with the analysis of the blurriness and sharpness metrics. Several instances of over-optimized prompts were observed, as depicted in Figure 4.6a and Figure 4.6b. Figure 4.6a provides an example for the blurriness metric, where the snow globe, the primary component of the prompt, entirely dissolves into a cloud-like structure. Conversely, Figure 4.6b displays the outcomes from an over-optimized prompt for the sharpness metric, where the primary motif vanishes entirely, replaced by increasingly fine lines.

Interestingly, the tendency towards over-optimization was more frequently identified in the sharpness metric, as opposed to the blurriness metric. Many of the images generated exhibited only a marginal level of blurriness, suggesting that the effect could possibly be enhanced with an increasing number of iterations amplifying the blurriness optimization.

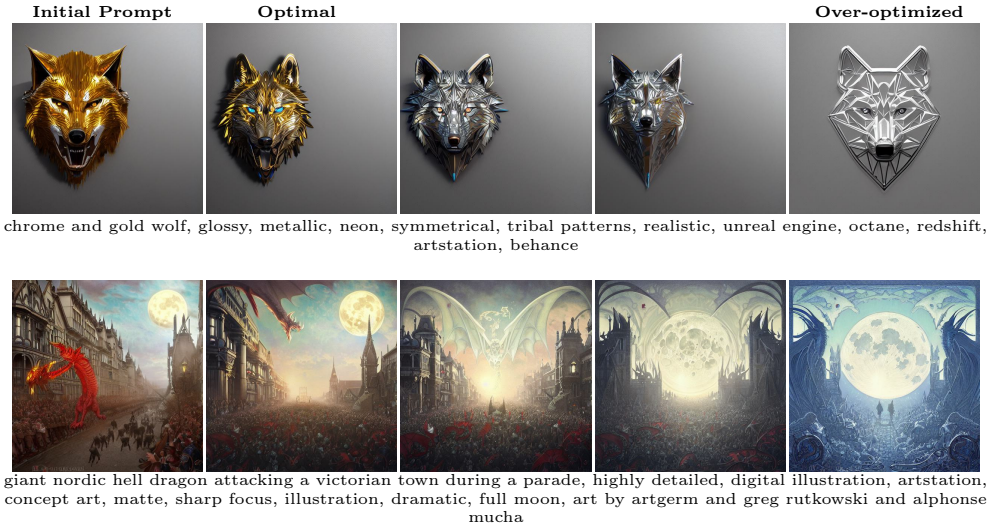
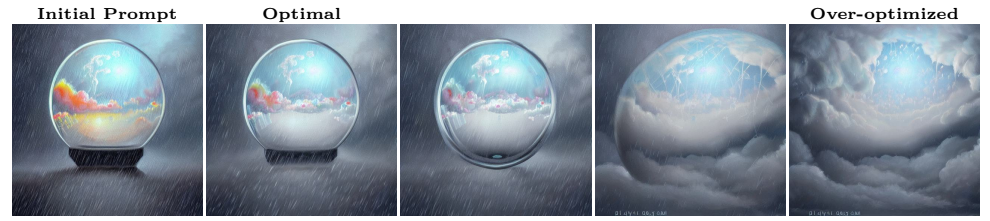


Figure 4.5: Overfitting during aesthetic score optimization.



(a) Overfitting during blurriness optimization.

(b) Overfitting during sharpness optimization.

Figure 4.6: Examples of overfitting during image optimization.

These findings underscore the potential risk of over-optimizing image prompts, where the resultant images may drift away from the original intent or description. A pivotal consideration involved setting a threshold which, when met, would signal the end of the metric optimization. The cosine similarity served as the chosen metric for this purpose. Throughout the optimization process, the similarity between the initial image and every subsequently generated image was calculated. Contrary to expectations of a steady course, the cosine similarity did not consistently decrease. There were moments when significant rises were noted. Consequently, the images generated often complied more with the target metric. The trajectory of the metric optimization was not consistent either, especially the aesthetic score showed periodic decreases. The blurriness and sharpness metrics were not entirely monotonic as well. These irregularities, paired with the inconsistent behavior of the cosine similarity, complicated the task of establishing an universally applicable stopping criterion. Therefore, determining the optimization's adequacy is at the discretion of the user. For future aspirations, it might be more insightful to explore discrepancies on a contextual level rather than merely on a visual scale. One such approach could involve leveraging models like CLIP to assess whether the generated images remain semantically consistent with the original prompt.

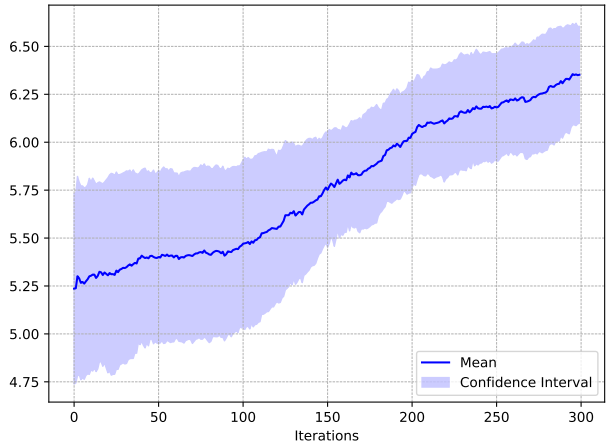
4.5.3 Generalization

The ability to alter the seed can be a valuable tool for adjusting image attributes like composition or sparking creativity. When utilizing or creating new prompt modifiers, users often desire consistent effects regardless of the chosen random seed. It is beneficial to identify prompt modifiers that operate irrespective of the seed. A similar outcome was expected for this method: Even with optimization limited to one seed, the altered prompt embedding should enhance the metric compared to the initial prompt when tested on various seeds. Using the implemented metric, the adaptability and seed independence of the method are explored. This consideration is especially evident, since users often desire prompt modifiers exhibiting the intended effects across various random seeds.

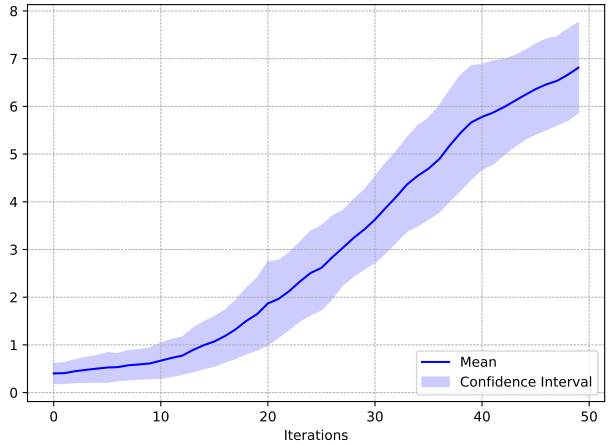
In this context, the embedding for the specified prompt `highly detailed photoreal eldritch biomechanical rock monoliths, stone obelisks, aurora borealis, psychedelic` is optimized over 300 iterations towards the aesthetic score and for 50 iterations for both, the blurriness and sharpness metrics. Subsequently, images for these optimized embeddings are generated across 65 unique seeds and the metric scores are computed. The aim is to investigate whether the modified prompt embeddings, optimized for a single

seed, can also improve the metric when applied to different seeds. This strategy measures the method's generalization ability by assessing the consistency of the metric values' trend.

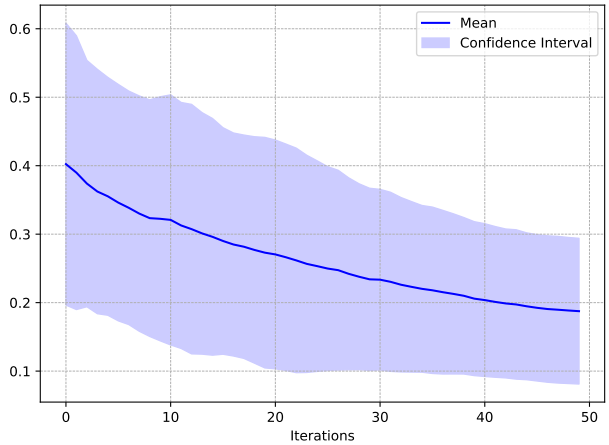
As depicted in Figure 4.7, the results not only indicate a general improvement in the metrics but also reveal a progressively narrowing confidence interval for the aesthetic and blurriness metrics. The confidence interval for the sharpness scores becomes slightly wider while still maintaining the most narrow interval as well as the strongest metric optimization compared to the others. These observations suggest that the modified prompt embeddings possess a degree of seed independence, aligning with the hope that the metric-optimization method will have a similar effect across different seeds. While more intricate optimization strategies incorporating multiple seeds during runtime can certainly be envisioned, these initial findings affirm the method's robustness in the face of varying seed values.



(a) Aesthetic Score



(b) Sharpness



(c) Blurriness

Figure 4.7: Metric-based optimization for 65 different seeds.

Chapter 5

Iterative User Interaction

Generative text-to-image models serve as useful tools for a range of creative endeavors. Often there are scenarios, where a general thematic guideline exists but no specific target image has been determined. Adjusting the seed can offer a certain level of variation and inspiration, yet this approach has its limitations and provides minimal control. In light of prompt engineering, this can often descend into a tedious cycle of trial and error, as users grapple with different prompt modifiers aiming to enhance their results. The objective of this method to significantly improve upon this limitation is adopting an iterative approach. By presenting users with a series of related images, generated from subtly varying the underlying prompt embeddings, creative possibilities can be expanded providing structured and iterative inspirations to users. In the following this approach is precisely illustrated and a thorough evaluation is detailed. Afterwards, the limitations are addressed with a concluding analysis of the method's generalization ability.

5.1 Methodological Approach

The framework starts by initializing the current prompt embedding, denoted as $C = \psi(P)$, where initially P corresponds to the prompt provided by the user and is updated by interpolation throughout the process.

Generation of Candidate Prompt Embeddings

Starting with iterative image suggestion through prompt embedding modification, the methodology unfolds as follows. Offering a range of options for the users, candidate prompt embeddings $\hat{C}_i$ are generated according to the equation:

$$\hat{C}_i = \text{SLERP}(C, \tilde{C}_i, \gamma_i)$$

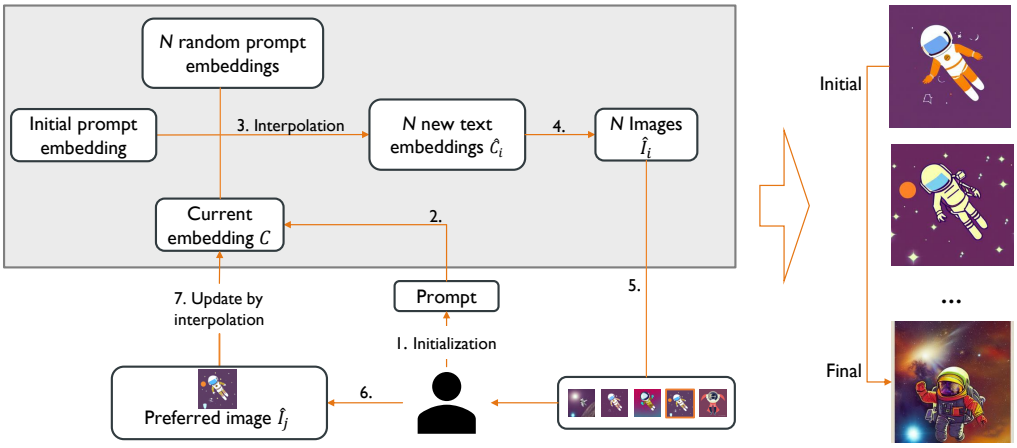


Figure 5.1: Workflow of the User Interaction method.

Accordingly, $\tilde{C}_i$ are embeddings of random prompts $\tilde{P}_i$, constructed primarily by combining random alphanumeric characters. Among a large pool of such candidates, a subset is determined based on maximizing the pairwise cosine distance, ensuring diversity in the candidate prompt embeddings.

The interpolation parameter γ_i is methodically selected to maintain a constant and equal cosine distance between C and $\hat{C}_i$, aiming for perceived uniformity in the choices presented to the user. Through an empirical verification, utilizing the dataset described in Section 4.3, the standardization of CLIP embeddings could be determined. As a result, the cosine distance remains equivalent to the euclidean.

Derivation of the Interpolation Parameter γ_i

The derivation for γ_i starts by considering a completely random prompt embedding C_i^*. Initially, LERP was used throughout the procudure. Since $\hat{C}_i$ is an interpolated version of C and C_i^*, the relationship can be represented as:

$$\hat{C}_i = (1 - \gamma_i)C + \gamma_i C_i^* = C + \gamma_i(C_i^* - C)$$

Therefore, by introducing a fixed, predefined distance d_0 the Euclidean distance between $\hat{C}_i$ and C becomes:

$$\begin{aligned} d_0 &= ||\hat{C}_i - C||_2 \\ &= ||\gamma_i \cdot (C_i^* - C)||_2 \\ &= \gamma_i \cdot d(C_i^*, C) \end{aligned}$$

This implies that γ_i is the ratio of the distance between C_i^* and C to the fixed predefined distance d_0:

$$\gamma_i = \frac{d(C_i^*, C)}{d_0}$$

It is important to note that despite this derivation, SLERP is the interpolation method of choice. The individual interpolation parameter is not necessarily tuned to SLERP. Yet, since the random embeddings were generated to maximize distances between them, the diversity of C_i^* is inherently present. Testing suggested that the initial derivation of the parameter d_0 from LERP did not negatively affect the perceived distances of the resulting images from the current prompt embedding. Thus, the calculation for d_0 remains unmodified.

Refinement of Candidate Embeddings

In a crucial refinement step, each $\hat{C}_i$ is further modified towards the original prompt P. Similarly to the random prompts $\tilde{P}_i$ creation, this is achieved by applying other randomly chosen modifiers to P from the same predetermined list of effective modifiers. This step ensures the preservation of aesthetic quality and prevents undue divergence from the original prompt's intent.

Update by User Selection

The system then generates images $\hat{I}_i$ from the refined embeddings:

$$\hat{I}_i = \mathrm{LDM}(\hat{C}_i)$$

Users are invited to select an image, denoted by the choice j, and assign an interpolation parameter $\alpha \in [0, 1]$. This α is used to update C for the next iteration as follows:

$$C = \mathrm{SLERP}(C, \hat{C}_j, \alpha)$$

Finally, the new current image I is displayed, computed as $\mathrm{LDM}(C)$. Throughout the iterative process the seed is kept constant to heighten the predictability of the results. By executing this method, the system facilitates a more directed optimization of user satisfaction, offering a balanced combination of creative freedom and targeted results.

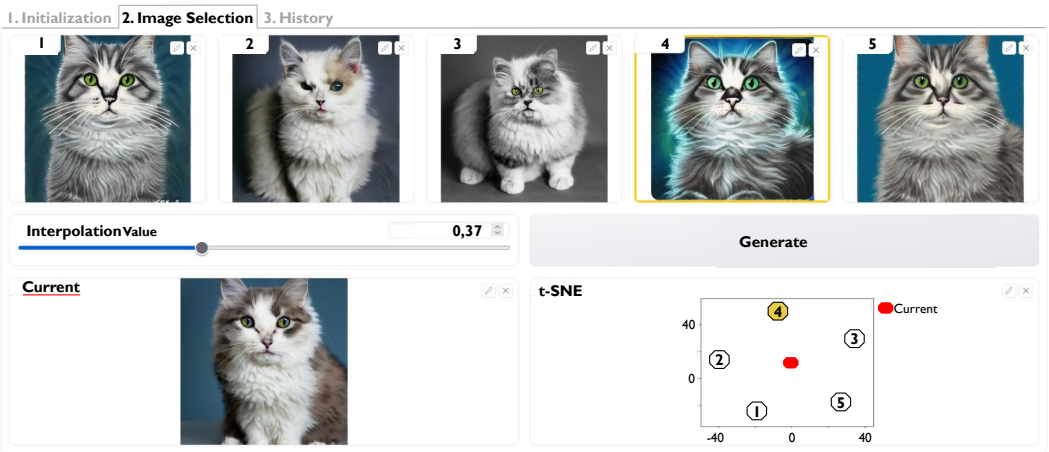


Figure 5.2: User interface of the user interaction system. The interface prominently displays the current image selection at the lower left. The options available for user selection are arrayed across the top. On the bottom right, a t-SNE [van der Maaten and Hinton, 2008] plot visualizes the dimensionality reduction, placing the current embedding at the center with the five selectable options distributed around it.

To provide a comprehensive overview of the entire process, Figure 5.1 illustrates the workflow for the user interaction method and Figure 5.2 showcases the actual user interface utilized. The process initiates with the user entering an initial prompt, which the system uses to calculate the current embedding. The resultant image, displayed at the bottom left in Figure 5.2, stems from this embedding. This initial image and its corresponding prompt embedding are iteratively refined to yield a more satisfying visual result. To facilitate this, $N = 5$ random prompt embeddings, along with the original one, are augmented by incorporating specific, predetermined prompt modifiers. These modified embeddings are then interpolated to generate $N = 5$ candidate images for user selection, displayed at the top of Figure 5.2. Note that the prompt augmentation step is omitted from the figures for simplicity. The user then chooses their preferred image (for example, image 4) and specifies an interpolation value in the interface, determining the degree of influence the selected image will have on the subsequent image outcome. Based on this input, a new current embedding is computed, from which the next iteration's image and selection options are derived, advancing the user-guided image evolution.

5.2 Dataset

For assessing the user study, a selection of initial prompts from DiffusionDB and Lexica was employed. Most of the incorporated prompt modifiers in these were eliminated. A complete list of the prompts used during the user study can be found in Appendix C.2.

5.3 Implementation

This section details the specifics of realizing the User Interaction method illustrated in Section 5.1. Although the methodological foundation offers a theoretical roadmap, certain adjustments were demanded.

In the User Interaction method, interpolation is applied solely to the conditional embedding. No modifications were carried out on the unconditional part.

As described in the methodological approach (Section 5.1), SLERP is utilized. The sum of the coefficients derived for linear interpolation in this method does not add up to 1. Consequently, as discussed in Section 2.2.6, special attention must be paid to ensure that the start token of the conditional embedding remains unaltered and is excluded from the interpolation process, accordingly.

5.4 Evaluation

This section provides an in-depth analysis of the proposed user interaction method, comparing it to the conventional prompt engineering technique. By incorporating feedback from real participants, this section draws distinctions between the advantages, challenges and areas of enhancement for the proposed approach. Topics discussed include the user segment most likely to benefit from the user interaction approach, the scope of its generalizability provided by the user interaction implementation and notable limitations within the experimental setup.

5.4.1 Comparative Analysis

A user study experiment was conducted with eight participants. The corresponding questionnaire specified within this study is presented in Appendix C.1. They were instructed to generate images from given prompts using the proposed user interaction technique, adjusting these images based on their preferences. As a means of comparison against traditional prompt engineering, an equivalent system with a similar interface was utilized as a reference point. Each participant had 20 iterations to craft their ideal image using both methods. Figure 5.3 presents a selected subset of image outcomes, generated during the user study.

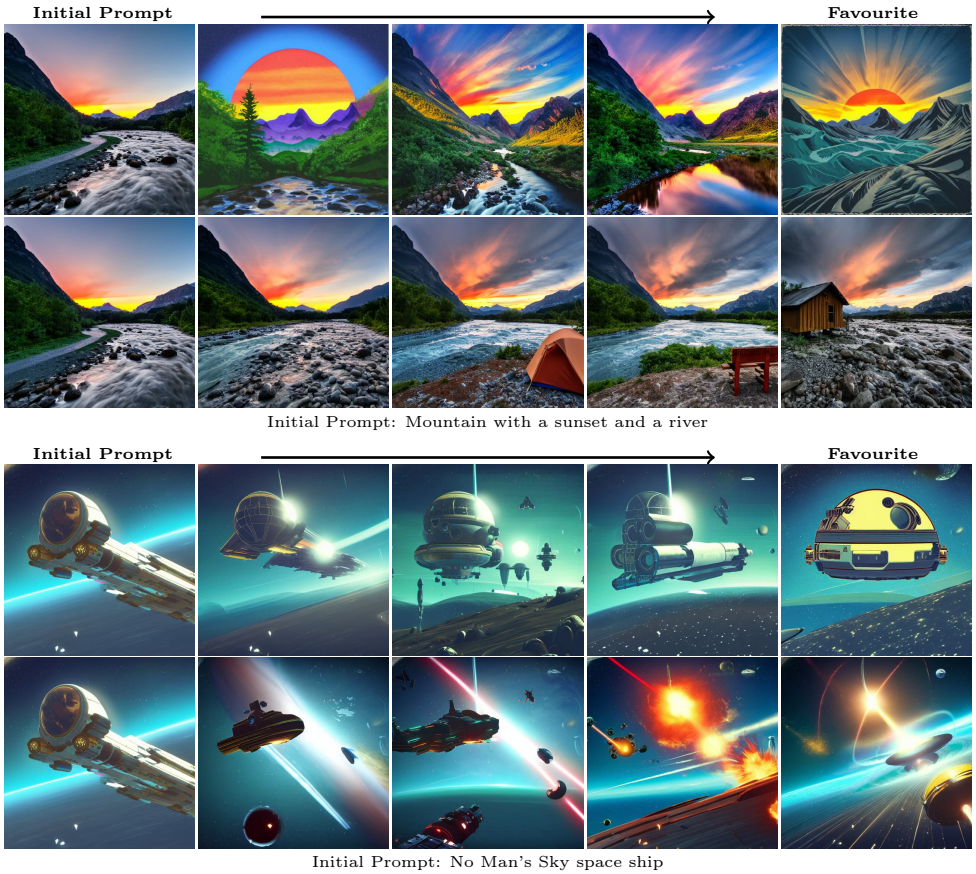


Figure 5.3: Selected illustrations from the user study: A comparison between the user interaction method (first line for each prompt) and prompt engineering (second line for each prompt).

To maintain objectivity in the assessment and to consider the possible influence of sequence, the starting method was alternated among participants. Therefore, half of them began with the interactive technique, while the others started with prompt engineering. During the experiment, participants shared the strategies they adopted. In addition to the images, the associated embeddings were stored for a subsequent evaluation of the proposed method's generalizability, which is further detailed in section 5.4.3. Concluding the activity, participants were requested to comparatively rank the images derived from both strategies.

The user study results highlight the potential advantages of the user interaction method over traditional prompt engineering, particularly for those who had limited experience with prompt engineering. A majority of participants found the user interaction method to be intuitive and favorable.

Moreover, the interactive technique offers an exploratory avenue for users to generate images, particularly when they do not have a fixed visualization in mind. This approach facilitates a brainstorming phase in a creative process, where users can be inspired by the presented options and then iteratively refine their ideas. Traditional prompt engineering, in contrast, may require users to have a clearer vision right from the start, which can be limiting in some contexts.

While there is room for improvement in terms of runtime efficiency, the extended wait was generally considered an acceptable trade-off for the benefits offered by the user interaction method. The approach was described as less cumbersome and more enjoyable compared to prompt engineering. Notably, six out of eight participants favored the images derived from the user interaction procedure over their results obtained by prompt engineering.

The user interface for the interactive method utilized an intuitive slider for setting the interpolation parameter. A value of 0, when the slider was fully set to the out most left, implied dependence solely on the optimized prompt embedding. In contrast, a value of 1, when moved fully to the right, ensured total reliance on the chosen image and its linked prompt embedding.

Figure 5.4 visualizes participants' preferences also using such a slider. Similarly, a slider value of 0 denoted a complete preference for the interactive method over the top image from prompt engineering, accordingly a value of 1 refers to absolute affinity for the prompt engineering result. Each blue dot corresponds to individual participants' ratings, while the red dot represents

the average rating of the group, offering a holistic view of the users' preferences. The group's average rating of 0.37, has a tendency towards the user interaction method.

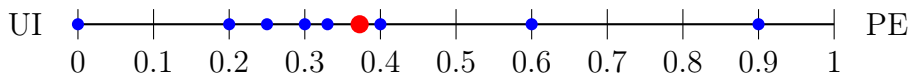


Figure 5.4: User preferences between methods: Blue dots for individual ratings, with the left end (UI) favoring user interaction and the right (PE) prompt engineering. The red dot marks the average rating.

5.4.2 Limitations

Due to the temporal constraints, the experimental design presents certain restrictions. One notable limitation of the study's design lies in the inability to initiate the process with differing prompts for each of the two methods. Consequently, a direct comparison between the techniques is somewhat restricted. This limited comparability arises because the sequence in which a participant begins with a method can introduce bias towards the succeeding one. For example, users beginning with prompt engineering already have a mental picture while engaging in the user interaction method, which operates mainly through inspiration. This process makes the user reliant on receiving suggestions that align with their envisioned target, which may not necessarily match the image they initially had in mind. In contrast, users who commenced with the user interaction method found it easier to articulate a prompt, afterwards, as the suggestions already offered them a directional framework for their creativity.

Other constraints also emerged. Participants expressed a desire for a backtrack feature, which might be a feasible addition in forthcoming iterations. Occasionally, the updated image was less preferable than its predecessor, urging participants to reconsider their choices or make minor interpolative adjustments.

Two of eight participants also expressed the need for a feature controlling image diversity. While some prompts led to noticeably varying suggestions, others produced rather similar outputs. This might be attributed to the interpolation parameter γ_i which is not ideally tailored for SLERP. An adjustment of the parameter's computation could lead to a higher diversity for the candidate images of these two specific prompts.

Lastly, the non-optimized runtime serves as a major impediment. Image generation occurs in a sequential manner, and even parallelized generation of candidate embeddings incurs additional time. This leads to considerably extended waiting periods as compared to traditional prompt engineering. Considering that only 25% of the present GPU resources were used operationally, future iterations could employ enhanced parallelization strategies to drastically reduce runtime without compromising on performance for comparable hardware.

The limitations presented provide useful insights for future research. Addressing the issue of sequence bias can lead to evaluations with higher comparability. Implementing features like backtracking, adjusting for diversity, and improving runtime efficiency can elevate the user experience. Especially, the desire for backtracking and image diversity emphasizes the users' need for control and variety when using such tools in creative endeavors.

5.4.3 Generalization

Similar to the metric-based method discussed in Section 4.5.3, the ability to generalize using the user interaction method was also evaluated. Throughout the user study, embeddings resulting from both the user interaction and prompt engineering methods were saved.

For each of the preferred images selected from both techniques, additional image outputs were generated using 5 distinct seeds for the associated prompts. As outlined previously in Section 5.4.1, participants were providing their rating using the slider mechanism.

Figure 5.5 illustrates the ratings for the five seeds using a slider-based visualization. In contrast to Section 4.5.3, the prompt embeddings did not generalize across different seeds demonstrated in this analysis. The mean rating across the seeds hover around a value of 0.5. This might suggest that while the images generated during the initial exploration are well-received, the underlying prompt embeddings do not consistently translate to favorable images across different seeds.

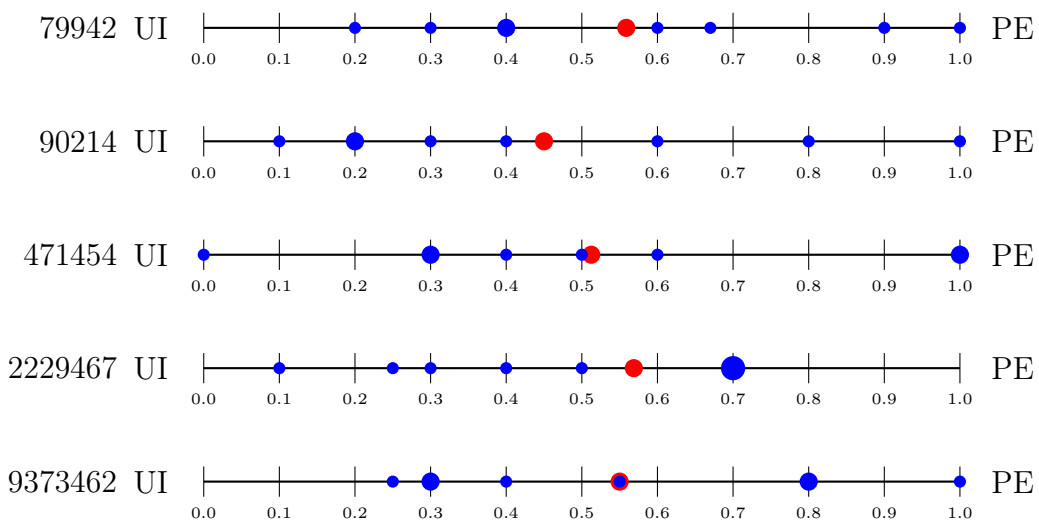


Figure 5.5: User preferences for different seeds: Blue dots represent individual ratings, with varying thickness indicating repeated values (thicker for duplicates, thickest for triplicates), and the red dots signify the mean ratings. Each slider is marked with a scale from 0.0 to 1.0. The left end (UI) favors user interaction and the right (PE) indicates prompt engineering.

Chapter 6

Seed-Invariant Optimization

The role of seeds in the prompt engineering process is both intricate and influential. While they offer a way to introduce randomness or unique styles into the generated images, they also introduce challenges. When paired with certain prompts seeds can drastically affect the outcome, leading to outputs that may deviate considerably from the user's intent. In some cases, a good prompt only produces satisfactory results with specific seeds, highlighting the prompt's inadequacy in capturing all essential details. Furthermore, minor changes to the prompt can bring about unexpected and substantial alterations to the generated image, even when the intent was to change only a specific detail.

This chapter shifts the focus towards seed-invariant prompt embeddings. By directly modifying the prompt embeddings, the goal is to eliminate the inherent vagueness and unpredictability in prompts [Hutchinson et al., 2022]. This way, a consistent and predictable output can be maintained across a range of seeds. The implementation of seed-invariant embeddings aims to reduce the random factors that seeds introduce, offering a more precise, user-friendly and adaptable approach for generating images.

Prompt engineering often involves users experimenting with various seeds to discover creative ideas. Upon encountering a particularly intriguing element, such as a unique object or style, they attempt to articulate this feature to incorporate it into the prompt, which can be quite challenging. As demonstrated in Figure 5, the seed can significantly impact the image generation when paired with specific prompts. Dependence on the seed for user satisfaction suggests that the prompt lacks comprehensive details.

The ultimate goal of this method is to derive a highly accurate description for the prompt. It is envisioned to convert this description into text form, allowing for targeted modifications at the textual level, such as altering the color of an object, while maintaining the surrounded areas. Moreover, in the future, there is an aspiration to allow for the preservation of preferred sections in the image and the ability to regenerate the neighboring regions. This way, the method aims to enable flexible and targeted editing on both smaller and larger regions within the image. Significantly, with this approach desired modifications are intended to be executed instantly, bypassing the often tedious process of iterative prompt refinement to achieve the desired image outcome.

6.1 Methodological Approach

Given a target image I derived from a prompt P and a primary latent z_T, the aim is to identify an optimal prompt embedding C^* that ensures:

$$\mathrm{LDM}(\psi(P), z_T) = \mathrm{LDM}(C^*, \hat{z}_T) = I$$

for any feasible initial latents $\hat{z}_T$. This process is detailed in Algorithm 1 and Figure 6.1a.

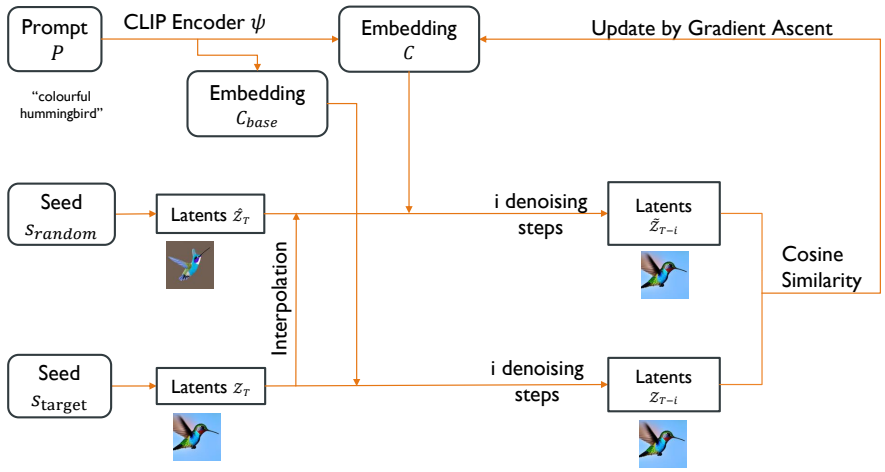
The algorithm operates by refining the prompt embedding C through gradient descent, aiming to ensure it closely aligns with the target image across different seeds. This is achieved via the interpolation parameter α, which increasingly incorporates varying seeds. Here, the loss computation is performed in the latent space.

Furthermore, ϵ_θ represents the denoising U-Net. Initially, the loss is determined based on latents after just a single denoising operation. This involves considering both the original seed and seeds that are incrementally diverging from the original. As the algorithm evolves, the latents are subjected to further denoising steps. Throughout, α ensures the consistent introduction of seeds that progressively deviate from the primary seed.

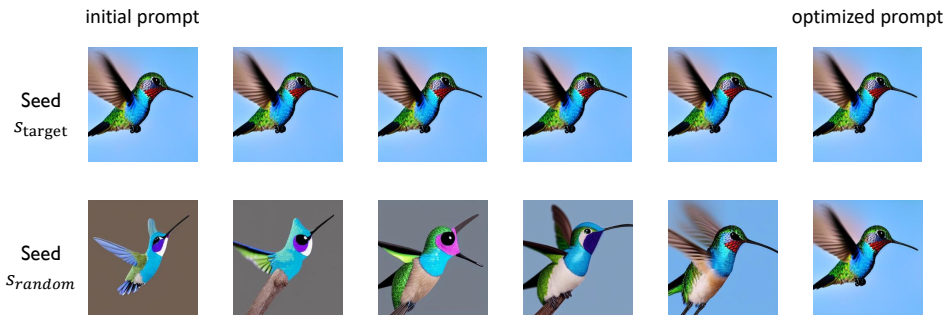
In the end, the algorithm aims to compute an optimized prompt embedding by applying gradient descent that reliably corresponds to the target image, regardless of the chosen seed.

Figure 6.1b illustrates the intended outcomes following the algorithm's application. Each row contains images crafted from the progressively refined embedding C. The top row showcases images generated from a seed aligned

perfectly with the designated prompt, resulting in no noticeable changes during the prompt optimization. This is because the prompt consistently becomes a more accurate representation of the target image. Conversely, the second row uses a seed for validation. As the prompt is optimized, this image converges towards the top row's ideal image, ultimately yielding identical results regardless of the initial seed used.



(a) Illustration of the seed-invariant embedding generation approach. Images created merely for validation purposes based on the latents above. Due to interpolation, the resultant image outcomes based on $\tilde{z}_{t-1}$ and z_{t-1} turn out nearly identical.



(b) Representation of the resultant images, obtained from the seed-invariant embedding generation approach. During prompt optimization, the outcome generated from random seeds converges towards the target image.

Figure 6.1: Overview of the seed-invariant prompt embeddings generation. The displayed images are obtained by interpolation and do not equate to true results.

Despite the incremental incorporation of random seeds via SLERP, it is possible to limit the computation of $\hat{\epsilon}_\theta(\hat{z}_t, t, C)$ to the randomly sampled initial latents $\hat{z}_T$ merely without using interpolation. This will be conducted in the following ablation study in Section 6.4.

Algorithm 1 Seed-Invariant Prompt Embeddings

$C \leftarrow \psi(P)$
$C_{\text{base}} \leftarrow \psi(P)$
for $i \leftarrow 1, \ldots, N$ **do**
  for $j \leftarrow 1, \ldots, M$ **do**
    $\alpha \leftarrow \frac{j}{n}$
    Sample $\hat{z}_T$ as a batch of random initial latents
    $\tilde{z}_T \leftarrow \text{SLERP}(z_T, \hat{z}_T, \alpha)$
    for $t \leftarrow T, \ldots, T - i$ **do**
      $z_{t-1} \leftarrow \hat{\epsilon}_\theta(z_t, t, C_{\text{base}})$
      $\tilde{z}_{t-1} \leftarrow \hat{\epsilon}_\theta(\tilde{z}_t, t, C)$
    end for
    $L \leftarrow -\frac{z_{t-1} \cdot \tilde{z}_{t-1}}{\|z_{t-1}\|_2 \cdot \|\tilde{z}_{t-1}\|_2}$
    $C \leftarrow C - \eta \nabla_C L$
  end for
end for
return C

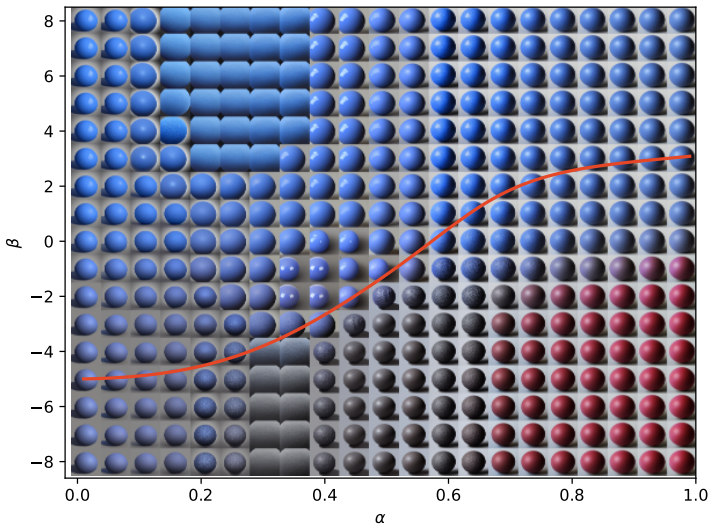


Figure 6.2: Exploring the prompt embedding space with a progressively adjusted seed. The parameter α represents the SLERP interpolation between Seed 683395 (left) and Seed 417016 (right). On the vertical axis, the prompt embedding space is depicted, where sigmoid(β) signifies the SLERP interpolation between `Single Color Ball` (bottom) and `Blue Single Color Ball` (top). The plotted orange curve captures the evolution of β at each increment of α.

To further illustrate the method under consideration, an overly simplified example is employed. The images displayed in Figure 6.3 are generated from 5 different random seeds $s_1, \ldots, s_5$. Using these seeds the aim is to compute a prompt embedding that generates images being visually as close as possible to the first of Seed s_1, even when prompted with any of the further seeds $s_2, \ldots, s_5$.

The algorithm is simplified by confining the parameter space to one dimension. Hence, the gradient is not computed aiming to update the embedding C but the interpolation parameter β. The interpolation process utilizes SLERP between the prompt embeddings of `Single Color Ball` and `Blue Single Color Ball`. The latter equates to a more precise description for the target image, containing an additional information. According to the underlying hypothesis, as the method advances, it is expected that the embedding C will increasingly incorporate details of the target image, leading the resultant prompt to align closer with the more specific `Blue Single Color Ball` prompt.

Before performing SLERP, the sigmoid function is applied to β. This restricts the resulting embedding space to those embeddings, which lie exactly between the two involved in this interpolation, ensuring the traceability of the result. This setup is depicted in Figure 6.2.

If the method functions as hypothesized, C will increasingly align with the more detailed prompt `Blue Single Color Ball`. This would be evidenced by an upward trend of the curve towards a positive β as α rises, implying an ascending curve in Figure 6.2. Experimental findings confirm this expectation.

Figure 6.3: Images generated using the prompt `Single Color Ball` across five distinct selected seeds.

6.2 Implementation

This section elaborates on the practical aspects of implementing the method discussed in Section 6.1. While this technique serves as a theoretical foundation, some modifications were required during implementation for various reasons, including computational limitations and optimization for performance.

Delving into the specifics of Algorithm 1, it undergoes a total of 250 iterations, with this number derived from $M = 50$ updates to the interpolation parameter α for a maximum of $N = 5$ denoising steps. To clarify, the parameter N was set to 5, indicating the procedure calculates the cosine similarity for the latents incrementally, starting from the first denoising step and culminating at the fifth, each step being repeated 50 times. n equates to 200, implying an increase of α by 0.05.

The value for M was determined based on the similarity computation solely for the once-denoised latents, $\tilde{z}_{T-1} = \hat{\epsilon}_\theta(\tilde{z}_T, T, C)$ and $z_{T-1} = \hat{\epsilon}_\theta(z_T, T, C_{\text{base}})$. This ensured that the resulting embeddings from the chosen three prompts were robust against overfitting. Initially, $\tilde{z}_{T-1}$ and z_{T-1} were the only latents compared, which led to overfitting after about 50 iterations, resulting in increasingly deviating images for employed validations seeds.

In addition, Algorithm 1 illustrates a simplified version of this methodology, as it suggests drawing only one random seed (line 6). Later it became evident that leveraging batches of seeds significantly enhances the outcomes. Due to existing hardware constraints, the gradient for a batch could encompass at most three seeds. Similar to the specifications mentioned in Section 4.4, the gradient is calculated for only one denoising step of the latents, particularly for the last computed $(T-i)$, as defined by the outer for loop of the algorithm (line 3).

As highlighted in Section 2.2.6, it is imperative to maintain the embedded representation for the start token unaltered, rather than updating it via gradient descent. Furthermore, the parameter space for gradient descent was intentionally limited. Solely the numerical representation of the last token of the conditional embedding was taken into account. Given that this token intrinsically carries information regarding preceding tokens, as expounded in Section 2.2.6, it was repetitively appended 76 times behind the start token within this procedure. Consequently, the resulting embedding was utilized as the conditioning element during the denoising phase of the latents.

6.3 Evaluation

To assess the methodology for generating seed-independent embeddings, the three prompts and five seeds showcased in Figure 6.4 are referenced. For each of the three prompts, the far-left image, produced using the initial prompt embedding and seed s_1, serves as a reference. This reference image should

also be generated for seeds $s_2, \ldots, s_5$ using the optimal prompt embedding. The outcomes are depicted in Figure 6.5. The far-left image corresponds to the target and was again generated using the original prompt and seed s_1 for comparability, whereas seeds $s_2, \ldots, s_5$ were all produced with the optimized prompt embedding. Analyzing the results, it is evident that images within a row resemble each other more than in Figure 6.4, which were generated using the original prompt. However, the images created with the validation seeds and the optimal prompt are not identical to the target.

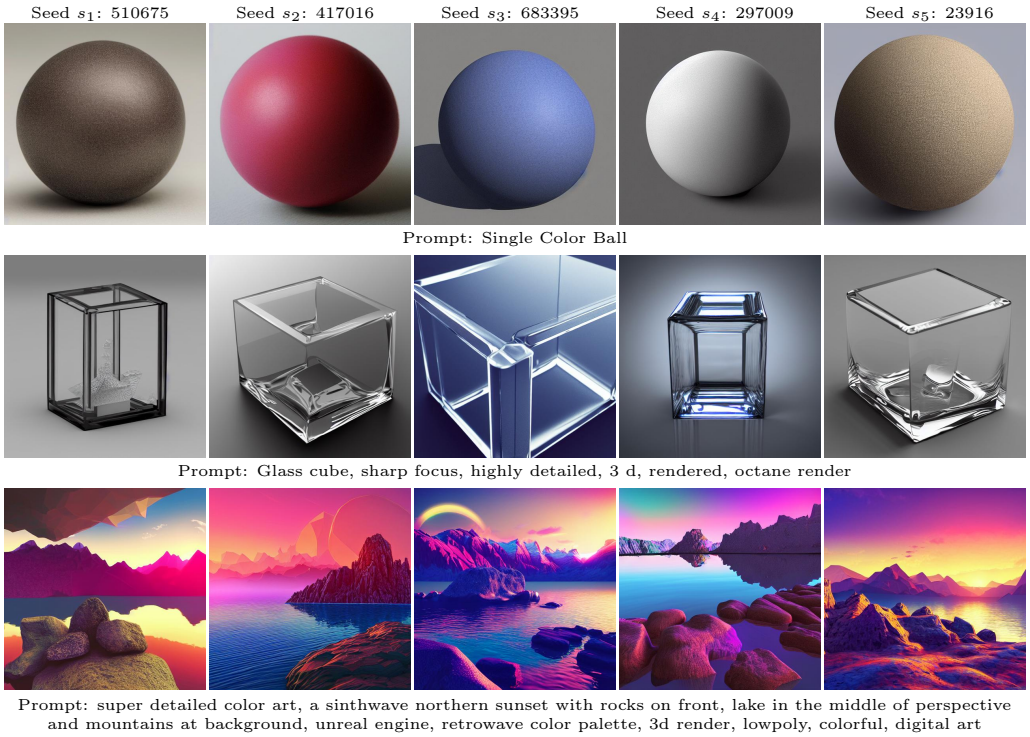


Figure 6.4: Images generated by using the unmodified initial prompts below based on the seeds $s_1 - s_5$. s_1 (most left) represents the target seed, yielding the image, towards which gradient descent is performed. $s_2 - s_5$ correspond to validation seeds.

The results of the optimized embedding for the prompt `Single Color Ball` have significantly approached the reference in terms of color. Yet, unlike the reference, the depicted balls are not uniformly colored, and the background, while similar, is not identical. Nevertheless, there is a clear resemblance to the reference. A similar trend is observed for the final prompt. The middle images represent the results of the optimized embedding for the prompt `Glass cube, sharp focus, highly detailed, 3D, rendered, octane render`. Here, the deviations from the reference are more pronounced compared to the other prompts, especially in terms of color. Still, all four cubes, based on the op-

timized embedding and validation seeds, have gained a frame and internal objects, akin to the corresponding reference.

In order to further refine these outcomes, it might be necessary to increase the number of iterations. Additionally, more criteria can be incorporated for determining the initialization parameters of the algorithm, beyond just comparing the originating latents $\tilde{z}_{T-1}$ and z_{T-1}. Instead of increasing i and thus the denoising steps after a fixed number of M iterations, an option would be to dynamically check, if the resulting prompt is overfitting and add a further denoising step based on this. Consequently, this process would be aligned much more precisely to the optimized prompt embedding. Further, the introduction of random seeds might not be conducted with the appropriate velocity. Raising the parameter n to a greater value would cause the increase of randomness of the initial latents to progress with reduced speed. Lastly, increasing the parameter space could increase the similarity to the target. For this purpose the conditional or even the entire embedding can be considered during this procedure.

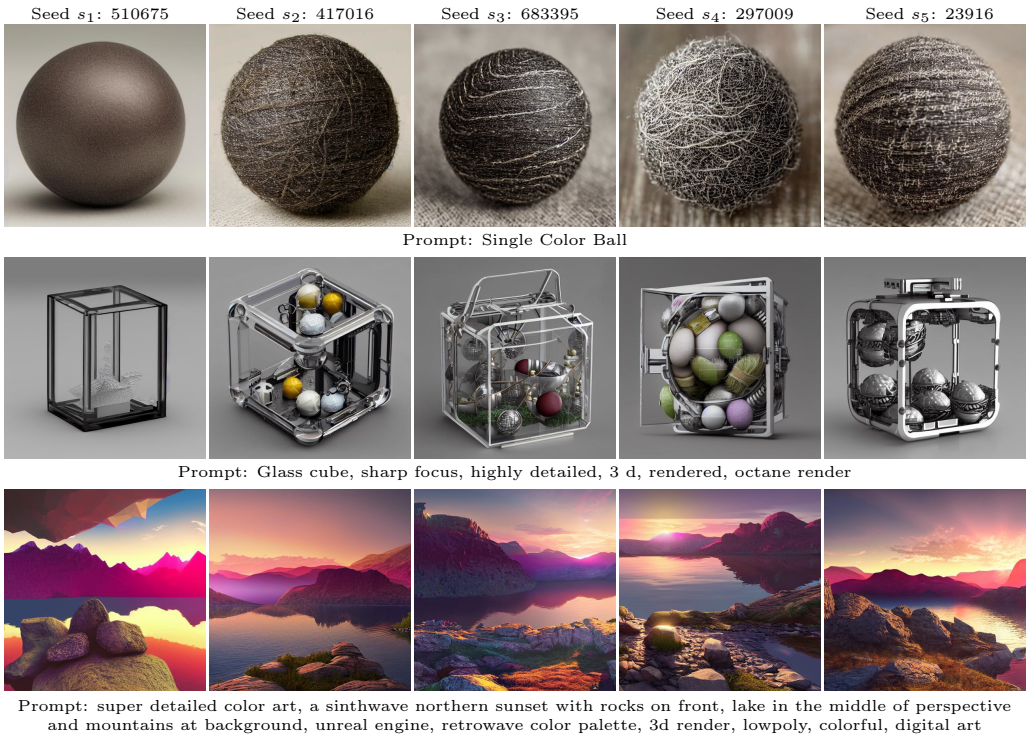


Figure 6.5: Images (except the most left) generated using the optimized prompt with validation seeds $s_2, \ldots, 2_5$. The first image is generated with the initial prompt embedding and the target seed s_1.

6.4 Ablation Study

To investigate whether a simplification of Algorithm 1 could yield similar or even superior results, this ablation study was conducted. In particular, it pertains to the simplification referenced in Section 6.1. Rather than incrementally enhancing the randomness of the seeds via interpolation, the initial latents are now denoised based solely on entirely random seeds. This described simplification is illustrated in Algorithm 2 in Appendix D.

The validation setup is identical to the one described in Section 6.3. The seeds $s_2, \ldots, s_5$ from Figure 6.4 are once again employed for validation, aiming to determine if the images produced by the updated prompt embeddings align with the designated target image.

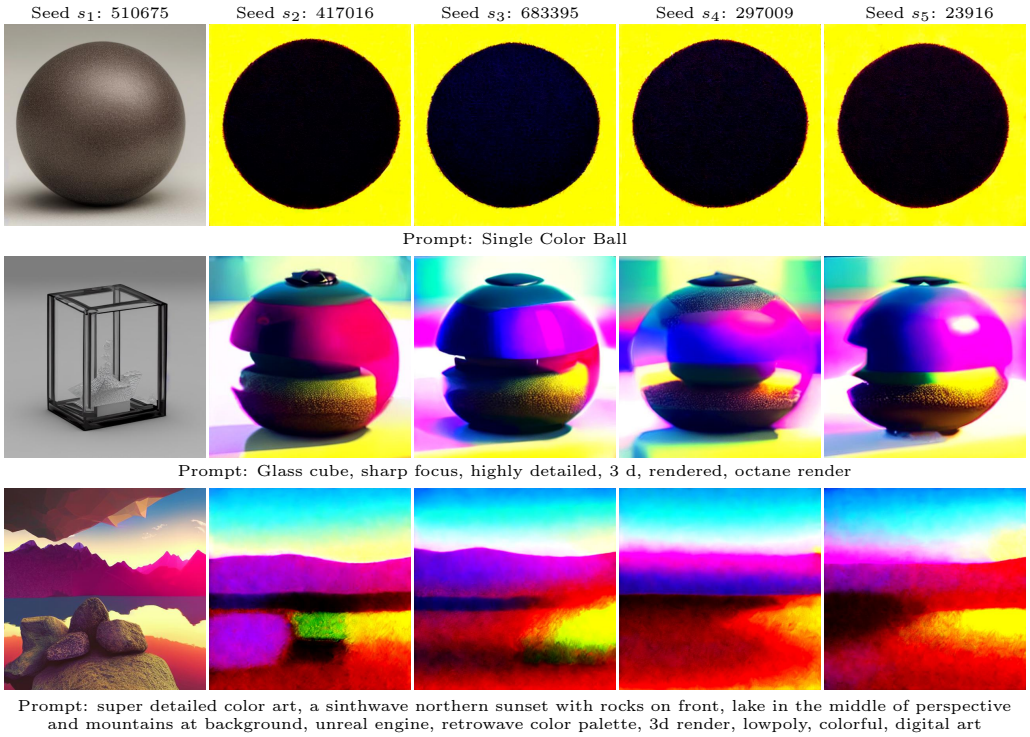


Figure 6.6: Images (except the most left) generated using the optimized prompt with validation seeds $s_2, \ldots, s_5$ utilizing the simplified approach. The first image is generated with the initial prompt embedding and the target seed s_1.

The results of this ablation study are visualized in Figure 6.6. Notably, while the images demonstrate consistency across different seeds, they diverge substantially from their respective reference images, signaling a potential over-optimization. This substantial deviation highlights the critical role that the

stepwise integration of seed randomness via interpolation plays, as described in the original Algorithm 1. Such interpolation seems to be instrumental in preserving the key features of the reference image throughout the denoising process. Concludingly, removing the interpolation steps does not present a straightforward solution. Enhancing this technique demands a more nuanced approach.

6.5 Connection with Textual Inversion

The inversion methodology introduced by Gal et al. [2022] was outlined in Section 3.2. This concept, namely Textual Inversion, closely aligns with the technique presented in this chapter.

In the approach of Textual Inversion, utilizing the forward chain of a DM, images are systematically distorted until only Gaussian noise remains (Section 2.2.1). In contrast, the reverse chain's responsibility coincides of reconstructing these images, a process detailed in Section 2.2.2. Throughout this phase, the word embedding, represented as v_*, is progressively learned. Notably, this entire procedure ensures that the weights of both the denoising model and the CLIP text encoder remain static, devoid of any adjustments. To capture a new word representation v_*, a dataset comprising 3 to 5 images displaying the same concept is essential. These images correspondingly align with their respective pseudo word, symbolized as S_*. With this pseudo word at disposal, textual prompts can be crafted, for example: `A photo of` S_*. Gal et al. [2022] showcase a variety of image renditions corresponding to prompts formulated in this style, as illustrated in Figure 6.7.

Figure 6.7: Conduction of Textual Inversion based on the displayed input samples. The results correspond to the image variations generated for the provided prompt. Created by Gal et al. [2022].

Drawing parallels, just as the seed-invariant embeddings rely on gradient-based calculations for embeddings, the Textual Inversion methodology also harnesses similar gradient-based embedding computation. Additionally, both methodologies optimize only a single vector of an embedding. However, rather than employing the resulting vector to derive the conditional part of the embeddings, the technique by Gal et al. [2022] maps this vector to a specific entity, such as an object.

Furthermore, the gradient in the seed-invariant optimization methodology is determined solely for a single denoising step, whereas this restriction is not stated anywhere for Textual Inversion. Therefore, the reasonable assumption can be applied that the gradient of each denoising step is utilized to update the embedding. In order to guarantee this, a close inspection of the code must be conducted.

The primary objectives of these two methods are distinct. The seed-invariant optimization seeks to stabilize entire images or regions within them, while Textual Inversion aims to internalize a concept visualized by a set of images and express this concept verbally within a prompt. The resultant prompt might still harbor underspecifications. Therefore, leveraging Textual Inversion, multiple image variations for a single prompt can be generated (Figure 6.7). This feature contrasts with the goals of seed-invariant text optimization, which aims to generate identical images regardless of the seed.

Moreover, the seed-invariant optimization method does not align directly with the concept of inversion as no images are obscured and subsequently reconstructed. The emphasis is solely on the image's denoising process. Seed-invariant optimization may result in the convergence of two fundamentally different images, whereas the approach by Gal et al. [2022] focuses on predicting the noise residual and thus, learning to gradually add details to the noise within the forward procedure.

In summary, despite the similarities, which lie in the gradient-based optimization of an embedding vector, both methods exhibit significant differences in their approaches. Nevertheless, closely examining the code behind Textual Inversion could prove instructive in terms of efficiency for gradient-based optimization techniques. It may offer valuable insights not only for seed-invariant embedding generation but also for metric-based optimization, as a further gradient-based approach implemented in the scope of this thesis.

Chapter 7

Conclusion

In the field of generative text-to-image models, significant advances could be observed in the recent years. However, effectively influencing this image generation process remains a challenge due to the practice of prompt engineering. This involves iteratively refining prompts to get the desired output, which often relies on a time-consuming trial-and-error approach. Users struggle with the unpredictability of the models and have difficulty obtaining exact results, especially when adjusting for specific details and accounting for random seed variations.

Addressing these challenges associated with prompt engineering in the scope of this thesis, three methods are established for automated manipulation of prompt embeddings.

Metric Based Optimization focuses on automatically refining the prompt embeddings towards a specific metric by applying gradient optimization. The algorithm iteratively updates the embeddings until the generated image closely matches the desired outcome, avoiding the trial-and-error process of prompt engineering. Proving the ability of generalization the pronouncement of this metric could be preserved for seeds that were not included during the procedure. Unfortunately, over-optimization occurred for certain prompts resulting in images deviating from the prompt's description. This condition occurred to an individual extent. A criterion for termination could not be assigned and remains under obligation for future research. The cosine similarity did not represent an adequate indicator for the deviation from the textual description. A technique exploring contextual discrepancy rather than visual might be beneficial. For instance, CLIP could be applied to verify if the image still reflects the prompt.

Iterative User Interaction aims to support users without a visual image in mind. By offering variations closely related to the user prompt, the system refines the initial prompt embedding based on interactions with the user. Most participants in the user study considered this technique less demanding and more enjoyable compared to adjusting textual descriptions. In addition, the resulting image was mainly preferred over the outcome of prompt engineering. However, this mechanism is not offering a sufficient level of control. The possibility to enhance the diversity of the proposals as well as a backtracking feature were frequently requested during the user study. With regard to the feedback expressed, appropriate improvements can be conducted accordingly in the future.

Seed-Invariant Optimization confronts the underspecification of well designed prompts resulting in entirely different image outcomes. By means of this gradient-based algorithm, only an approximation to the objective image was obtained. At this point, the resultant images still differ evidently for various seeds. Nevertheless, prominent characteristics of the reference were adopted. As a potential answer to this challenge, further parameterizations of the procedure could be explored, omitted due to temporal constraints. Thus, the iteration amount can be further extended. Secondly, the gradient is determined iteratively up to a maximum of the fifth denoising step. The number of these steps may be incremented in addition. A dynamic solution is conceivable. Lastly, the parameter space might be increased by considering up to the entirety of the embedding

Following the realization and critical assessment of the introduced image guidance techniques, their applicability has been convincingly substantiated, while acknowledging existing limitations. In conclusion, this research contributes to enhancing user experience in interacting with text-to-image models by granting users greater freedom and flexibility in their interactions with these generative models without the challenges connected with prompt engineering.

Chapter 8

Outlook

The research presented in this thesis opens several avenues for future work, aimed at refining and expanding the existing methodologies. One of the immediate areas of focus is enhancing the efficiency of the optimization process. Specifically, future work could aim to reduce the number of iterations required for embedding optimization, whether for metric-based or seed-invariant image generation.

In the area of seed-invariant embeddings, there lies the potential for substantial advancements. The ultimate aim would be to evolve the method to a point, where it can generate embeddings that are genuinely independent of the initial seed. Likewise, a refinement of the user interaction methods would enhance user control and satisfaction when utilizing this approach.

Metric-optimized embeddings offer another promising direction. Future work could leverage these for the possibility of deriving and sharing continuous prompt modifiers in a manner similar to conventional prompt modifiers. By utilizing interpolation techniques, further modifiers can be established efficiently based on the initially derived.

An integrative approach that combines multiple techniques offers another layer of potential. For instance, a seed-invariant embedding can be derived. Once this optimized embedding is in place, it could be transformed into a textual prompt. When combined with established methods as Prompt-to-Prompt by Hertz et al. [2023], an opportunity to develop a comprehensive and precise image editing tool might be enabled. Additionally, the derived continuous prompt modifiers can be integrated as an additional feature. A sufficient benchmark for validation could be SVDiff by Han et al. [2023].

Further user studies centered on the user interaction method would provide valuable insights into optimizing existing methods. Such studies could reveal new perspectives and offer an inspiration for the development of approaches that comply more profoundly with user needs and expectations.

Lastly, as focused in Section 3, the challenges associated with prompt engineering are not restricted to image generation. There is potential to transpose the methodologies conceptualized in this thesis to other modalities like audio, video and text generation, amplifying the scope of application fields.

Appendix A

Interpolation

Performing interpolation in Section 2.4.2, particularly as illustrated in Figure 2.9, two distinct prompts were leveraged to execute interpolation. Detailed below are the exact text specifications for each prompt:

1. **Prompt 1:** `A dream of an apple tree, stormy sky, high detail, concept art, matte painting, trending on artstation and deviantart, 8 k, high resolution.`
2. **Prompt 2:** `Epic landscape with a lake, golden hour, misty ground, rocky ground, distant mountains, hazy, foggy, atmospheric perspective.`

In order to perform an interpolation between initial latents, based on the seeds `61582` and `9168745`, the following prompt was employed in Section 2.4.3:

3. **Prompt 3:** `a cybernetic samoyed and beagle, concept art, detailed face and body, detailed decor, fantasy, highly detailed, cinematic lighting, digital art painting, winter, nature, running`

Appendix B

Metric Based Optimization

Based on the prompts 1 - 3, Figure 4.2 was created, displaying the resulting images during the aesthetic score optimization. In order to illustrate the evolution of the sharpness and blurriness metrics in Figure 4.3, Prompt 4 was chosen.

1. **Prompt 1:** `sun rising in digital art`
2. **Prompt 2:** `Realistic spaceship rocket design.`
3. **Prompt 3:** `an armchair made from an avocado`
4. **Prompt 4:** `Dreadfort asoiaf, dreadfort castle, house Bolton, sinister, Game of Thrones, volumetric lighting, fantasy artwork, very beautiful scenery, very realistic painting effect, hd, hdr, cinematic 4k wallpaper, 8k, ultra detailed, high resolution, artstation`

Appendix C

Iterative User Interaction

Henceforth, the questionnaire as well as a list of short prompts leveraged within the user study are illustrated. This user study was performed for the conduction of a comperative analysis of prompt engineering and the user interaction method (Section 5.4.1).

C.1 Questionnaire

Name:
Time:
First approach:

Introduction for the Prompt Engineering method:

You are given the text "—" and your task is to create an image that fits this description and follows your individual preferences. You are given a tool that converts a textual description into an image, starting with the image shown here, coming from the description shown here. You have 20 attempts to generate your optimal image and will afterwards be able to choose the preferred image from all images that you generated.
Statements that were given by the user during the experiment:

-

Questions after executing the Prompt Engineering method:

1. What experience did you have with prompt engineering in the context of generative text-to-image models before (e.g., DALL-E, Midjourney, Stable Diffusion)?

-

2. When developing your image, did you have a specific target image in mind? Did this develop while iterating or was it fixed from the beginning?

-

3. Did you feel in control when trying to specify the direction in which the image should be adjusted?

-

4. Was there a strategy or finding for the method?

-

Introduction for the User Interaction method:

You are given the text "—" and your task is to create an image that fits this description and follows your individual preferences. You are given a tool that helps you improve your image step by step. Each step starts with the current image in the bottom left. You are now given five suggestions on the top. You need to choose one of them, which will then be used to adjust your current image. For the next step, the current image will be modified towards the selected image. The slider is used to specify to what extent the selected image should be part of the newly generated image, i.e., an interpolation between the bottom left and the selected image is used. A slider value of 0 means: Take only the image on the bottom left. A slider in the middle means: Mix both images. A slider value of 1 means: Take only the selected image. You have 20 attempts to generate your optimal image and will afterwards be able to choose the preferred image from all images that you generated.
Statements that were given by the user during the experiment:

-

Questions after executing the User Interaction method:

1. When developing your image, did you have a specific target image in mind? Did this develop while iterating or was it fixed from the beginning?

-

2. Did you feel in control when trying to specify the direction in which the image should be adjusted?

-

3. Was there a strategy or finding for the method?

-

Evaluation:

For each method, choose the image that follows your individual preferences best.

- For PE: -
- For UI: -

Please give a rating between these two images by adjusting this slider. Slider full to the left means: You 100% prefer the left image. Slider full to the right means: You 100% prefer the right image.
What did you like specifically better for each image?

- For PE: -
- For UI: -

Please again assign a rating for the following image pairs:

- Seed 2: -
- Seed 3: -
- Seed 4: -
- Seed 5: -
- Seed 6: -

C.2 User Study Prompts

The following prompts were utilized within the conducted user study, as described in Section 5.2.

- `3d colorful steampunk robot`
- `portrait of a lion`
- `dark night, full moon`
- `hummingbird with colorful flowers`
- `Mountain with a sunset and a river`
- `building cyberpunk night`
- `No Man's Sky space ship`
- `scary and horror old house`

Appendix D

Seed-Invariant Optimization: Simplified Algorithm

Algorithm 2 Seed-Invariant Prompt Embeddings

```
C ← ψ(P)
C_base ← ψ(P)
for i ← 1, ..., N do
    Sample ẑ_T as a batch of random initial latents
    for t ← T, ..., T − i do
        z_{t−1} ← ε̂_θ(z_t, t, C_base)
        ẑ_{t−1} ← ε̂_θ(ẑ_t, t, C)
    end for
    L ← −(z_{t−1} · ẑ_{t−1}) / (‖z_{t−1}‖_2 · ‖ẑ_{t−1}‖_2)
    C ← C − η∇_C L
end for
return C
```

Bibliography

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

Niklas Deckers, Maik Fröbe, Johannes Kiesel, Gianluca Pandolfo, Christopher Schröder, Benno Stein, and Martin Potthast. The Infinite Index: Information Retrieval on Generative Text-To-Image Models. In Jacek Gwizdka and Soo Young Rieh, editors, *ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR 2023)*, pages 172–186. ACM, March 2023. doi: 10.1145/3576840.3578327. URL `https://doi.org/10.1145/3576840.3578327`.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P. Xing, and Zhiting Hu. Rlprompt: Optimizing discrete text prompts with reinforcement learning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 3369–3391. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.222. URL `https://doi.org/10.18653/v1/2022.emnlp-main.222`.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021,*

virtual, pages 8780–8794, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

Wenkai Dong, Song Xue, Xiaoyue Duan, and Shumin Han. Prompt tuning inversion for text-driven image editing using diffusion models. *arXiv preprint arXiv:2305.04441*, 2023.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*, 2022.

Ligong Han, Yinxiao Li, Han Zhang, Peyman Milanfar, Dimitris N. Metaxas, and Feng Yang. Svdiff: Compact parameter space for diffusion fine-tuning. *CoRR*, abs/2303.11305, 2023. doi: 10.48550/arXiv.2303.11305. URL https://doi.org/10.48550/arXiv.2303.11305.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=_CDixzkzeyb.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *CoRR*, abs/2207.12598, 2022. doi: 10.48550/arXiv.2207.12598. URL https://doi.org/10.48550/arXiv.2207.12598.

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

Chin-Wei Huang, Jae Hyun Lim, and Aaron C Courville. A variational perspective on diffusion-based generative models and score matching. *Advances in Neural Information Processing Systems*, 34:22863–22876, 2021.

Ziqi Huang, Kelvin C. K. Chan, Yuming Jiang, and Ziwei Liu. Collaborative diffusion for multi-modal face generation and editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 6080–6090. IEEE, 2023. doi: 10.1109/CVPR52729.2023.00589. URL https://doi.org/10.1109/CVPR52729.2023.00589.

Ben Hutchinson, Jason Baldridge, and Vinodkumar Prabhakaran. Underspecification in scene description-to-depiction tasks. In Yulan He, Heng Ji, Yang Liu, Sujian Li, Chia-Hui Chang, Soujanya Poria, Chenghua Lin, Wray L. Buntine, Maria Liakata, Hanqi Yan, Zonghan Yan, Sebastian Ruder, Xiaojun Wan, Miguel Arana-Catania, Zhongyu Wei, Hen-Hsen Huang, Jheng-Long Wu, Min-Yuh Day, Pengfei Liu, and Ruifeng Xu, editors, *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2022 - Volume 1: Long Papers, Online Only, November 20-23, 2022*, pages 1172–1184. Association for Computational Linguistics, 2022. URL `https://aclanthology.org/2022.aacl-main.86`.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in Neural Information Processing Systems*, 34: 21696–21707, 2021.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL `http://arxiv.org/abs/1312.6114`.

Verena Elisabeth Kremer. Quaternions and slerp. In *Embots. dfki. de/doc/seminar ca/Kremer Quaternions. pdf*, 2008.

Senmao Li, Joost van de Weijer, Taihang Hu, Fahad Shahbaz Khan, Qibin Hou, Yaxing Wang, and Jian Yang. Styledifffusion: Prompt-embedding inversion for text-based editing. *arXiv preprint arXiv:2303.15649*, 2023.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9):195:1–195:35, 2023. doi: 10.1145/3560815. URL `https://doi.org/10.1145/3560815`.

Vivian Liu and Lydia B Chilton. Design guidelines for prompt engineering text-to-image generative models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–23, 2022.

Calvin Luo. Understanding diffusion models: A unified perspective. *CoRR*, abs/2208.11970, 2022. doi: 10.48550/arXiv.2208.11970. URL `https://doi.org/10.48550/arXiv.2208.11970`.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023.

Jonas Oppenlaender. Prompt engineering for text-based generative art. *arXiv preprint arXiv:2204.13988*, 2022.

Suraj Patil, Pedro Cuenca, Nathan Lambert, and Patrick von Platen. Stable diffusion with diffusers. *Hugging Face Blog*, 2022. [https://huggingface.co/blog/rlhf](https://huggingface.co/blog/stable_diffusion).

David A. Patterson, Joseph Gonzalez, Quoc V. Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R. So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *CoRR*, abs/2104.10350, 2021. URL `https://arxiv.org/abs/2104.10350`.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022a. doi: 10.48550/arXiv.2204.06125. URL `https://doi.org/10.48550/arXiv.2204.06125`.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and MarkChen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022b.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International*

Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, pages 234–241. Springer, 2015.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=BJrFC6ceg`.

Christoph Schuhmann. Laion-aesthetics, 2022a. URL `https://laion.ai/blog/laion-aesthetics/`. Accessed: 16.09.2023.

Christoph Schuhmann. Clip+mlp aesthetic score predictor, 2022b. URL `https://github.com/christophschuhmann/improved-aesthetic-predictor`. Accessed: 16.09.2023.

Aditya Sharma. Variational autoencoder in tensorflow, 2021. URL `https://learnopencv.com/variational-autoencoder-in-tensorflow/`. Accessed: 16.09.2023.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, 2020.

Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, 1985.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics.

In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2256–2265. JMLR.org, 2015. URL `http://proceedings.mlr.press/v37/sohl-dickstein15.html`.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3738–3746, 2016. URL `https://proceedings.neurips.cc/paper/2016/hash/6ae07dcb33ec3b7c814df797cbda0f87-Abstract.html`.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a. URL `https://openreview.net/forum?id=St1giarCHLP`.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021b. URL `https://openreview.net/forum?id=PxTIG12RRHS`.

stability.ai. Stable diffusion public release, 2022. URL `https://stability.ai/blog/stable-diffusion-public-release`. Accessed: 16.09.2023.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for modern deep learning research. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 13693–13696. AAAI Press, 2020. doi: 10.1609/aaai.v34i09.7123. URL `https://doi.org/10.1609/aaai.v34i09.7123`.

Yu Takagi and Shinji Nishimoto. High-resolution image reconstruction with latent diffusion models from human brain activity. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 14453–14463. IEEE, 2023. doi: 10.1109/CVPR52729.2023.01389. URL `https://doi.org/10.1109/CVPR52729.2023.01389`.

Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. Context-tuning: Learning contextualized prompts for natural language generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6340–6354, 2022.

Guy Tevet, Brian Gordon, Amir Hertz, Amit H. Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to CLIP space. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXII*, volume 13682 of *Lecture Notes in Computer Science*, pages 358–374. Springer, 2022. doi: 10.1007/978-3-031-20047-2_21. URL `https://doi.org/10.1007/978-3-031-20047-2_21`.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Dimitri von Rütte, Elisabetta Fedele, Jonathan Thomm, and Lukas Wolf. Fabric: Personalizing diffusion models with iterative feedback. *arXiv e-prints*, pages arXiv–2307, 2023.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing nlp. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

Zijie J. Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 893–911. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.51. URL `https://doi.org/10.18653/v1/2023.acl-long.51`.

Tom White. Sampling generative networks. *arXiv preprint arXiv:1609.04468*, 2016.

Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Better aligning text-to-image models with human preference. *arXiv preprint arXiv:2303.14420*, 2023.

Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *CoRR*, abs/2302.05543, 2023. doi: 10.48550/arXiv.2302.05543. URL `https://doi.org/10.48550/arXiv.2302.05543`.

Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual probing is [mask]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, 2021.