Manipulating Embeddings of Stable Diffusion Prompts

Niklas Deckers **Julia Peters** **Martin Potthast**
Leipzig University and ScaDS.AI
niklas.deckers@uni-leipzig.de, j.peters@studserv.uni-leipzig.de, martin.potthast@uni-leipzig.de

Abstract

Generative text-to-image models such as Stable Diffusion allow users to generate images based on a textual description, the prompt. Changing the prompt is still the primary means for the user to change a generated image as desired. However, changing the image by reformulating the prompt remains a difficult process of trial and error, which has led to the emergence of prompt engineering as a new field of research. We propose and analyze methods to change the embedding of a prompt directly instead of the prompt text. It allows for more fine-grained and targeted control that takes into account user intentions. Our approach treats the generative text-to-image model as a continuous function and passes gradients between the image space and the prompt embedding space. By addressing different user interaction problems, we can apply this idea in three scenarios: (1) Optimization of a metric defined in image space that could measure, for example, image style. (2) Assistance of users in creative tasks by enabling them to navigate the image space along a selection of directions of "near" prompt embeddings. (3) Changing the embedding of the prompt to include information that the user has seen in a particular seed but finds difficult to describe in the prompt. Our experiments demonstrate the feasibility of the described methods.

1 Introduction

Generative text-to-image models such as Stable Diffusion (Rombach et al. 2022) allow users to generate images based on a textual description called a prompt. If a generated image does not satisfy the user directly, adjusting the prompt is currently the primary *directed* way to change the generated images. Since users have found that certain prompts are more likely to produce satisfactory images than others, several approaches to write and refine prompts have emerged. The resulting variety of prompt design patterns and best practices is collectively referred to as prompt engineering (Hao et al. 2022; Witteveen and Andrews 2022). As shown in the upper left of Figure 1, prompt engineering is an iterative process of refining an original prompt until the resulting image is satisfactory. At each iteration, the user interprets the image generated by the model and evaluates how successfully they have modified the prompt. For the next iteration, the user tries to guess which reformulation might have the desired effect. Over time, the user learns how the model interprets a prompt, as it were, its "prompt language".

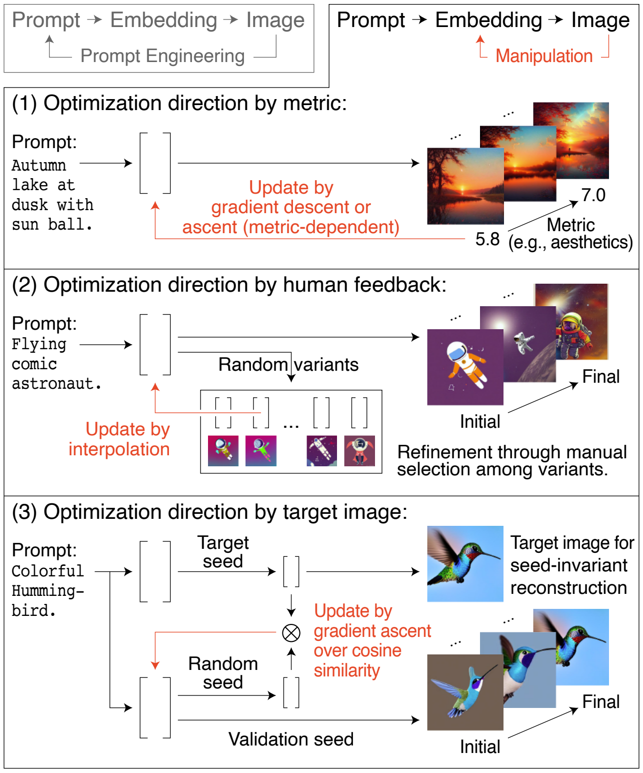
Prompt → Embedding → Image
Prompt Engineering
Prompt → Embedding → Image
Manipulation
(1) Optimization direction by metric:
Prompt:
Autumn lake at dusk with sun ball.
Update by gradient descent or ascent (metric-dependent)
7.0
5.8
Metric (e.g., aesthetics)
(2) Optimization direction by human feedback:
Prompt:
Flying comic astronaut.
Random variants
Update by interpolation
Initial
Final
Refinement through manual selection among variants.
(3) Optimization direction by target image:
Prompt:
Colorful Humming-bird.
Target seed
Target image for seed-invariant reconstruction
Update by gradient ascent over cosine similarity
Random seed
Validation seed
Initial
Final

Figure 1: Our three techniques for manipulating prompt embeddings enable a user to (1) optimize an image quality metric, (2) navigate the prompt embedding space towards nearby variants, and (3) reconstruct a preferred image by introducing seed invariance.

Prompt engineering has several shortcomings: The prompt language of a model is opaque to the user, and its interpretation by the model may differ arbitrarily from that of the user, due to the inherent ambiguity of natural language as well as potentially misleading correlations in the model's training data. In addition, a model may not consider the same parts of a prompt as important as the user, so clearly phrased prompts may have little to no impact on the generated image. Certain aspects of an image are difficult to describe, such as stylistic and aesthetic aspects and minute details. Generative models are often non-deterministic in that a new random seed is used to initialize inference for each new prompt, which can lead to unpredictable results. Overall, users report that they have a sense of direction in prompt engineering, but no control over the process (Deckers et al. 2023). We attribute this to the iterative nature of prompt engineering and a fundamental mismatch between user expectations during prompt engineering and model behavior: A generative text-to-image model does not use information about the user's previous interactions in a prompt engineering session, while the user builds a mental model from their interactions to reformulate prompts. This leads the user to presume predictable model behavior in situations where none can be expected. For inexperienced users, prompt engineering may therefore basically seem not much better than trial and error.

In this paper, we propose and analyze a new targeted approach to guide the user in the desired direction when generating their image (see Figure 1). Instead of modifying a prompt, we develop three techniques that allow the user to manipulate the prompt's embedding in a meaningful way. In typical text-to-image models, a prompt is mapped into an embedding space before the corresponding image is generated. Based on the observation that small changes to the embedding of a prompt lead to small changes in the generated image, modifying the embedding of the prompt allows the continuous modification of the information originally contained in the prompt in arbitrarily fine steps. This relieves the user of verbalizing the desired changes as well as finding a wording that the model understands, leading to a better satisfaction with each iteration.

Our three methods differ primarily in the way they determine the direction in which to modify the prompt embedding (Section 3): (1) A method based on a metric living in the image space that captures, for example, certain stylistic or aesthetic aspects. (2) A human feedback-based method in which the user selects the direction in which to modify the prompt embedding from a list of alternatives. (3) A method based on a target image generated from a prompt and a specific seed that allows the user to generate an image similar to the target image, regardless of the seed used. We evaluate these three methods in experiments and a user study (Section 4). All code and data underlying our method and experiments is provided in the supplementary material.[1,2]

2 Background and Related Work

To further motivate the idea of modifying prompt embeddings instead of prompts, this section addresses the pipeline used for generative text-to-image models like Stable Diffusion. We will also refer to related approaches that allow users to control the generation of the image without or with limited prompt engineering.

2.1 Stable Diffusion

Stable Diffusion (Rombach et al. 2022) is based on the concept of diffusion probabilistic models (Sohl-Dickstein et al. 2015), while implementing a U-Net as an autoencoder in the denoising step. This makes this architecture suitable for image generation. The denoising process, which is executed to generate an image, starts with a randomly initialized latent, making it possible to use a seed for the generation. What makes Stable Diffusion useful as a generative text-to-image model is its conditioning mechanism. It uses a cross-attention mechanism (Vaswani et al. 2017) and allows for various input modalities. For training their text-to-image model, the LAION dataset (Schuhmann et al. 2022) with text-image pairs is used. The texts (and thus also the prompts), however, are not directly used in the conditioning mechanism, but are converted into embeddings beforehand using the CLIP encoder (Radford et al. 2021). The embeddings are a numerical representation of the provided prompts.

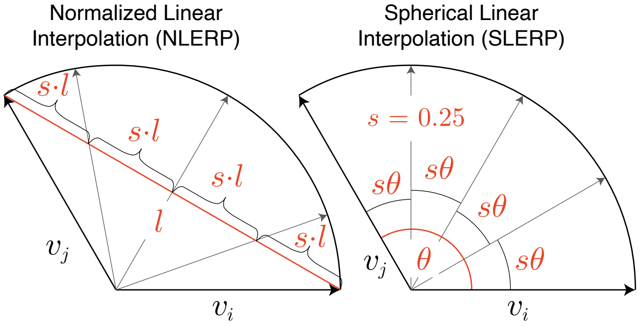

Figure 2: Comparison of two approaches for interpolating prompt embeddings.

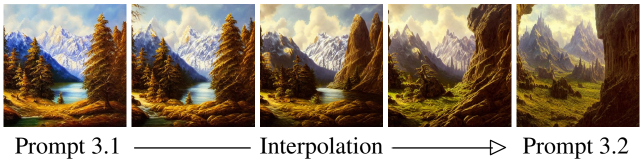

Figure 3: Selected example of an interpolation between two prompts, which can be found in the supplementary material.

Other state-of-the-art generative text-to-image models use a similar pipeline, either by employing CLIP embeddings (Ramesh et al. 2022; Nichol et al. 2022), by using different encoders like T5-XXL (Raffel et al. 2019; Saharia et al. 2022; Chang et al. 2023), or a combination of both (Balaji et al. 2022). For our experiments, we use Stable Diffusion (Rombach et al. 2022) as the generative text-to-image model due to the availability of the model weights. However, the model is mostly treated as a black box (with the exception of computing the gradients), making our approaches applicable for other models, too.

2.2 Interpolation of Prompt Embeddings

The CLIP embeddings used by Stable Diffusion to generate images encode both content and style described in the prompt. We observe that the map from the prompt embedding space to the image space that is defined by Stable Diffusion is continuous in the sense that small adjustments in the prompt embedding space lead to small changes in the image space.

[1] https://doi.org/10.5281/zenodo.8274626

[2] https://github.com/webis-de/arxiv23-prompt-embedding-manipulation

Not only is this true when considering the distance of the pixel values in the images, it also holds for the perceived difference of content and style of the generated images. It should also be noted that, more specifically, small adjustments of a well-working prompt embedding also yield an image that still has a high quality.

For larger single-step adjustments, we use an interpolation between two prompt embeddings. As a consequence of the cosine similarity being used to train CLIP, a linear interpolation (LERP) between the prompt embeddings is not suitable: If the norm of an embedding is not within a certain range, Stable Diffusion produces corrupted images. This also means that not all values that can be denoted in the same matrix format as prompt embeddings are also suitable to be used as such. Correcting the norm of linearly interpolated prompt embeddings is a feasible way to prevent this issue (see Figure 2). Using SLERP (Shoemake 1985), a spherical linear interpolation, is also possible. This method is well established in the context of interpolating prompt embeddings (Han et al. 2023). Figure 3 shows an example for the interpolation between two prompt embeddings. It can be observed that the perceived style and content are also interpolated after generating the corresponding images using Stable Diffusion. CLIP embeddings and Stable Diffusion can thus be called robust in this regard.

Our proposed methods use small-step adjustments of prompt embeddings induced by gradient descent or defined by small steps along a SLERP interpolation, and larger adjustments created by directly performing SLERP interpolation between the embeddings of two prompts. This allows for a fine-grained and effective control when modifying the prompt embeddings. It should also be noted that an interpolation between the initial latents (which are initialized randomly using a seed) is possible. While SLERP often seems to be feasible for this task, problems have been observed with this method, leading to the development of more advanced methods (Samuel et al. 2023).

2.3 Related Work

The concept of optimizing prompts has also been investigated in the context of language models that are used to generate text. Here, text prompts can be considered discrete, requiring special methods for optimization (Deng et al. 2022). Continous prompt embeddings have been introduced, allowing a training without finetuning the used model itself (Liu et al. 2021; Lester, Al-Rfou, and Constant 2021).

In the context of generative text-to-image models, the process of prompt engineering can be supported by frameworks like the AUTOMATIC1111 web UI,[3] which provides useful tools for suggesting prompt modifiers or modifying specific image areas (inpainting).

One idea to provide the text-to-image model with the information that would usually be given in the prompt is by making it possible to input images. While inpainting is a very direct method as it simply copies information between the image spaces, other methods have been introduced that allow for a more indirect and complex interaction with the provided images. Textual inversion (Gal et al. 2023) takes images of existing objects and learns a concept, which can then be referred to in custom prompts without having to precisely describe the learned concept verbally. This is achieved by finetuning the weights of the diffusion model. Some related methods use a similar technique (Ruiz et al. 2022; Han et al. 2023). Other methods allow to modify given real images by computing a prompt embedding that would generate the given image, and modifying this embedding based on a modification of a textual prompt (Hertz et al. 2023; Mokady et al. 2022; Li et al. 2023). Methods like LDEdit (Chandramouli and Gandikota 2022) can be framed as instead computing a latent that could be used to generate a given image, and then generating a new image using a modified prompt.

In order to incorporate human feedback, e.g., through an aesthetics metric, it is also possible to finetune the diffusion model (Black et al. 2023). Compared to modifying the underlying prompt embedding, this is expensive and does not reflect the process of prompt engineering. Human feedback can also be used to finetune the CLIP encoder to align the models better with the users' preference (Wu et al. 2023). This has a direct effect on the prompt embeddings, but does not allow for individual adjustments of single prompts. Human feedback in the form of binary ratings can also be used to iteratively modify the weights of the U-Net's self-attention module, which results in an individual optimization for a given prompt (von Rütte et al. 2023).

It can be helpful to provide the model with information in different input modalities. ControlNet (Zhang and Agrawala 2023) allows to extend Stable Diffusion so that it can be finetuned on accepting, e.g., segmentation maps, depth maps, or human scribbles. Different ways to control diffusion models go as far as using brain activity instead of text prompts (Takagi and Nishimoto 2023).

2.4 Prompt Datasets

The experiments in this paper require a diverse range of prompts. For the evaluation and some of the figures, we used a subset of the prompts from DiffusionDB (Wang et al. 2023). For the user study in Section 4.2, some initial prompts came from https://lexica.art, a database of well-engineered prompts, where we removed some of the contained prompt modifiers. All used prompts are given in the supplementary material.

3 Methodology

This section introduces our proposed methods. Experimental results and figures can be found in Section 4. Figure 1 gives an overview of the proposed methods.

3.1 Metric-Based Optimization

During prompt engineering, users often use prompt modifiers to achieve a certain style or aesthetic. This is often achieved by, e.g., appending phrases like `4k high resolution award-winning image`. These modifiers seem highly arbitrary. Our proposed method optimizes the embedding of a given prompt with respect to a metric defined on the image space. If the style sought by the user can be expressed using a such a metric and its gradients can be computed, our

[3] https://github.com/AUTOMATIC1111/stable-diffusion-webui

method can automatically improve the prompt embedding and provide the user with better images.

In the usual process, an image $\mathcal{I}$ is generated from a prompt $\mathcal{P}$ by embedding the prompt and applying the Latent Diffusion Model:

$$\mathcal{I} = \mathrm{LDM}(\psi(\mathcal{P})) \tag{1}$$

Given a metric m, we use gradient descent (or gradient ascent in case the metric denotes an improvement by an increasing value) to optimize the prompt embeddings with

$$\mathcal{C}^* = \arg\min_{\mathcal{C}} m(\mathrm{LDM}(\mathcal{C})), \tag{2}$$

where the prompt embeddings are initialized as

$$\mathcal{C} = \psi(\mathcal{P}). \tag{3}$$

The resulting image is

$$\mathcal{I}^* = \mathrm{LDM}(\mathcal{C}^*). \tag{4}$$

It should be noted that we do not update the model weights during our training (i.e., perform finetuning). Using gradient descent allows to apply relatively small modifications to the prompt embedding, leaving most aspects of the generated image intact, while still optimizing with respect to the desired metric.

During the optimization, we keep the used seed fixed. However, we will discuss the generalization across seeds in our experiments (Section 4.1).

We implement our method for three metrics: One pair of simple metrics (blurriness and sharpness) and a complex deep-learning based aesthetic metric. The blurriness metric is defined by converting the image to grayscale, computing the discrete Laplacian by applying a 2D nine-point stencil via convolution, and returning the variance of the Laplacian. The sharpness metric is simply defined as the negative of the blurriness metric. To describe the aesthetic quality of an image in the pixel space, we need to employ human ratings. We do this by using the pretrained LAION aesthetic predictor.[4,5] Its score is determined by first computing the CLIP embedding of the image that is to be evaluated, and then feeding it into a linear model that has been pretrained to predict a score between 1 and 10 based on 176,000 human ratings of images. This pipeline forms a metric that we use to describe and optimize aesthetic quality. Note that all three metrics allow their gradients to be computed automatically, making them feasible for the proposed method.

3.2 Iterative Human Feedback

Generative text-to-image models are often used for creative tasks where a general theme is given, but the user does not have a specific target image in mind. Users can vary the seed to gain inspiration, but this method is quite limited and lacks control. In the context of prompt engineering, this can lead to a process of trial and error where users apply different prompt modifiers to improve their prompt locally. Our goal is to iteratively provide inspiration to the user in the form of suggested related images based on a modified prompt embedding.

After initializing the current prompt embedding as $\mathcal{C} = \psi(\mathcal{P})$ with a initial prompt $\mathcal{P}$, each step is defined as follows: To generate choices for the user to select from, we generate prompt embeddings $\hat{\mathcal{C}}_i$ as

$$\hat{\mathcal{C}}_i = \mathrm{SLERP}(\mathcal{C}, \tilde{\mathcal{C}}_i, c_i), \tag{5}$$

where the prompt embeddings $\tilde{\mathcal{C}}_i$ are generated from random prompts $\tilde{\mathcal{P}}_i$, which are mainly created by concatenating random alphanumeric characters. From a large set of such potential candidates, a subset is selected that approximates a maximum pairwise cosine distance. This creates a diverse range of prompt embedding candidates. The interpolation parameter c_i is chosen to keep $\mathcal{C} \cdot \hat{\mathcal{C}}_i$ constant and equal for each individual choice, allowing for an equal perceived distance of the choice from the current prompt embedding.

In a second step identical to the above, we modify the embedding $\hat{\mathcal{C}}_i$ towards the original prompt (also in this case modified by a randomly selected prompt modifier from a list of established modifiers). This re-introduces aesthetic quality and prevents the interative method from diverging too much from the original meaning.

The choices given to the user are thus

$$\hat{\mathcal{I}}_i = \mathrm{LDM}(\hat{\mathcal{C}}_i). \tag{6}$$

The user is now able to select a choice i and assign an interpolation parameter $\alpha \in [0, 1]$, that is used to determine the new current prompt embedding for the next step as

$$\mathring{\mathcal{C}} = \mathrm{SLERP}(\mathcal{C}, \hat{\mathcal{I}}_i, \alpha). \tag{7}$$

The new current image can now be displayed as

$$\mathring{\mathcal{I}} = \mathrm{LDM}(\mathring{\mathcal{C}}). \tag{8}$$

Again, this method can be considered an optimization, where each step locally optimizes the user satisfaction. During the iterative process, we keep the used seed fixed to improve the predictability of the results.

Figure 4 shows an implementation of the user interface for this method. In each step, the user can choose between five options.

3.3 Seed-Invariant Prompt Embeddings

During the process of prompt engineering, users typically try out different seeds to seek inspiration. If they discover something interesting, e.g., an object or style, the users typically try to verbalize this aspect to include it in the prompt, which can be very difficult. As shown in Figure 5, the seed can have a large effect when using certain prompts. If the user's satisfaction depends on the seed, this may indicate that the prompt does not contain all the necessary information. We propose an automatic method to remove the underspecification of the prompt (Hutchinson, Baldridge, and Prabhakaran 2022) by modifying the prompt embeddings directly.

Given a target image $\mathcal{I}$ created using a prompt $\mathcal{P}$ and an initial latent z, the goal is to find a prompt embedding $\mathcal{C}^*$ such that

$$\mathrm{LDM}(\psi(\mathcal{P}), z) = \mathrm{LDM}(\mathcal{C}^*, \tilde{z}) \tag{9}$$

[4] https://laion.ai/blog/laion-aesthetics/

[5] https://github.com/christophschuhmann/improved-aesthetic-predictor

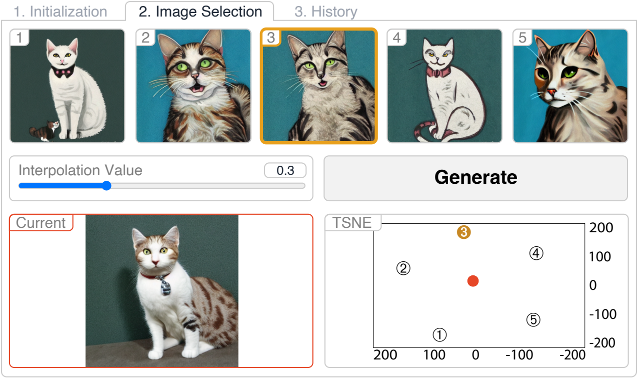

Figure 4: The user interface for our iterative human feedback method. The current image is shown on the bottom left. The choices are shown at the top. The bottom right shows a t-SNE (van der Maaten and Hinton 2008) dimensionality reduction of the current embedding in the center and the five options scattered around.

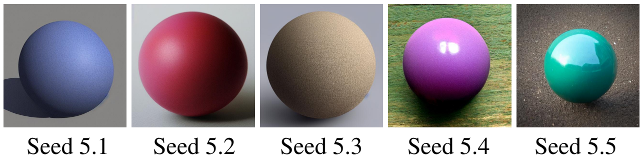

Figure 5: Selected images generated with the prompt `Single Color Ball` and different random seeds.

for any feasibly initial latent $\tilde{z}$.

The pseudocode in Algorithm 1 outlines the proposed method.

This algorithm uses gradient descent to optimize a loss w.r.t. the current prompt embedding $\mathcal{C}$, bringing its image for random seeds closer to the target image. It should be noted that the random seeds are only introduced gradually using the interpolation parameter α. It is feasible to restrict the output of $\text{LDM}(\dots)$ to only the first latents in the beginning.

To further illustrate the proposed method, we will use an oversimplified example. We will reuse the images from Figure 5 and will try to reach a prompt embedding which still shows an image like that of Seed 5.1 when prompted with a different seed like Seed 5.2. We simplify the algorithm above by restricting the space for $\mathcal{C}$ to a one-dimensional interpolation between the prompt embeddings of `Single Color Ball` and `Blue Single Color Ball`. This setup is shown in Figure 6. If our intuition about our method is correct, our $\mathcal{C}$ will move towards a prompt that encodes seed-specific information about our target image. This means that the curve in Figure 6 should move up towards a positive β as our α increases. As can be seen from this experimental result, this is the case.

Algorithm 1: Seed-Invariant Prompt Embeddings

```
$\mathcal{I} \leftarrow \text{LDM}(\psi(\mathcal{P}), z)$
$\mathcal{C} \leftarrow \psi(\mathcal{P})$
for $\alpha \leftarrow \frac{1}{n}, \dots, \frac{n}{n}$ do
    Sample $\tilde{z}$ as a batch of random initial latents
    $L \leftarrow \|\mathcal{I} - \text{LDM}(\mathcal{C}, \text{SLERP}(z, \tilde{z}, \alpha))\|_2^2$
    $\mathcal{C} \leftarrow \mathcal{C} - \eta \nabla_{\mathcal{C}} L$
end for
return $\mathcal{C}$
```

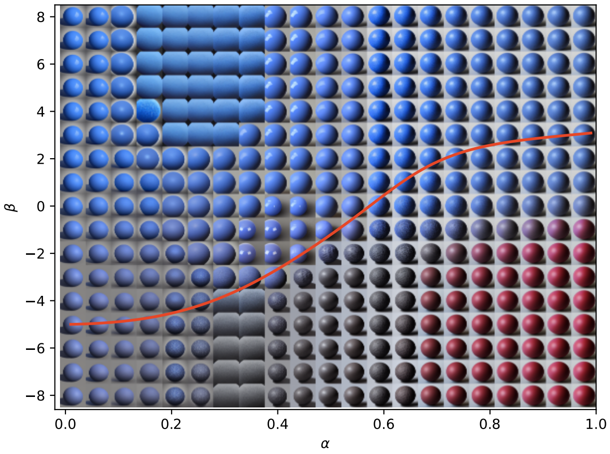

Figure 6: Traversing the prompt embedding space for a gradually modified seed. α denotes the SLERP interpolation parameter between two seeds Seed 5.1 (left) and Seed 5.2 (right). The ordinate represents the prompt embedding space with $\text{sigmoid}(\beta)$ denoting the SLERP interpolation parameter between `Single Color Ball` (bottom) and `Blue Single Color Ball` (top). The orange curve denotes the learned β for each α step.

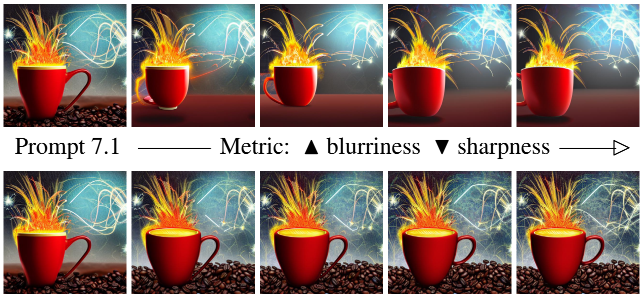

Figure 7: Selected examples of optimizing metrics blurriness (top) and sharpness (bottom).

4 Experimental Results

For the computation of our experimental results, we used Pytorch 2.0 under Python 3.10 in a dockerized Ubuntu system on an A100 GPU. However, not the full memory of the GPU was used as Stable Diffusion is able to run with 8 GB of VRAM. Further details can be found in the supplementary material.

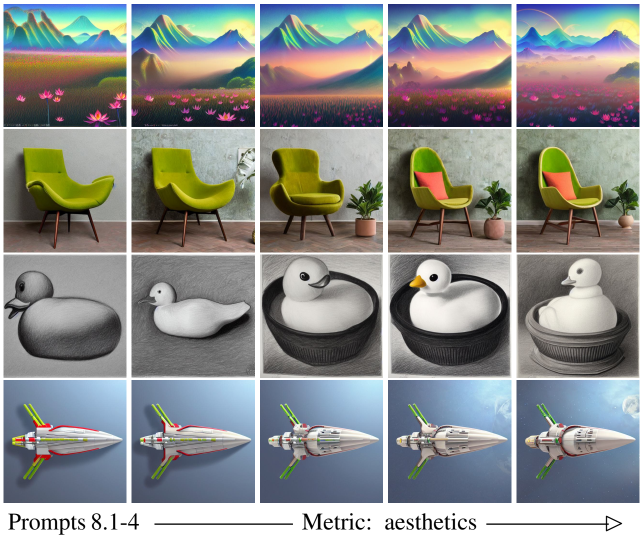

Figure 8: Selected examples of optimizing the aesthetics metric.

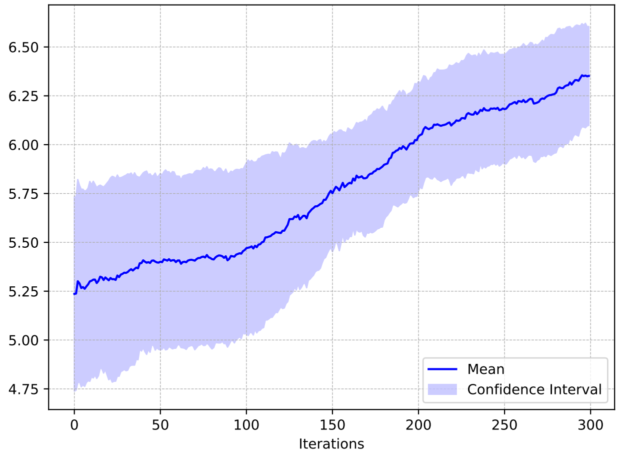

Figure 9: Values of the aesthetic metric over the iteration steps of the metric-based optimization (Section 3.1) for 65 different seeds.

4.1 Metric-Based Optimization

Figure 7 shows the images generated from the updated prompt embeddings at selected time steps for the optimization of the blurriness and the sharpness metric for a single initial prompt. In Figure 8, results of the optimization of the aesthetic metric are shown in a similar way. By comparing the different initial prompts it can be seen that the modified aspects of the images depend on the used prompt. Nevertheless, the results are very promising.

Note that the specific values of a metric required for an image to be perceived as optimal depend on the specific prompt used. Therefore, we propose to leave it to the user to inspect the images generated in increasing iterations so they can terminate the method. Continuing the optimization beyond this point shows that the used metrics can be prone to overfitting. For the blurriness and sharpness metrics, this results in an image with artifacts. This could indicate that the direction implied by the metric's gradient is outside of the prompt embedding space that Stable Diffusion is trained on (see Section 2.2). The aesthetic metric does not seem to have this problem because it takes such effects into account. However, it is possible to optimize the images to the point where they no longer fit the original prompt.

When using or developing new prompt modifiers, users often want them to have the desired effect regardless of the random seed used. They sometimes need the flexibility of being able to change the seed used as a tool to adjust certain aspects of the image, such as composition, or to seek creative inspiration. Finding prompt modifiers that work independently of the seed is very helpful in this regard. We hoped to see a similar effect for our method: Despite restricting the optimization to a single seed, the modified prompt embedding should also provide an improvement regarding the metric compared to the original prompt when being applied on different seeds. To investigate this idea, we ran the optimization of the aesthetic metric for the prompt `highly detailed photoreal eldritch biomechanical rock monoliths, stone obelisks, aurora borealis, psychedelic` for a single seed, and stored the updated prompt embeddings for each iteration. For 65 different seeds, we now computed the values of the aesthetic metrics for these prompt embeddings. The results can be seen in Figure 9. Not only does it show a general trend of an improving metric, it also shows a narrowing confidence interval. It can be concluded that the modified prompt embeddings are at least to some extent independent of the seed used. One could also imagine more complex methods for the optimization, which could involve multiple seeds at runtime (cf. Section 3.3), but the results shown are nevertheless remarkable.

4.2 Iterative Human Feedback

In a user study with 8 participants, we asked users to use our method to create an image fitting to a given description, following their individual preferences. To compare this approach with prompt engineering, we also implemented a similar user interface as a reference baseline. For each method, the users had 20 iterations to come up with an optimal image. During the process, they were asked to describe their approach. Afterwards, they were asked for a relative ranking between the optimal images for both methods. Details can be found in the supplementary material. Figure 10 shows selected results.

We noticed that our method is especially helpful for creative tasks, where the user does not have a clear target image in mind. This could be noticed in the different behavior of users that first used our proposed method versus users that first performed prompt engineering. The latter case can be considered a limitation of the method: In this case, the user is dependent on being shown suggestions pointing into the direction of a desired target. Our method is also feasible for users with limited experience in prompt engineering, for whom the latter has been a rather frustrating experience. Our method was found to be less tedious, and six users even

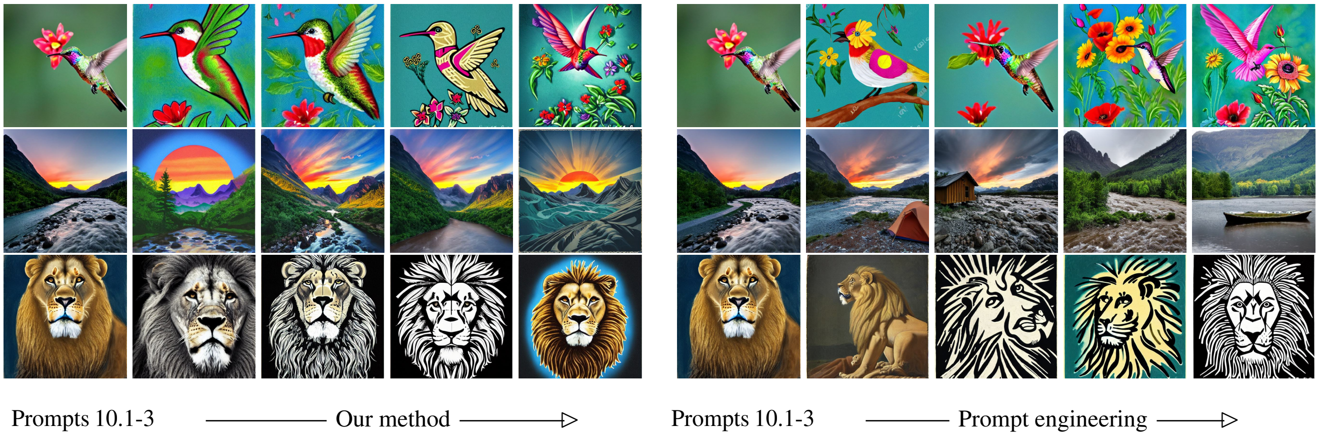

Figure 10: Selected examples for images created in our user study using our method based on iterative human feedback and using prompt engineering.

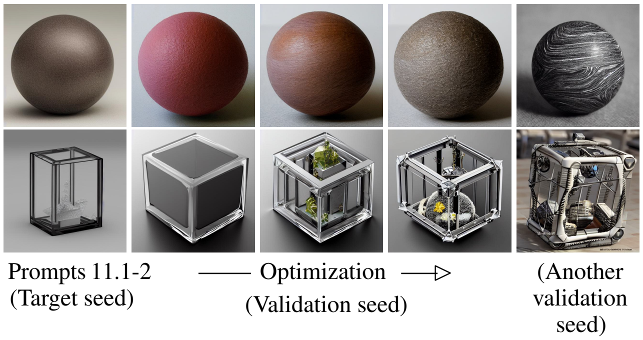

Figure 11: Selected examples of the unguided seed-invariant prompt embedding method.

preferred the image generated by our method to the one generated by prompt engineering. Contrary to the findings in Section 4.1, the prompt embeddings generated in this experiment did not generalize across the given seed, as the relative ratings seemed to differ when the seed for the optimal prompt embeddings for both methods was changed.

4.3 Seed-Invariant Prompt Embeddings

In a less restricted experiment than the one in Section 3.3, we inspect the feasibility of our implementation for more general problems. Now, we directly optimize the highly-dimensional embedding $\mathcal{C}$ without providing a low-dimensional subspace tailored for this specific experiment.

Figure 11 shows the first experimental results. They show that the current implementation is capable of sensing a general direction of optimization, but lacks precision, especially for complex prompts. We hope that this limitation could be overcome by borrowing implementation details from approaches like Mokady et al. (2022).

5 Conclusion

In this paper, we introduced three methods for modifying the embeddings of Stable Diffusion prompts. Supported by our experimental results, we are able to identify a common use case for the proposed methods: Prompt engineering often consists of iteratively refining an existing prompt. As seen in our user study, this is true in the case of creative tasks, where the user seeks inspiration, as well as in the case where the user has a fixed target image in mind. Our proposed methods provide support in both scenarios. In addition, the optimization of a metric can be used to modify the image in the case where the user simply wants to improve the image but does not know how. By introducing and evaluating these methods, we are able to demonstrate the feasibility of prompt embedding manipulation. This paper contributes to improving the user experience when using generative text-to-image models by allowing prompt engineering to be bypassed in certain scenarios, thereby increasing the accessibility of the models.

Our methods can be generalized beyond Stable Diffusion, as other models have a similar architecture.

Future applications of this work revolve around the idea of reusing the optimized prompt embeddings. Due to their demonstrated robustness (in some cases even with invariance with respect to the seeds), they can potentially be used to improve other prompts. This is already possible with an interpolation. However, different ways of integrating them could be investigated (cf. Section 2.3). It might also be possible to share the optimized prompt embeddings with a community, similar to e.g. Lexica.

Another idea would be to extend the seed invariance of the prompt embeddings to a prompt invariance of the introduced changes in the prompt embeddings. This could lead to a single representation of, e.g., prompt embeddings with a high aesthetic quality. These prompt embedding modifiers could then be used instead of prompt modifiers.

We have also shown the parallels of our approach to the domain of language models for text generation. One could try to transfer our proposed methods to this or possibly other domains.

References

Balaji, Y.; Nah, S.; Huang, X.; Vahdat, A.; Song, J.; Kreis, K.; Aittala, M.; Aila, T.; Laine, S.; Catanzaro, B.; Karras, T.; and

Liu, M. 2022. eDiff-I: Text-to-Image Diffusion Models with an Ensemble of Expert Denoisers. *CoRR*, abs/2211.01324.

Black, K.; Janner, M.; Du, Y.; Kostrikov, I.; and Levine, S. 2023. Training Diffusion Models with Reinforcement Learning. *CoRR*, abs/2305.13301.

Chandramouli, P.; and Gandikota, K. V. 2022. LDEdit: Towards Generalized Text Guided Image Manipulation via Latent Diffusion Models. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*, 267. BMVA Press.

Chang, H.; Zhang, H.; Barber, J.; Maschinot, A.; Lezama, J.; Jiang, L.; Yang, M.; Murphy, K.; Freeman, W. T.; Rubinstein, M.; Li, Y.; and Krishnan, D. 2023. Muse: Text-To-Image Generation via Masked Generative Transformers. *CoRR*, abs/2301.00704.

Deckers, N.; Fröbe, M.; Kiesel, J.; Pandolfo, G.; Schröder, C.; Stein, B.; and Potthast, M. 2023. The Infinite Index: Information Retrieval on Generative Text-To-Image Models. In Gwizdka, J.; and Rieh, S. Y., eds., *ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR 2023)*, 172–186. ACM.

Deng, M.; Wang, J.; Hsieh, C.; Wang, Y.; Guo, H.; Shu, T.; Song, M.; Xing, E. P.; and Hu, Z. 2022. RLPrompt: Optimizing Discrete Text Prompts with Reinforcement Learning. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 3369–3391. Association for Computational Linguistics.

Gal, R.; Alaluf, Y.; Atzmon, Y.; Patashnik, O.; Bermano, A. H.; Chechik, G.; and Cohen-Or, D. 2023. An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Han, L.; Li, Y.; Zhang, H.; Milanfar, P.; Metaxas, D. N.; and Yang, F. 2023. SVDiff: Compact Parameter Space for Diffusion Fine-Tuning. *CoRR*, abs/2303.11305.

Hao, Y.; Chi, Z.; Dong, L.; and Wei, F. 2022. Optimizing Prompts for Text-to-Image Generation. *CoRR*, abs/2212.09611.

Hertz, A.; Mokady, R.; Tenenbaum, J.; Aberman, K.; Pritch, Y.; and Cohen-Or, D. 2023. 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.

Hutchinson, B.; Baldridge, J.; and Prabhakaran, V. 2022. Underspecification in Scene Description-to-Depiction Tasks. In He, Y.; Ji, H.; Liu, Y.; Li, S.; Chang, C.; Poria, S.; Lin, C.; Buntine, W. L.; Liakata, M.; Yan, H.; Yan, Z.; Ruder, S.; Wan, X.; Arana-Catania, M.; Wei, Z.; Huang, H.; Wu, J.; Day, M.; Liu, P.; and Xu, R., eds., *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*, 1172–1184. Association for Computational Linguistics.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In Moens, M.; Huang, X.; Specia, L.; and Yih, S. W., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, 3045–3059. Association for Computational Linguistics.

Li, S.; van de Weijer, J.; Hu, T.; Khan, F. S.; Hou, Q.; Wang, Y.; and Yang, J. 2023. StyleDiffusion: Prompt-Embedding Inversion for Text-Based Editing. *CoRR*, abs/2303.15649.

Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; and Tang, J. 2021. GPT Understands, Too. *CoRR*, abs/2103.10385.

Mokady, R.; Hertz, A.; Aberman, K.; Pritch, Y.; and Cohen-Or, D. 2022. Null-text Inversion for Editing Real Images using Guided Diffusion Models. *CoRR*, abs/2211.09794.

Nichol, A. Q.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; and Chen, M. 2022. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvári, C.; Niu, G.; and Sabato, S., eds., *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 16784–16804. PMLR.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, 8748–8763. PMLR.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *CoRR*, abs/1910.10683.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical Text-Conditional Image Generation with CLIP Latents. *CoRR*, abs/2204.06125.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, 10674–10685. IEEE.

Ruiz, N.; Li, Y.; Jampani, V.; Pritch, Y.; Rubinstein, M.; and Aberman, K. 2022. DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. *CoRR*, abs/2208.12242.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E. L.; Ghasemipour, S. K. S.; Lopes, R. G.; Ayan, B. K.; Salimans, T.; Ho, J.; Fleet, D. J.; and Norouzi, M. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In *NeurIPS*.

Samuel, D.; Ben-Ari, R.; Darshan, N.; Maron, H.; and Chechik, G. 2023. Norm-guided latent space exploration for text-to-image generation. *CoRR*, abs/2306.08687.

Schuhmann, C.; Beaumont, R.; Vencu, R.; Gordon, C.; Wightman, R.; Cherti, M.; Coombes, T.; Katta, A.; Mullis, C.; Wortsman, M.; Schramowski, P.; Kundurthy, S.; Crowson, K.; Schmidt, L.; Kaczmarczyk, R.; and Jitsev, J. 2022. LAION-5B: An open large-scale dataset for training next generation image-text models. In *NeurIPS*.

Shoemake, K. 1985. Animating rotation with quaternion curves. In Cole, P.; Heilman, R.; and Barsky, B. A., eds., *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1985, San Francisco, California, USA, July 22-26, 1985*, 245–254. ACM.

Sohl-Dickstein, J.; Weiss, E. A.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In Bach, F. R.; and Blei, D. M., eds., *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*, 2256–2265. JMLR.org.

Takagi, Y.; and Nishimoto, S. 2023. High-resolution image reconstruction with latent diffusion models from human brain activity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14453–14463.

van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5998–6008.

von Rütte, D.; Fedele, E.; Thomm, J.; and Wolf, L. 2023. FABRIC: Personalizing Diffusion Models with Iterative Feedback. *CoRR*, abs/2307.10159.

Wang, Z. J.; Montoya, E.; Munechika, D.; Yang, H.; Hoover, B.; and Chau, D. H. 2023. DiffusionDB: A Large-scale Prompt Gallery Dataset for Text-to-Image Generative Models. In Rogers, A.; Boyd-Graber, J. L.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, 893–911. Association for Computational Linguistics.

Witteveen, S.; and Andrews, M. 2022. Investigating Prompt Engineering in Diffusion Models. *CoRR*, abs/2211.15462.

Wu, X.; Sun, K.; Zhu, F.; Zhao, R.; and Li, H. 2023. Better Aligning Text-to-Image Models with Human Preference. *CoRR*, abs/2303.14420.

Zhang, L.; and Agrawala, M. 2023. Adding Conditional Control to Text-to-Image Diffusion Models. *CoRR*, abs/2302.05543.