Preference Learning for Text-to-Image Prompt Tuning using RL

Arnav Gudibande [*1] **Tyler Zhu** [*1]

Extended Abstract

We propose a method to optimize text prompts for text-to-image diffusion models by using RL to search over a discrete text space. Our pipeline allows users to discover text modifiers that are needed to generate similar images to a given user-submitted image. This effectively automates the process of prompt engineering to discover a suitable prompt to generate a specific goal image.

Method

We propose an extension to RL-Prompt (Deng et al., 2022), by utilizing a Stable Diffusion model in order to generate a reward signal over a discrete text space.

In Figure 1, the user submits some target image t_i they intend to discover text modifiers for. We start with some seed text prompt p_i. The policy module uses the reward signal to propose some additional text modifiers $p_i{}'$ in the entire vocabulary V. The full text prompt is inputted to a frozen stable diffusion model $D(s)$, which takes a text prompt string s and outputs an image. The reward function $R(s,t)$ outputs the similarity between the generated image $D(p_i + p_i{}')$ and the target image t_i. Formally, the optimization objective is as follows:

$$\arg\max_{p_i{}' \in V} R(D(p_i + p_i{}'), t_i) \tag{1}$$

We use the same method in RL-Prompt to optimize over this objective function, which uses the SQL algorithm () to integrate the reward signals into the MLP.

Experimental Setup

Datasets We construct datasets from the larger Krea.ai Open-Prompts dataset. For each pair of prompts and generated images in this dataset, we create a seed prompt by removing several key modifiers from the user submitted prompt. Our goal is to generate similar images to the target image while only assuming the seed prompt as input.

Models We use distillGPT2 as the frozen base LM and StableDiffusionv1.3 as the frozen diffusion model.

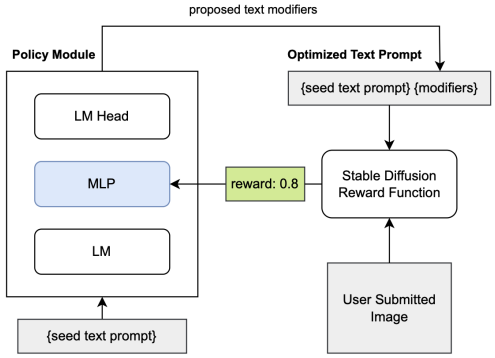

Figure 1. Extending RL-Prompt to Stable Diffusion. The learnable component is the MLP shaded in blue, all other components are frozen. The reward function generates a scalar which is used as input to the MLP. The policy module takes in a seed prompt and a reward to generate a set of proposed text modifiers.

Main Findings

In Figure 2, we report the average similarity of the images generated by the optimized prompts to the target image. We find that querying the model for 2-5 additional text modifiers yields an image similarity higher than the image generated from the seed prompt alone. In doing so, we show that RLPrompt can be extended to diffusion in order to discover text modifiers that improve the similarity of the seed image.

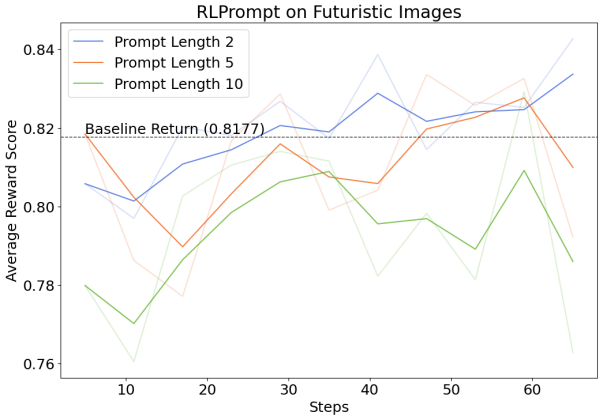

Figure 2. Training curves of our method on the Futuristic dataset over three different choices of prompt lengths. We include a baseline return of the ground truth prompt. We see that prompting the model for 2 additional tokens yields the greatest increase in similarity as compared to the other methods.

[*]Equal contribution [1]UC Berkeley. Correspondence to: Arnav Gudibande <arnavg@berkeley.edu>, Tyler Zhu <tyler.zhu@berkeley.edu>.

UC Berkeley CS285 Fall 2022 Final Course Project

1. Introduction

Text-to-image synthesis has been a hallmark of vision and language research ever since its inception. It requires deep understanding of both domains and how they relate with each other, often in a grounded manner. Within the last few years, research in the area has happened at an unprecedented pace and resulted in photorealistic generations that anyone can use for a wide ranging set of applications.

The difficult part of utilizing such methods is creating prompts that are descriptive enough to get the models to create images exactly as we desire. Much effort has been focused on discovering better prompts to use with these models to delve even deeper into these models knowledge bases. For example, it has been well documented that adding in "unreal engine" leads to a dramatic improvement in the quality of the outputs (Snell, 2021). This led to a wide amassing of highly productive prompts that could give you a specific style of images, dubbed *prompt engineering*. Researchers even discovered emergent behaviors in large language models (LLMs) that allowed for in-context learning if they were prompted with prompts such as "Using a scratchpad" or "Let's think step by step" (Nye et al., 2022; Kojima et al., 2022).

However, it's unclear how to find such prompts in an automated fashion, as the engineering effort necessary (as indicated in the name) proves quite cumbersome when thinking about applying these methods at scale. This is especially concerning given the trend in which techniques are going in. From a practical standpoint, it seems undesirable for humans to explore the high-dimensional space of prompts on a trial-and-error basis. Thus it is quite attractive to create a technique which could learn an optimal set of prompts for a specific purpose. A key intuition is that neural networks should be able to learn how to navigate these high-dimensional manifolds that these sets of prompts live in. Assuming that our desired images exist within the space of possible output images which our text-to-image models enclose, our focus should be on creating better methods of aligning the inputs with our wishes.

Currently, the workflow for such a process starts with an example prompt, which gets fed into the model, which outputs a set of candidate images, after which the human decides which aspects of the images they like and dislike, and subsequently adjusts the prompt with extra tokens that reflect those changes. This process gets repeated over and over again until the human gets to a point where the images are satisfactory. Our insight is that we can utilize human feedback with reinforcement learning to learn a preference module for individual styles or tasks based on a few examples, inspired by similar works which do this (Ouyang et al., 2022a; Lee et al., 2021). This module can then be used to automatically search through candidate "prompt tokens" to append to a base prompt (corresponding to the starting prompt in our example above) to eventually find the best translation of human preferences to a tokenized prompt.

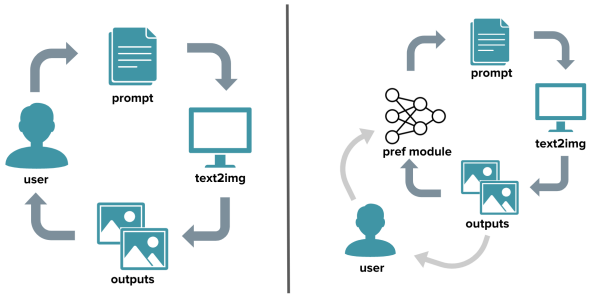

Figure 3. Overview of our framework. On the left is the typical setup of prompt engineering, where a human has to be in the loop to manually adjust prompts based on what they see from the outputs. In our proposed setup, the human is sparsely queried, with the preference module doing the bulk of the work, allowing much faster automation of the tuning process.

In this work, we propose a framework (Figure 3) towards automating prompt tuning for text-to-image synthesis based on human preferences using reinforcement learning with human feedback. Due to computational and time constraints, we develop our tasks in simplified situations which are representative of the real-world settings that we are attempting to model. However, we still achieve promising results which indicate a potential for greater performances at larger scales.

2. Related Work

Text-to-image synthesis Photorealistic text-to-image synthesis has been a long standing goal of the vision-language communities. As a task, it requires the ability to coherently understand input text while also being able to correlate that understanding with an understanding of visual data. All the while it also needs to be able to generate visually diverse outputs given a single input prompt to express the large distribution of possible results. Many approaches have been proposed to tackle this issue, the most popular of which utilize Generative Adversarial Networks (GANs) by conditioning them on a text latent input to steer them towards our desired outputs (Reed et al., 2016; Zhang et al., 2017; Dash et al., 2017).

However, despite much effort towards conditioning GANs on inputs, they are difficult to control and generate specific examples from. Other methods found more success by utilizing multimodal embeddings of text and vision trained jointly as supervisory signal and instead turning to transformer-based approaches which can dually handle both images and text in one format (Ramesh et al., 2021; Radford et al.,

Figure 4. Examples of generated images given the seed prompt (left) and our completions using learned prompts (middle) compared to the ground truth result image (right). We see that the images in the middle column are semantically closer to the target images in the right column than the baseline images in the left column. In all cases, we see that the style and content is sufficiently preserved between the target images and our generated images.

2021). These results were suddenly plausibly photorealistic and could be generated in a much more diverse manner than previously thought possible. Recently, the introduction of diffusion models has led to another explosion of text-to-image synthesis, resulting in DALLE v2 which has captured mainstream media by a storm (Ramesh et al., 2022).

This coincided with the rise of open-sourced model efforts after models such as GPT-3 weren't released to the public. The most well known model is likely the Stable Diffusion models based on (Rombach et al., 2022) from Stability.ai and RunwayML. This model can run on GPUs with only 10GB of VRAM by performing diffusion in the latent space, keeping things small and fast and allowing anyone to do text-to-image synthesis on commodity hardware. Due to the open sourced and lightweight nature of the model, we opt to use the Stable Diffusion v1.3 and v1.4 checkpoints for our experiments, but our method is agnostic to the choice of text-to-image synthesizer.

Prompt tuning Several gradient-based methods exist to tune language prompts for performance on downstream natural language tasks. These methods (Wallace et al., 2019) (Shin et al., 2020) work by creating a number of "[MASK]" tokens inside of the natural language prompt, then iteratively backpropogating through the model to optimize those tokens to maximize the label likelihood for a particular downstream task. This has been shown to be an effective way to optimize prompts for natural language tasks. However, it should be noted that these methods require propagating through a task model – which requires both access to the weights as well as a large amount of compute. As a result, these gradient based techniques will not be practical for using proprietary model APIs, where weights are unknown, or very large models, where compute resources will quickly become a limitation.

Reinforcement learning from human feedback Reinforcement learning from human feedback (RLHF), is a technique that has been recently popularized by OpenAI, who have used it to instruction-tune versions of GPT-3 (Ouyang et al., 2022b). These models have been shown to be highly expressive and capable at following human instructions on a wide variety of tasks. At a high level, RLHF is a method that takes into account human feedback by using a relatively small amount of human demonstrations and preferences to train a preference model – which learns how to rank the quality of outputs like a human does. This model is then used to provide reward signals when training the final model which takes into account human preferences. This method has been shown to align the model with human preferences, while being fairly data efficient.

Soft Q-Learning for Text Generation The authors of (Guo et al., 2022) present a modification to Soft Q-Learning (Haarnoja et al., 2017), originally introduced in robotics, which allows it to work with text generation models. They reformulate SQL to Q-values as being logits from a text generation model. As they mention in their paper, this is equivalent to treating the probability of a next token being generated given the previous tokens, as the probablity of a given action given some state. This formulation allows SQL to be applied to text generation problems. Overall, this method was shown by Guo et. al to perform well in settings with sparse reward signals such as open ended text generation. Later, it was adapted by RL-Prompt for other kinds of supervised text tasks with limited data.

RL-Prompt The authors of RL-Prompt (Deng et al., 2022) propose an RL based method for learning an optimal set of prompts for the tasks of few-shot text classification and text style transfer. In particular. For the task of few-shot classification, the goal of the model is to predict the sentiment of a sentence with high accuracy. This is accomplished by using a masked language model (MLM) such as RoBERTa

(Liu et al., 2019) to predict the most likely token for the following template "this food is delicious! [MASK]", where MASK can be one of "great" or "terrible". They then use RL to learn optimal prompts to fill in the template "[Input] [Prompt] [MASK]". The RL-Prompt pipeline is similar to Figure1, with the exception of their reward function being a metric that measures the accuracy of the MASK tokens across an evaluation set. The authors use Soft Q-Learning (Guo et al., 2022) as defined above in order to optimize the following objective, where $y_{LM}(p_i', x)$ is the output of the language model on some input x prompted by p_i' and R is a reward function.

$$\arg\max_{p_i' \in V} R(y_{LM}(p_i', x)) \tag{2}$$

They also test this method on a text-style transfer task, and show that this technique can adapt to using uni-directional LMs such as GPT2 (Radford et al., 2019) on generative tasks. The authors find that their pipeline discovers novel, and mostly semantically varied prompt tokens that improve the performance of frozen sentiment classification models on public benchmarks.

3. Method

In the previous section, we explained how the RL-Prompt (Deng et al., 2022) pipeline and algorithm works for the tasks of few-shot text classification and text style transfer. Now, we discuss how we extend RLPrompt to work in the setting of optimizing prompts for diffusion models.

3.1. Extending RLPrompt to Diffusion

We consider a particular task in order to learn optimized text prompts for stable diffusion models. In particular, we intend to discover text modifiers that can be appended onto a seed prompt in order to learn semantically similar images for a fixed user submitted image. This would allow users to automatically discover what text modifiers are necessary for generating a particular image given a particular stable diffusion model. Moreover, the framing of this task allows the method to be easily extended to any diffusion model. We discuss some terminology used henceforth below.

Target Image A user submits a target image t_i, which they intend to discover certain text modifiers for. This image can be of any dimension that is supported as an output from stable diffusion.

Seed Prompt We assume each image has some corresponding seed prompt p_i, or a partial prompt that specifies the content of the image. In our testing, we synthetically craft seed prompts which have a minimum length and level of descriptive granularity. However, this can easily be extended to harder use-cases, where seed prompts are shorter and have a lower level of descriptiveness.

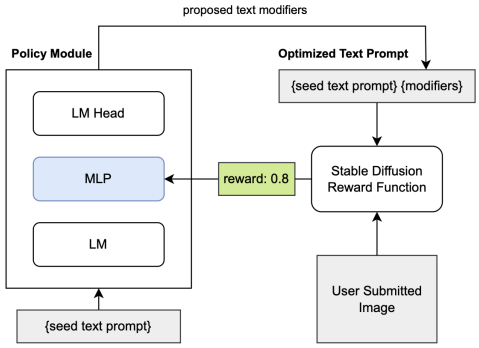

Figure 5. Extending RL-Prompt to Stable Diffusion. The learnable component is the MLP shaded in blue, all other components are frozen. The reward function generates a scalar which is used as input to the MLP. The policy module takes in a seed prompt and a reward to generate a set of proposed text modifiers.

Text Modifiers The output of our pipeline are text modifiers, which consist of a fixed number of tokens that can be appended onto a seed text prompt in order to yield an image that is semantically similar to the target image. These text modifiers must be tokens that are present in the tokenizer, and are usually dependent on the selected diffusion model used in the pipeline.

In Figure 5, we illustrate how our pipeline extends the RL-Prompt (Deng et al., 2022) architecture to work with this objective. In particular, we add a Stable Diffusion Reward Function, which uses a frozen diffusion model to generate images from the proposed optimized text prompt $p_i + p_i{}'$. This function returns a scalar value, which represents the reward of using particular text modifiers $p_i{}'$.

As mentioned earlier, we start with some seed text prompt p_i. The policy module uses the reward signal to propose some additional text modifiers $p_i{}'$ in the entire vocabulary V. The full text prompt is inputted to a frozen stable diffusion model $D(s)$, which takes a text prompt string s and outputs an image. The reward function $R(s, t)$ outputs the similarity between the generated image $D(p_i + p_i{}')$ and the target image t_i. Formally, the optimization objective is as follows:

$$\arg\max_{p_i' \in V} R(D(p_i + p_i{}'), t_i) \tag{3}$$

Similar to RLPrompt, we utilize Soft Q-Learning (Guo et al., 2022) in order to optimize this objective. We note that this algorithm is sensitive to choice of reward function. We discuss different design choices as well as their implications for learning below.

Futuristic	Art	Painting
highly detailed	astral	john harris
octane render	bismuth	macabre
artstation	dendritic	rembrandt
4k	earth art	dramatic lighting
...	...	...

Table 1. Examples of salient text modifiers most frequently used in each category of image data-sets we sampled. We removed all these keywords from the selected text prompts in order to create a seed prompt. This ensures our method has enough freedom to discover possibly novel text modifiers that can move the baseline image semantically closer to the target image.

a city made entirely out of Rubik's cubes, daylight, sunlight, lens flare, digital painting, smooth, sharp focus, photorealistic, 25mm f/1.7 ASPH Lens, ultra realistic steampunk illustration, art by greg rutkowski and alphonse mucha

a city made entirely out of Rubik's cubes, daylight, sunlight, lens flare, **highly detailed, digital painting**, **artstation**, **concept art**, smooth, sharp focus, **8k**, photorealistic, 25mm f/1.7 ASPH Lens, ultra realistic steampunk illustration, art by greg rutkowski and alphonse **mucha**

Figure 6. Example of user written prompt and corresponding generated image in the futuristic dataset on the right. We highlight keywords that belong to the futuristic category in the user generated prompt. Some examples are "highly detailed", "artstation" and others as detailed in Table 3.3. The prompt on the left shows the image with those category modifiers removed. We see that the image on the left is different from the right image in a few subtle ways – it lacks detail, resolution, and other details that would otherwise be shared between images in the Futuristic data category. As a result, the image on the left is used as a baseline starting point that our algorithm will be able to iterate on.

3.2. Models

For the language model component, we use a frozen distillGPT2, an 82M parameter language model that was pretrained on openwebtext. The model is split into two parts, an LM component which embeds the seed prompt and a LM head, which receives an input from the MLP before decoding the output.

For the diffusion model, we use the latest diffusion models from Stability.ai, StableDiffusionv1.3 and StableDiffusionv1.4 (Rombach et al., 2022). These models were pretrained on subsets of the LAION-5B (Schuhmann et al., 2022) dataset. These models embed text using a frozen CLIP ViT-L/14 text encoder.

3.3. Dataset

We use Krea.ai's Open-Prompts (Krea.ai, 2022) dataset, which consists of 10 million user submitted text prompts and image pairs scraped from Discord during the beta testing of StableDiffusionv1.3. We filter this dataset into particular categories in order to constrain the training and evaluation procedure. We also use salient text-modifiers from Open-Prompts in order to define these certain categories.

In particular, we create 3 datasets – Futuristic, Art, and Painting. As described in Table 3.3, each of these datasets consists of images that contain keywords according to 1 of 3 categories. Images are deemed to be in a particular category if their corresponding text prompt has a sufficient text overlap with category modifiers. We then proceed to construct seed prompts for each image in the dataset. We do this by removing at least 3 text modifiers from each of the user-written prompts. This yields an image which is a baseline starting point from which we can further discover further text modifiers to add.

In Figure 6, we show an example of a target image and a baseline image in the Futuristic dataset. The target image is generated using an original user submitted prompt, and the baseline image is the image generated after removing particular categorical text modifiers. We see that the baseline image is stylistically similar to the target image, but still leaves enough freedom for the RL-Prompt pipeline to discover modifiers to make the two images semantically similar.

The collected datasets were also relatively small – on the order of 100 image-prompt pairs or less, utilizing a 80/20 train/test split. We follow the same setup as RL-Prompt, whose training setup for few-shot classification was on the order of 50-100 examples or less. We also note that due to the computational requirements of running StableDiffusion on commercial GPUs, we stick to using datasets that are relatively small. However, given access to larger compute infrastructure, our method should be able to scale proportionately to the data size.

3.4. Reward Function Design

The RL-Prompt (Deng et al., 2022) authors discuss the importance of designing well-behaved reward functions for the performance of the RL-Prompt architecture on tasks such as few-shot classification and text style transfer. To begin with, we propose a reward function based on measuring the similarity of the image generated by the optimized text

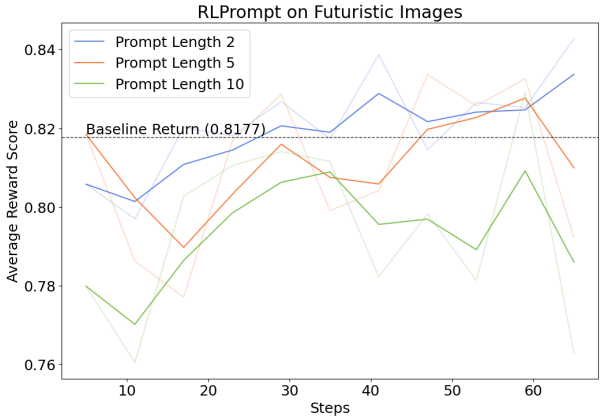

Figure 7. Training curves of our method on the Futuristic dataset over three different choices of prompt lengths. We include a baseline return of the ground truth prompt. As expected, it is quickest for the model to learn how to utilize 2 additional prompt tokens, followed by 5 tokens and then 10 tokens.

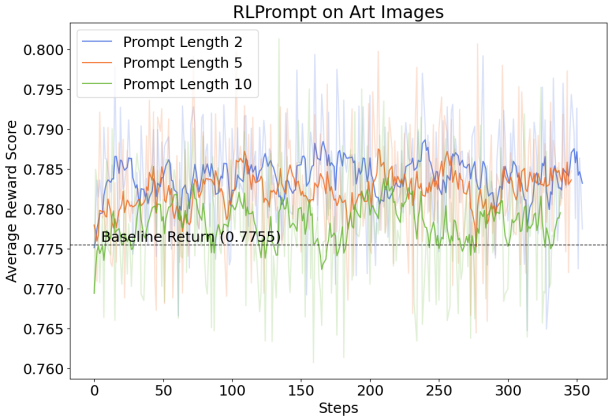

Figure 8. Training curves of our method on the Art dataset over the same three different choices of prompt lengths. However, this time it fails to learn to adapt the *art* preference style. This most likely may be due to how weak the signal in this style is, and could be fixed by collecting a better dataset

prompt with the target image. We hypothesize that this is provides a sufficient reward signal for an agent to search over the vocabulary of possible tokens.

In particular, we design a reward function that is as follows. Given a target image t, a candidate text prompt s, and a diffusion model D, which converts text prompts to images, we define the reward to be the cosine similarity between the generated diffusion image $D(s)$ and the target image t. We note that cosine similarity is a bounded metric – defined between 0 and 1, which allows our reward function to not take on extreme values during the course of training.

$$\mathcal{R}_{image}(s,t) = \frac{D(s) \cdot t}{||D(s)|| \cdot ||t||} \tag{4}$$

We use the above reward function for the majority of experiments. However, we also note that its possible to define a reward function that is defined in the text space, rather than the image space. This would allow us to make use of more information that is stored in the raw text modifiers that is outputted by RL-Prompt. Given a candidate text prompt s, the user written ground truth prompt g for a particular target image t, and a CLIP model that embeds text, we can define cosine similarity in the text space as follows:

$$\mathcal{R}_{text}(s,g) = \frac{CLIP(s) \cdot CLIP(g)}{||CLIP(s)|| \cdot ||CLIP(g)||} \tag{5}$$

We hypothesize that there is a tradeoff between increasing $\mathcal{R}_{image}$ and $\mathcal{R}_{text}$. Ostensibly, higher values of $\mathcal{R}_{text}$ imply that the tokens discovered by RL-Prompt are semantically similar to those that would be written by users. However, higher values of $\mathcal{R}_{image}$ may skew the model to push for images that are more similar to the target image, at the expense of discovering tokens that are semantically not interpretable. We can modulate this tradeoff with the following reward function, which assumes we have a candidate text prompt s, and a batch of ground truth prompts g and corresponding images t.

$$\mathcal{R}(s) = \lambda_i \cdot \mathcal{R}_{image}(s,t) + \lambda_j \cdot \mathcal{R}_{text}(s,g) \tag{6}$$

For simplicity, we set λ_j to 0 for the following experiments.

4. Results

We evaluate our method on a wide suite of different settings to understand its performance better. In an ideal world we would be able to run it averaged out over many stable diffusion samples and over many random seeds. Unfortunately, we were heavily compute and time constrained, utilizing only 3 RTX 2080 Ti's at any given point in time.

Our experiments fall under three broad categories. We first look at the effects of the datasets, followed by an ablation on the choice of backbone model. We end on a discussion of the prompts learned and how we might be able to improve their results.

4.1. Baseline Results on Futuristic and Art Preferences

For the baseline results, we first benchmarked our method on the *futuristic* dataset of (prompt, image) tuples. For this, we ran RLPrompt with the default settings of lr $= 5 \times 10^{-5}$ and on Stable Diffusion v1.4 while varying the prompt lengths between 2, 5, and 10 extra prompt tokens for learning the preference of the dataset. In Figure 7, you can see that the baseline return has an 81.77% similarity to the ground truth. While it may seem like it should be

Table 2. Qualitative comparisons of the generated pictures throughout training for the *art* dataset. The three rows in order correspond to tuning an additional prompt of length 2, 5 and 10 respectively. We notice that the images do gradually become more detailed an artistic, a possible sign that our method is learning the preferences of the dataset.

100%, we chalk this gap to a difference in the model used as our backbone and the model on which the dataset was collected (Stable Diffusion v1.3). In fact, our testing also showed that while across models there sometimes is a big gap between prompts, holding the model constant generally produces images which are highly similar to one another, deviating on average by about 5% similarity.

We observe that between the three prompt lengths, SQL was able to pick up better tokens for the shorter lengths than longer lengths as expected. Near the end of training, we also observed some trials outperforming the baseline in a few amount of steps. This is impressive especially for a gradient-free method, which also validates our use of CLIP similarity as a reward metric.

We also ran a trial on the *art* dataset using the Stable Diffusion model v1.3, with the same settinigs as in the *futuristic* dataset but with more timesteps. The results are in Figure 8. This time our results are much noisier, but the general trend of length 2 prompts being easier to learn than length 5 followed by length 10 prompts still holds. We find that it

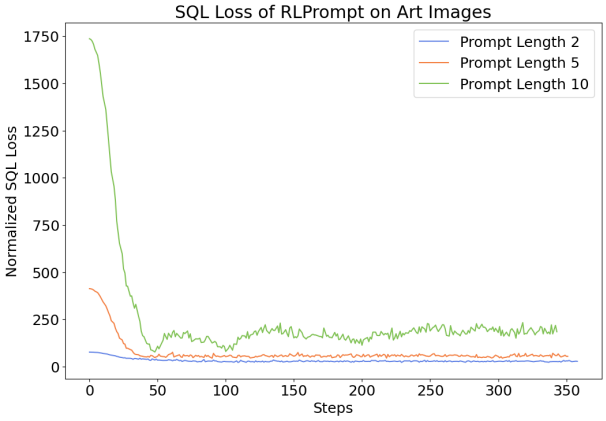

Figure 9. SQL loss on the *art* dataset. Despite our prompts achieving around the same rewards on the images, SQL still managed to quickly reduce loss towards a suitable minima before oscillating around and being unable to converge.

was difficult to learn, with our model essentially hovering around the same spot in which is started. We attribute this to dataset quality as the theme of *art* was not as easily targeted

as the theme of *futuristic* was. However, SQL was still able to minimize its loss to some degree before oscillating, even being more stable than the original RLPrompt, meaning that our framework is sensible and that our rewards and/or dataset likely need improving.

We also provide qualitative results of how our training images evolved over the course of training in Table 2. Qualitatively, we can at least observe some perceivable changes in style, although whether these can be attributed to chance is hard to know.

4.2. Backbone Choice Matters

Given that we had access to multiple versions of stable diffusion, we wanted to see what effect our choice of backbone made on the feasibility of learning. We calculated the similarity rewards of the ground truth prompts with their associated images in the Open Prompts dataset to see how much of an effect the model had. Recall that the dataset was collected with v1.3 of the model.

Dataset	Model	GT Similarity
Futuristic	Stable Diffusion v1.4	81.77%
Futuristic	Stable Diffusion v1.3	82.10%
Art	Stable Diffusion v1.4	19.32%
Art	Stable Diffusion v1.3	77.55%
Painting	Stable Diffusion v1.4	9.25%
Painting	Stable Diffusion v1.3	80.90%

Table 3. Backbone choice can make some preferences easier to learn, but ultimately better data can make up for a difference in models by providing a better signal.

As shown in Table 4.2, we see that the model used has a huge effect on the ground truth similarity. As the data was collected using v1.3, we would expect all of the similarities between the prompts and the corresponding images to be roughly similar, as indicated by the high similarity scores. Crucially, the *art* and *painting* datasets are extremely difficult to learn under a different model, which mostly likely points to the low quality of the dataset that resulted after our preprocessing.

However, if our data is salient enough, then our preferences can be learned across models, as evidenced by the *futuristic* dataset, which has high ground truth similarity between both choices of backbone. Hence, this reaffirms what we likely already knew that the performance of our results depends heavily on how good our data is. This signifies the potential of our method to be agnostic to the data collection process as well. This gives hope to the possibility of our method being able to apply in general for arbitrary and flexible preference learning as outlined in Figure 3.

4.3. Examples of Prompts Learned

Finally, we provide some examples of the prompts learned throughout training. We note that while we added in all the extra modifiers used in the dataset, it was difficult to limit our model to pick prompts only from those. On the other hand, it means that it was also nearly impossible for our model to happen upon the specific words that were used in the dataset. There is evidence that there is a secret language of combinations present in language models, so we assume that such learning is happening that allows our preference modules to be salient. This is another example of how brittle or different models can be from what we think of.

Dataset	Step	Prompt Tokens
Art	300	Political City
Art	300	Action Offline Customer Range Station
Art	300	Resource Services Tips Scope Overview Test Database Appearance Testing Country

Table 4. Examples of prompts learned throughout training. They are largely nonsensical for us as humans, but for the language models it can be bizarrely different.

5. Conclusion

We present a method for optimizing prompts for text-to-image diffusion models via Reinforcement Learning. Specifically, we frame a new task – given a particular user-submitted image, what tokens can be added onto a seed prompt in order to create a high level of semantic similarity between the diffusion generated image and the user submitted image. We construct several synthetic datasets, consisting of seed prompt, ground truth prompt and ground truth image triplets, in order to test our methodology. We present an extension to RL-Prompt and introduce a Stable Diffusion based reward function. We then test our method on a variety of datasets, and conduct multiple ablations. We show that our method can successfully learn optimal prompts that perform better than the baseline in certain settings.

Applications and Further Work There are two potential larger use cases for this task of reconstructing target images with diffusion models. Firstly, this method would allow for an automated mode of prompt discovery, which would help align the outputs of diffusion models with human preferences. Secondly, this could even be extended as a form of lossy image compression, where large memory intensive image datasets can be reduced to an equivalent set of text prompts. Then, instead of storing raw images, one would be able to store a much cheaper set of text prompts without incurring a large loss of information.

6. Contributions

Arnav and Tyler split up the work 50/50. Arnav handled collection of data, and setting up the policy model in the pipeline. Tyler handled integrating diffusion models and setting up the reward function. Arnav and Tyler ran experiments and wrote this report jointly.

References

Dash, A., Gamboa, J. C. B., Ahmed, S., Afzal, M. Z., and Liwicki, M. Tac-gan-text conditioned auxiliary classifier generative adversarial network. *arXiv preprint arXiv:1703.06412*, 2017.

Deng, M., Wang, J., Hsieh, C.-P., Wang, Y., Guo, H., Shu, T., Song, M., Xing, E. P., and Hu, Z. Rlprompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3369–3391. Association for Computational Linguistics, 2022.

Guo, H., Tan, B., Liu, Z., Xing, E., and Hu, Z. Text generation with efficient (soft) q-learning, 2022. URL https://openreview.net/forum?id=9TdCcMlmsLm.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pp. 1352–1361. PMLR, 2017.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. In *ICML 2022 Workshop on Knowledge Retrieval and Language Models*, 2022. URL https://openreview.net/forum?id=6p3AuaHAFiN.

Krea.ai. Open prompts. https://github.com/krea-ai/open-prompts, 2022.

Lee, K., Smith, L., and Abbeel, P. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. 2021.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Nye, M., Andreassen, A. J., Gur-Ari, G., Michalewski, H., Austin, J., Bieber, D., Dohan, D., Lewkowycz, A., Bosma, M., Luan, D., Sutton, C., and Odena, A. Show your work: Scratchpads for intermediate computation with language models, 2022. URL https://openreview.net/forum?id=iedYJm92o0a.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Gray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022a. URL https://openreview.net/forum?id=TG8KACxEON.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022b.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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. Learning transferable visual models from natural language supervision. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8821–8831. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/ramesh21a.html.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents, 2022. URL https://arxiv.org/abs/2204.06125.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. Generative adversarial text to image synthesis. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1060–1069, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/reed16.html.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10684–10695, June 2022.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C. W., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S. R., Crowson, K., Schmidt, L., Kaczmarczyk, R., and Jitsev, J. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=M3Y74vmsMcY.

Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., and Singh, S. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.

Snell, C. Alien dreams: An emerging art scene. 2021. URL https://ml.berkeley.edu/blog/posts/clip-art/.

Wallace, E., Feng, S., Kandpal, N., Gardner, M., and Singh, S. Universal adversarial triggers for attacking and analyzing nlp. *arXiv preprint arXiv:1908.07125*, 2019.

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., and Metaxas, D. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017.