# HERD: Fine-tuning Diffusion Models with Reinforcement Learning

Jasper Tan
jaspertan@utexas.edu

Haakon Mongstad
haakon@utexas.edu

The University of Texas at Austin

## Abstract

This project aims to explore the application of reinforcement learning (RL) techniques in the context of fine-tuning diffusion models. Diffusion models have shown great potential in various domains, such as image generation and text completion. However, their performance can be further improved by leveraging RL algorithms to optimize the model's behavior. In this work, we propose utilizing a hindsight experience replay buffer along with DDPO and evaluating prior approaches that combine RL with diffusion models to achieve better results in image synthesis. Leveraging the Transformer Reinforcement Learning library, we test various policy gradient reinforcement learning algorithms to evaluate each performance on an existing open-source text-to-image model, Stable Diffusion v1-5.

Our code is available on GitHub:

GitHub Repository

Our Presentation is available on YouTube:

YouTube Link

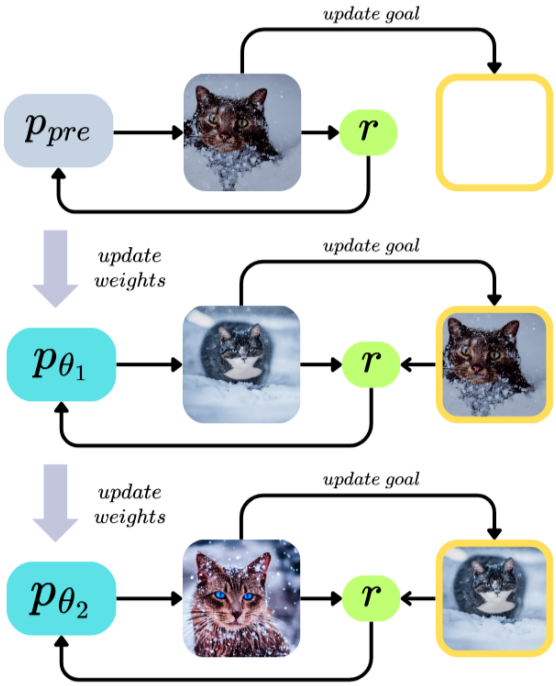



Figure 1: HERD: Fine-tuning Diffusion Model

## 1 Introduction

Recently image and video generation models have generated impressive results in converting text to various forms of multimedia. With the expansion of image generation, exploring improvements in this field utilizing RL methods has the potential to further align generated results with text prompts.

Our research focuses on leveraging reinforcement learning (RL) to enhance text-to-image diffusion models. Text-to-image generation has long been a goal in AI research. We're exploring RL, particularly policy gradient methods, to improve the fidelity of generated images. We're framing the problem as a Markov Decision Process, where each action represents a denoising step in the diffusion model. Our goal is to refine images based on textual prompts using policy gradient algorithms to adjust denoising steps to provide images aligned to a prompt.

In our experiments, we utilize the Transformer Reinforcement Learning Framework from Hugging Face [11] to evaluate Denoising Diffusion Policy Optimization (DDPO) [2] and Diffusion Policy Optimization with KL regularization (DPOK) [3], online

REINFORCE-based algorithms used to fine-tune text-to-image models. We generate a reward signal with pre-trained reward models, specifically ImageReward [13] and LAION-Aesthetics [10]. We propose a novel method to use Hindsight Experience Replay for Diffusion Models (HERD) with DDPO to achieve comparable results. In our empirical evaluations, we compare DDPO and implementations of DPOK (our baselines) to HERD to evaluate which algorithm performs best when aligning text to image generation.

Our contributions are as follows:

- We re-implement current online policy gradient methods, DPOK and DDPO, in fine-tuning text-to-image models with RL.
- We evaluate the RL algorithm's performance when implemented with Hugging Face's TRL Framework across different Reward Models.
- We study incorporating Hindsight Experience Replay for Diffusion Models (HERD) with DDPO to fine-tune a Stable Diffusion Model.

## 2 Related Work

In this section, we present the foundational principles of diffusion models and the reinforcement learning framework utilized in this paper, along with the main RL algorithms employed in our experimental framework.

### 2.1 Diffusion Models

Diffusion models [4] have garnered significant attention in recent years for their effectiveness in generative tasks across various modalities including images, videos, and robotics. Diffusion models work through iteratively denoising noise-corrupted versions of data, gradually refining their estimation of the underlying probability distribution. This, coupled with large language encoders, has provided a powerful framework for text-to-image synthesis.

Throughout the training process, a noised image $x_t = \sqrt{a_t}x + \sqrt{1-a_t}\epsilon$ is produced by adding Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ to a clean image $x$, with a decaying parameter $a_t \in [0, 1]$ such that $a_0 = 1$ and $a_T = 0$. The diffusion model $f_\theta$ is then trained to remove the noise $\epsilon$ and recover the original image $x$ by predicting the added noise. This is achieved by stochastically minimizing the objective $\frac{1}{N}\sum_i \mathbb{E}_{t,\epsilon}\mathcal{L}(x_i, t, \epsilon; f_\theta)$ , where

$$\mathcal{L}(x, t, \epsilon; f_\theta) = \|\epsilon - f_\theta(x_t, t)\|_2^2$$

### 2.2 Stable Diffusion

Stable diffusion is currently one of the largest and most popular text-to-image open-source diffusion models, with an expansive architecture comprising 890 million parameters [9].

While diffusion models excel in generating high-quality synthetic data, optimizing them directly in pixel spaces requires significant computational resources. Additionally, inference can be computationally expensive due to sequential evaluations. To address these challenges, Rombach et al. created Stable Diffusion [9] and proposed a novel approach: utilizing the latent space of pre-trained autoencoders. Through this process, they enabled efficient training and inference on limited computational resources while maintaining the quality and flexibility of diffusion models.

In practice, this involves encoding an input image $x$ into a latent representation $z = E(x)$ using an encoder $E$ and then reconstructing the image $x$ from the latent space using a decoder $D$. This reconstruction process is denoted as $\tilde{x} = D(z) = D(E(x))$. To incorporate conditional information $y$ from diverse modalities, such as textual prompts, the Stable Diffusion framework introduces a domain-specific encoder $\tau_\theta$ that maps $y$ to an intermediary representation. The model's training involves stochastic minimization of the objective:

$$\frac{1}{N}\sum_i E_{t,\epsilon}L(z_i, t, \epsilon; f_\theta)$$

Here, N represents the total number of samples and $f_0$ represents the model's parameters.

### 2.3 Reinforcement Learning for Diffusion Models

Reinforcement learning algorithms have proven to be extremely effective in leveraging human feedback to align models to human preferences, particularly in the domain of large language models. In the context of text-to-image generation, recent studies have explored the use of human preferences to enhance diffusion models' alignment and overall quality.

Human preferences are typically collected at scale through annotators' ratings of generated samples. Following this, a reward model is trained to learn scalar rewards for prompt-image pairs that match human preferences. These reward models are then used to provide the model with a signal that guides it toward optimizing its performance through iterative updates.

We build off of previous work that implements policy gradient methods to finetune diffusion models. Namely, DPOK [3] follows Lee et al. [7] to effectively align stable diffusion to reward models. They stabilized their reward with KL regularization and showed that, generally, reinforcement learning is superior to supervised learning for text-image alignment. DDPO [2] presents a similar method for aligning diffusion

models with reinforcement learning, but focuses on training on several prompts at once to generalize more to prompts outside of the dataset.

### 2.4 Hindsight Experience Replay

Hindsight Experience Replay (HER) presents a straightforward yet effective approach to enhancing reinforcement learning (RL) algorithms by leveraging replay buffers [1]. The core idea revolves around storing transitions not only with the original goal of an episode but also with alternative goals. This allows for the agent's actions to split from the environment dynamics during replay.

One important aspect of HER is the selection of additional goals for replay, which can significantly influence its effectiveness. In its simplest form, the algorithm replays each trajectory with the goal achieved in the final state of the episode. Experimentation, however, with different types of goals is essential to optimize performance. The algorithm's implicit curriculum dynamically adjusts the replayed goals, progressing from simpler to more challenging objectives over time. Unlike explicit curriculums, HER does not require control over the distribution of initial environment states, making it a flexible and adaptable approach to RL training. Furthermore, HER demonstrates remarkable performance in learning with sparse rewards, surpassing performance with shaped rewards in various experimental settings [1]. Overall, HER offers a promising avenue for enhancing RL algorithms' robustness and adaptability in tackling complex real-world problems with sparse feedback signals.

## 3 Method

We will discuss our implementation strategies to fine-tune the text-to-image diffusion model using reinforcement learning algorithms.

### 3.1 Problem Setting

In this setting, we define our objective as a Markov Decision Process (MDP). An MDP is represented as a tuple $(S, A, \rho_0, P, R)$, where $S$ is the state space, $A$ is the action space, $\rho_0$ is an initial state distribution, $P$ is the transition dynamics, and $R$ is the reward function. For our setting, the diffusion process can be mapped to an MDP framework in which at each time step $t$ a policy $\pi(a|s_t)$ observes a state $s_t \in a_t$ and takes an action $a_t \in A$. The environment then transitions to a next state $s_{t+1}$ $P(s_{t+1}|s_t, a_t)$ and then returns a reward $R(s_t, a_t)$. The aim of this policy is to maximize the total rewards it receives throughout the denoising process.

**State Space** In the context of a text-to-image diffusion model, the state space encompasses any given state the model can be in within the diffusion process. Each state $s_t$ at time step $t$ includes the latent representation of the current noisy image, the time step, and the text prompt provided for image generation.

**Action Space** The action space corresponds to the set of actions that the policy can take at each time step $t$ of the diffusion process. At each step, the model selects an action that modifies pixel values to denoise the current image in an effort to align closer to the desired output specified by the input text prompt.

**Reward** Reward models in this setting are typically pre-trained image evaluation models that score the generated image based on some alignment to human preferences.

### 3.2 Reward Model

To enhance our text-to-image fine-tuning process, we utilize a reward signal that assesses the alignment between the prompt and the image. After each sample batch of images is generated we feed the images into a specific reward model to see how the diffusion model should be updated. A reward model serves as a crucial component in guiding the training process. By incorporating the model's feedback into our optimization, we aim to enhance the quality and relevance of the generated images. This integration enables our diffusion model to better capture the semantic intent conveyed by the textual prompts, resulting in more coherent and contextually relevant image synthesis. We opt to evaluate two distinct reward models to gauge their respective impacts on image generation within each algorithm.

**ImageReward**. The ImageReward model, introduced by Xu et al. [13], was developed using real user prompts and human-annotated images to focus on assessing alignment, fidelity, and harmlessness. Leveraging cross-attention mechanisms, the model combines image and text features to generate alignment scores via a Multilayer Perceptron (MLP), with BLIP as its underlying architecture [8]. The model outputs a number that follows an approximate normal distribution, with a mean of 0 and a variance of 1, indicating the degree of alignment between the provided prompt and the resulting image.

**LAION-Aesthetics**. The LAION-Aesthetic model by Schuhmann et al. [10], evaluates the alignment between images and prompts. Developed using a dataset of synthetic images generated with AI models, LAION-Aesthetic utilizes CLIP image embeddings and incorporates a frozen CLIP ViT-L/14 with five additional MLP layers. These are are fine-tuned via regression loss. When provided with an image and corresponding text, the LAION-Aesthetic outputs a metric ranging from 0 to 1, indicating the alignment of the image with

the text.

### 3.3 Transformers Reinforcement Learning

As context for the proceeding sections, we use Transformers Reinforcement Learning (TRL) [11] to aid in our development. TRL is a framework built by Hugging Face to fine-tune transformer language and diffusion models using various methods that include popular reinforcement learning algorithms.

### 3.4 Denoising Diffusion Policy Optimization

The Denoising Diffusion Policy Optimization (DDPO) method introduces an algorithm that integrates reinforcement learning (RL) with diffusion-based generative models [2]. This method frames the generation process as a sequential decision-making problem, where each step in the diffusion sequence is optimized using RL techniques to meet specific downstream objectives like image quality.

DDPO's state at each timestep in the diffusion process, denoted by $s_t$, comprises the current image $x_t$, target text description $c$, and timestep $t$. The action $a_t$ corresponds to the generation step transitioning the image from $x_t$ to $x_{t-1}$, guided by the model's parameters $\theta$. The entire sequence runs backward from a purely noisy state $x_T$ towards the clean image $x_0$.

The primary objective in DDPO is to maximize a reward function $R(s_t, a_t)$ that quantifies the alignment of the generated image with the text description at the final timestep. The RL objective, denoted as $J(\theta)$, is formulated as the expected cumulative reward over a trajectory $\tau$ generated according to the policy $\pi_\theta$:

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[ \sum_{t=0}^{T} R(s_t, a_t) \right]$$

The policy $\pi_\theta(a_t|s_t)$, defined by the parameters $\theta$, dictates how the image adjusts at each step. The optimization of $J(\theta)$ is achieved through a policy gradient method, REINFORCE [12]. The gradient of the expected reward is computed as:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[ \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) R(s_t, a_t) \right]$$

This expression allows the diffusion model's parameters to adjust such that the generated images are more aligned with the given text descriptions through the training iterations [2].

In the context of text-to-image generation, DDPO adapts pre-trained diffusion models to handle complex image generation tasks directed by textual prompts. By employing DDPO, the model is fine-tuned to optimize the generation process specifically towards creating images that meet defined qualities inferred from textual descriptions.

To simulate the DDPO algorithm effectively, we utilized the TRL framework [11], as it provided a robust environment for the seamless execution and evaluation of DDPO. As a result, we decided to utilize DDPO as a baseline comparison for other reinforcement learning algorithms. It was a suitable benchmark as it is a well-established method known for its capability to optimize image generation processes.

### 3.5 Diffusion Policy Optimization with KL regularization

Diffusion Policy Optimization with KL Regularization (DPOK) [3] introduced a method to fine-tune stable diffusion with online reinforcement learning. In their algorithm, they use a policy gradient approach and particularly focus on the integration of Kullback Leibler (KL) divergence for regularization. This method follows REINFORCE [12] to compute the gradients of a loss function that combines the output from the reward model and a KL divergence term and use them to update the diffusion model's weights. The equation to calculate the gradients is defined as

$$\mathbb{E}_{p(z)} \mathbb{E}_{p_\theta(x_{0:T}|z)} \left[ a + b \right]$$

where,

$$a = -\alpha r \sum_{t=1}^{T} \nabla_\theta \log p_\theta(x_{t-1}|x_t, z)$$

and

$$b = \beta \sum_{t=1}^{T} \nabla_\theta \mathrm{KL} \left( p_\theta(x_{t-1}|x_t, z) \| p_{\mathrm{pre}}(x_{t-1}|x_t, z) \right)$$

The authors presented the use of KL regularization to prevent the model from deviating significantly from the pre-trained behavior of the model.

To aid in our experimentation, we implement DPOK using TRL. As DPOK is not available in the TRL library, we extend the preexisting DDPO class with added modifications. Namely, we add the KL divergence term which is added to the calculation of the loss. We also introduce a model that learns the value function estimate given the state of the diffusion model.

### 3.6 HER for Diffusion Models

In this section, we introduce a novel method to utilize Hindsight Experience Replay for Diffusion Models (HERD).

It is important to acknowledge that the original HER introduced by Andrychowicz et al. is not suitable for diffusion models' reward setting [1]. HER is typically effective in environments where rewards are sparse and binary; however, the reward model in this setting often involves a more continuous and nuanced evaluation based on how well the generated image matches the prompt.

To adapt HER for fine-tuning text-to-image diffusion models, we propose Hindsight Experience Replay for Diffusion models (HERD). This algorithm modifies the traditional application to accommodate the nuanced application of our task. In our setting, the goal of each iteration is to align the image with specific details of the text prompt. While the reward for this goal is neither sparse nor binary, we refine the goal to adjust the model's parameters on individual portions of the prompt. After each epoch iteration, a randomly selected batch of samples and their associated rewards of size $H$ is stored in the HER buffer for the next iteration. Each sample's prompt is then modified by setting one randomly selected sentence from the original prompt as the sample's new prompt. These new prompts are saved to match the current image, and a new reward is computed based on the modified prompt and the original image. The concept of modifying the prompt, essentially changing the model's goal in HER[1], allows the model to extract more reward signals from previous samples under slightly varied prompts that capture finer details.

The motivation behind HERD stems from the desire to enhance the model's sensitivity and responsiveness to intricacies within complex prompts. In traditional text-to-image tasks, a model may struggle to capture finer details that are specified in long, verbose prompts. By dynamically adjusting the goal of our model associated with previously generated images to include subsections of the original prompt, we encourage the model to focus not just on broad accuracy but also on details within each sentence or subsection. This will help incrementally train the diffusion model to generate images that are more contextually aware of its input text.

In HER implementation, we implemented the TRL framework [11] utilized for the DDPO algorithm [2] with an added memory bank to incorporate HER as seen in Algorithm 1. After each training iteration, HER was adjusted to fit the batching size, and prompt embeddings were updated to reflect prompt changes. Initially, the training sample for the first interaction did not include replay, as the buffer was initialized empty. Subsequent runs incorporated previous trajectories to assist the model in aligning to the prompt.

**Algorithm 1** HERD Algorithm

**Input:**
$\mathcal{R}$: Reward model
$\mathcal{P}$: Text to Image Diffusion model
$C$: Initial target prompt
$M$: Batch size
$E$: Inner epoch size
$T$: Total number of iterations
$H$: Total number of Replays

Initialize Sample Buffer $b_s$ of size $H$
Initialize Reward Buffer $b_r$ of size $H$
**for** $t = 0$ to $T$ **do**
  **for** each $m$ in $M$ **do**
    $s_{t,m} \leftarrow \mathcal{P}(C)$ ▷ Generate Trajectories
    $r_{t,m} \leftarrow \mathcal{R}(s_t)$ ▷ Compute Rewards
  **end for**
  $s_t \leftarrow b_s$ ▷ Append buffer samples
  $r_t \leftarrow b_r$ ▷ Append buffer rewards
  $r_t \leftarrow \text{Normalize}(r_t)$
  **for** $i = 0$ to $E$ **do**
    $s_t \leftarrow \text{Shuffle}(s_t)$
    $\text{Train}(p, s_t, r_t)$
  **end for**

  $b_s \leftarrow []$ ▷ Empty Previous Samples
  $b_r \leftarrow []$ ▷ Empty Previous Rewards
  **for** $i = 0$ to $H$ **do**
    $s_r \leftarrow \text{Random}(s_t)$ ▷ Get Random Trajectory
    $b_{s,i} \leftarrow \text{UpdatePrompt}(C)$
    $b_{r,i} \leftarrow \mathcal{R}(s_r)$ ▷ Re-compute Rewards
  **end for**
**end for**

## 4 Experiments

In this section, we describe the set of experiments designed to test the efficacy of our various algorithms.

### 4.1 Experiment Setup

We use Stable Diffusion [9] as our base generative model, pre-trained on extensive image-text datasets. To efficiently fine-tune Stable Diffusion, we implemented algorithms on the Transformer Reinforcement Library (TRL) [11] framework, which incorporates Low-Rank Adaptation (LoRA) [5], enabling training with reduced computational resources. LoRA freezes Stable Diffusion's pre-trained parameters, facilitating efficient updates to low-rank layers. For evaluating rewards, we utilize both ImageReward and LAION-Aesthetic to compare rewards received by various algorithms. We also wanted to see what kind of impact each reward model had on the final image output. While LAION-Aesthetic was originally part of the TRL library, we also were interested in implement-

ing ImageReward due to its superior preference accuracy compared to CLIP Score, LAION-Aesthetic, and BLIP Score [13].

**Standard Prompt**. The prompt we selected needed to feature a variety of objects while including specific descriptions to enable HERD to focus on particular aspects of the prompt. The chosen prompt reads as follows:

> "A black cat and golden retriever dog. A hot ocean-side beach. Dramatic atmosphere, centered, rule of thirds, professional photo."

This prompt encompasses specific objects, namely a cat and a dog, along with setting and image descriptions to aid our algorithms in visualizing a complex image. With this detailed description, our objective was to establish a baseline using pre-trained Stable Diffusion and then assess how fine-tuning could enhance the generated image.

**Hyperparameters**. In our configuration, we primarily focused on testing three main parameters: sample batch sizes, train batch sizes, and the number of epochs. Due to hardware constraints, we maintained consistent batch sizes throughout our experiments. We set the number of epochs to 100, as our observations indicated that DDPO typically converged within this timeframe.

We conducted a limited parameter search through trial and error to identify settings that could optimize our output while minimizing computational resources. In total, we ran approximately 200 trials, ranging from runs lasting up to 6 hours for comprehensive evaluations to shorter sessions lasting less than 30 minutes for debugging purposes.

### 4.2 Comparison of Fine-tuning Algorithms

We evaluate our fine-tuning algorithms by generating images based on a standardized multi-sentence prompt containing descriptions of various objects. To maintain consistency, we opted for a batch size of 12 samples and 3 training batches, ensuring efficient learning and image generation across all models. The average rewards of each batch serve as a qualitative measure of the performance of our algorithms relative to each other.

#### 4.2.1 Configurations

In Figure 2, we compare each algorithm using LAION-Aesthetic and ImageReward scoring models. The graphs are smoothed with a scale of 25 using a running average smoothing operator to illustrate the general trend of each algorithm. The faded-out results on the graphs represent the raw reward data collected. Our HERD model was tested with two different configurations: HERD with a buffer size of 1 batch and HERD-b2 with a buffer size of 2 batches. Although each algorithm underwent the same number of training epochs, the number of steps per epoch varied due to HERD and HERD-b2 sampling previous trajectories for retraining We deliberately sampled a low number of batches when considering previous trajectories to minimize computational resource requirements.

#### 4.2.2 Results

In both the LAION-Aesthetic and ImageReward settings, we observe that DDPO achieves a higher reward mean and converges in fewer steps. This indicates that while HERD theoretically offers novel improvements, DDPO achieves comparable results with fewer fine-tuning steps for the diffusion model. Specifically in Figure 2a, DDPO demonstrates faster convergence to a higher point, displaying its effectiveness in this task. When compared to our DDPO baseline the novel algorithms, HERD-b2 and HERD performed similarly to DDPO in Figure 2b; however, over more steps the HERD algorithm reward oscillated with a decreased reward mean.

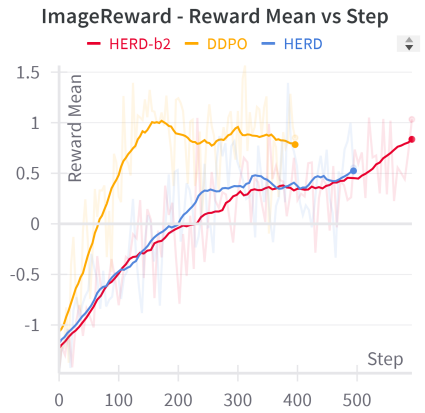
Although DPOK showed some improvement over the pre-trained Stable Diffusion model, its results were not included due to lower performance compared to the evaluation by Fan et al. [3].

### 4.3 Reward Model Differences

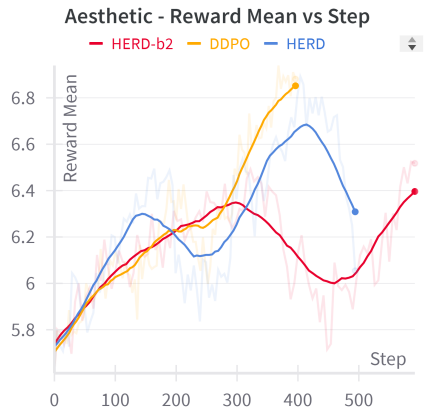
Throughout our experimentation, we compared the performance of our algorithms across two different reward models. These reward models had very different effects on the diffusion model after fine-tuning due to the nature of their reward signals.

**Image Reward.** Image reward receives the generated image and target prompt as inputs, and produces a score that evaluates the image based on its alignment to the prompt. This implies that diffusion models that were fine-tuned on Image Reward generate images that follow the input text prompt much closer than the pre-trained model.

To generate the images shown in Figure 3, we used the Standardized Prompt defined in Section 4.1. Figure 3 indeed shows that fine-tuning the diffusion model on Image Reward results in a model that better captures the information provided in the prompt. Compared to the image generated from the pre-trained Stable Diffusion model, the generated images after fine-tuning all begin to display the details of the prompt far more accurately. Namely, models fine-tuned on Image Reward excel in generating the correct type of objects, correct counts of each object, and correct colors.




(a) ImageReward Average Reward




(b) Aesthetic Average Reward

Figure 2: Comparison of Average reward achieved through Aesthetic and ImageReward Rewards Models over a 100 epoch period.

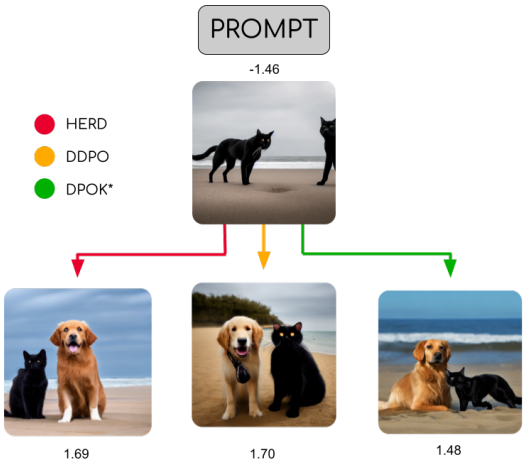



Figure 3: Example Image Generation After Fine-tuning on ImageReward

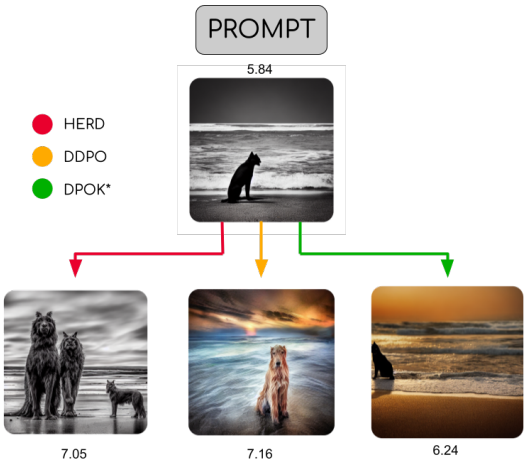



Figure 4: Example Image Generation After Fine-tuning on LAION-Aesthetic

We also visualize that HERD achieves comparable performance to DDPO, generating images of similar quality. DPOK, while still improving from the base model, has an asterisk as the implementation in the TRL framework appears to be behaving suboptimally in comparison to the results from their paper [3].

**LAION-Aesthetic** LAION-Aesthetic signals, on the other hand, focus on how well the image aligns with a human's perception of aesthetic quality. As a result, the generated images of a model fine-tuned on this reward model tend to capture less of the textual context from the prompt, but more on textures and colors within the generated image.

Based on the same Standardized Prompt, the generated images from Figure 4 depict how the fine-tuned models focus on generating images that match a preferred style rather than the exact content of the prompt.

This behavior is not ideal for the reward models used in HERD. This is because the motivation behind HERD is to break up a long prompt and follow the textual context within subsections of the prompt. Without a reward signal to evaluate the model's ability to match the content of the prompt, the model will not be fine-tuned to match the prompt.

### 4.4 Evaluate on Complex Prompt

After running an initial experiment with all 3 algorithms we decided to take a further step in testing

to explore whether more complex prompts with detailed descriptors could benefit from HERD. The chosen prompt reads:

> "Deep in an ancient forest, sunlight filters through towering trees to illuminate a tranquil clearing. Within this serene glade, a crystal-clear stream meanders lazily through beds of colorful wildflowers. Moss-covered rocks line the stream, where playful woodland creatures sip from the cool waters. Glistening fur and dappled sunlight surround the creatures as they frolic amongst the flowers. Beyond the clearing, the forest stretches away, hinting at hidden mysteries within its depths."

This prompt contains multiple objects, posing a challenge for both the pre-trained Stable Diffusion model and a fine-tuned version with the DDPO fine-tuning method. Our intuition was that HERD might perform better with a more detailed prompt, as it can segment the prompt into smaller, more manageable tasks for the diffusion model to fine-tune.

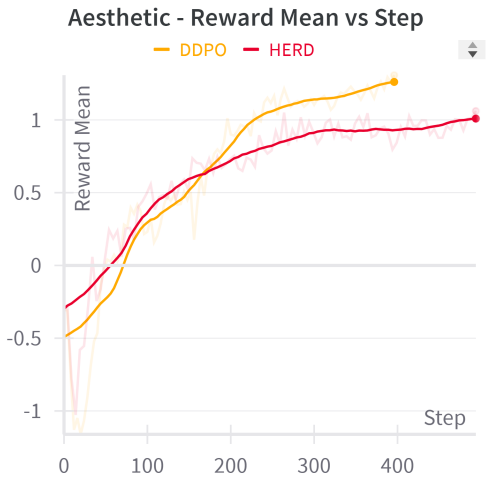



Figure 5: ImageReward Average Reward with Long Prompt

In Figure 5, we see that HERD initially outperforms DDPO by breaking down the prompt into smaller, comprehensible segments for the model to train on. DDPO struggles initially with the larger, more complex prompt; however, over many steps, it achieves a higher score. This displays that HERD is suitable for longer prompts with fewer iterations.

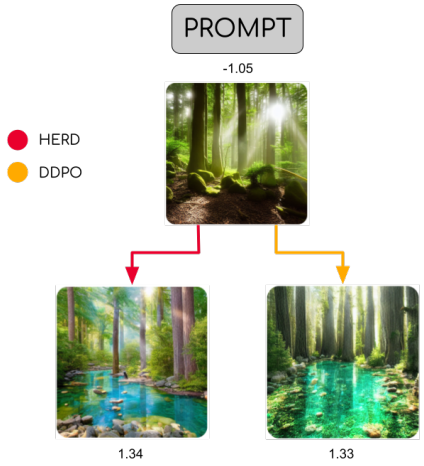



Figure 6: Example Image Generation After Fine-tuning on ImageReward with a Complex Prompt

When looking through the generated images manually, as displayed in Figure 6, we thought that HERD produced less grainy images compared to DDPO; however, the ImageReward model may prioritize other factors more, leading to an increase in the average reward of the DDPO algorithm to increase.

## 5 Discussion

In this study, we implemented established algorithms to fine-tune Stable Diffusion with reinforcement learning. Additionally, we introduced Hindsight Experience Replay with Diffusion Models (HERD) to incorporate previous samples, aiding in fine-tuning specific prompt details. While implementing the DPOK algorithm, we encountered issues, possibly due to mismatches with the TRL framework, resulting in lower-than-expected rewards. However, utilizing DDPO yielded the expected outcomes observed in hugging face, serving as a benchmark for our proposed method. Our findings demonstrate that HERD has the potential to be effective in preserving prompt details relevant to the image, especially when dealing with lengthy and verbose prompts.

### 5.1 Limitations

The primary constraint we faced was hardware limitations. Despite implementing LoRA [5], many of our laptop GPUs struggled to run inference on a diffusion model. To overcome this obstacle, we explored using the Texas Advanced Computing Center (TACC) to leverage the server's GPUs for fine-tuning tasks. Although we successfully utilized TACC for a run, we observed lengthy processing times for one of our programs and potential challenges with debugging and

image generation. Consequently, we opted to utilize personal desktops, enabling direct interfacing with the runs and immediate code compilation to facilitate project development. However, our desktop GPUs were limited to 12 GB of VRAM, necessitating a reduction in batch and sample sizes for model training. Despite these adjustments, hardware constraints prevented us from hyperparameter tuning and exploring a broader range of diverse and complex prompts as desired.

### 5.2 Future Work

Our novel proposal of HERD opens up numerous avenues for enhancing the fine-tuning process of text-to-image diffusion models using reinforcement learning. Since HERD leverages previous trajectories within a black-boxed training algorithm, it can readily applied to various algorithms.

**Bootstrapping Replays**. While our current HERD implementation initializes an empty buffer in the beginning, there is potential to bootstrap trajectories to further enhance diffusion model fine-tuning. By employing Evolutionary Reinforcement Learning [6], simpler goals can be created for text-to-image diffusion models, and trajectories from easier tasks can be saved for reuse when updating parameters for more challenging tasks. However, this approach may require additional iterations for the model to converge with each incrementally difficult task.

**Extended Replays**. Exploring extended replays is another promising direction. Unlike our current implementation, which clears the HERD buffer after each iteration, storing either the complete history or an extended history of trajectories could potentially enhance the diffusion model. We, however, opted against this approach initially due to concerns that early experiences might lead the model to learn inaccurate trajectories.

**Max Reward Samples**. Another potential route worth exploring to enhance the fine-tuning task involves selecting samples with the best rewards and utilizing those to help compute adjustments to the model's parameters. By taking the samples with the max reward, the algorithm would be taking the greedy choice and may lack exploration. In addition, this greedy choice would not be effective if all samples generated had consistently higher rewards than the previous rewards.

**HERD with DPOK**. Although HERD performed well with DDPO we decided not to combine it with DPOK due to suboptimal performance within the TRL framework. Exploring the applications of HERD within the DPOK framework developed by Google may yield more favorable outcomes [3].

## 6 Acknowledgements

We'd like to thank Amy Zhang and Peter Stone for their exceptional Reinforcement Learning: Theory and Practice Class and invaluable guidance on approaching this problem. We also appreciate the support and assistance of the teaching assistants, whose feedback and general assistance were instrumental in helping us achieve these results.

## References

[1] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay, 2018.

[2] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning, 2024.

[3] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models, 2023.

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[5] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[6] Shauharda Khadka and Kagan Tumer. Evolution-guided policy gradient in reinforcement learning, 2018.

[7] Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback, 2023.

[8] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022.

[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[10] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta,

Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022.

[11] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`, 2020.

[12] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, May 1992.

[13] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation, 2023.