VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

REPORT
SPECIALIZED PROJECT

IMPROVING TEXT-TO-IMAGE MODELS WITH ARTIFICIAL INTELLIGENCE FEEDBACK

Major: Computer Science

Thesis Committee: **Specialized Project 4**
Supervisors: **Nguyễn Quang Đức**
Nguyễn Đức Dũng
Member Secretary: **Trần Huy**
Student: **Võ Mạnh Quyền** - **2014318**

Ho Chi Minh City, December 2023

Date:

Mr. Nguyen Quang Duc (Thesis Instructor)
Bachelor of Engineering
Faculty of Computer Science and Engineering

Date:

Dr. Nguyen Duc Dung (Thesis Instructor)
Computer Science and Engineering PhD
Faculty of Computer Science and Engineering

Declaration of Authenticity

I declare that this thesis is my own work, conducted under the supervision and guidance of Mr. Nguyen Quang Duc and Dr. Nguyen Duc Dung. The result of my work is legitimate and has not been published in any forms prior to this. All materials used within this researched are collected myself by various sources and are appropriately listed in the references section.

In addition, within this research, I also used the results of several other authors and organizations. They have all been aptly referenced.

In any case of plagiarism, I stand by my actions and will be responsible for it. Ho Chi Minh City University of Technology - Vietnam National University HCMC therefore are not responsible for any copyright infringements conducted within my work.

Ho Chi Minh City, December, 2023
Author
Võ Mạnh Quyền

Acknowledgement

First of all, I sincerely thank the lecturer of the Faculty of Computer Science and Engineering in particular and the Ho Chi Minh City University of Technology in general for enthusiastically teaching me lot of professional knowledge, as well as valuable skills and experiences to lay the foundation for me to complete my thesis.

Especially I would like to express my deep and sincere gratitude to my supervisors, Mr. Nguyen Quang Duc and Dr. Nguyen Duc Dung, who directly guided me, created the most favorable conditions for me throughout the research process, helping me complete the project in the best way. Once again, I sincerely thank my two supervisors very much.

Through the research process, I have learned a lot of new knowledge and skills, as well as developed the ability to self-study and solve problems. Even so, during the process of implementing the thesis, due to limited theoretical and practical experience, it is inevitable to have mistakes, so I really hope to receive feedback from you to complete my thesis.

I sincerely thank you!

Abstract

Deep learning models for generating images from text have shown impressive results in text-to-image synthesis. However, current text-to-image models often generate images that do not match the text very well. For the reason, in this work, I propose a method to improve these models with artificial intelligence feedback, comprising two stages. First, I train a model that predict human preference using human feedback dataset. Then, I fine-tune the text-to-image model to maximize predicted feedback, making images better match the text. Although the result of experiment is not as good as expected, it helped me approach and see the potential of applying reinforcement learning algorithms to improve text-to-image models. In the future, I will try to improve my algorithm so that fine-tuned model generates better images that are aligned with text prompts.

Contents

List of Figures

List of Tables

I Introduction

Recent years have witnessed a rapid growth of deep generative models, with text-to-image models gaining significant attention from the public. Generative models have increasingly impacted relative tasks ranging from image revision and object detection in computer vision to interior design and idea illustration in more general fields. However, despite the advances in text-to-image (T2I) generation models [1]–[5], major challenges remain in domains where large-scale text-to-image models are challenging to generate images that meticulously conform to users' intentions [6]–[10]. Current models often fail to compose multiple objects [7], [8], [10], bind attributes to the wrong objects [7], and struggle to generate visual text [9].

Figure 1: *Images generated via DALL · E 2*

In language modeling, *learning from human feedback* has emerged as a powerful solution for aligning model behavior with human intent [11]–[16]. Such methods first learn a reward function intended to reflect what humans care about in the task, using human feedback on model outputs. The language model is then optimized using the learned reward function by a *reinforcement learning (RL)* algorithm, such as proximal policy optimization (PPO [17]). This RL with human feedback (RLHF) framework has successfully aligned large-scale language models (e.g., GPT-3 [18]) with complex human quality assessments.

In vision-language modeling, *learning from human feedback* has proven to be an effective means to overcome limitations of text-to-image models [6], [19]–[23]. Lee et al. [6] demonstrate that certain properties, such as generating objects with specific colors, counts, and backgrounds, can be improved by learning a reward function from human feedback, followed by fine-tuning the text-to-image model using supervised learning. They show that simple supervised fine-tuning based on reward-weighted loss can improve the reward scores, leading to better image-text alignment. However, supervised fine-tuning often induces a deterioration in image quality (e.g., over-saturated or non-photorealistic images). This is likely due to the model being fine-tuned on a fixed dataset that is generated by a pre-trained model.

Motivated by these successes, I want to experiment a simple fine-tuning method for aligning text-to-image models using artificial intelligence (AI) feedback. In this work, I using reinforcement learning (RL) for fine-tuning *stable diffusion* model [3] using ImageRewardDB [21] with 137k pairs of expert comparisons.

II Related Works

Text-to-image generative models: Text-to-image generative models have long been an active research area. Mansimov et al. [24] show that Deep Recurrent Attention Writer (DRAW) [25] can be conditioned on captions to generate novel scene compositions. Generative Adversarial Networks (GANs) improve image fidelity by training a discriminator to provide supervision for the generative model. DALL · E [2] firstly achieves open-domain text-to-image synthesis with the help of massive image-text pairs.

Diffusion models formulate the generative process as the inverse of the diffusion process [26], which was improved by Song and Ermon [27] and Ho et al. [28]. Dhariwal et al. firstly show the superiority of diffusion models over GANs on image generation [29]. Several following works, including DALL · E 2 [1], GLIDE [30], Imagen [5], ERNIEViLG [31], [32], Stable Diffusion [3], bring the magic of text-to-image

generation to the public attention. Among these models, Stable Diffusion is an open-source model with an active user community.

Learning from human feedback: There is often a gap between generative models' pre-training objectives and human intent. Thus human assessments of learned model outcomes have been used to guide learning on a variety of tasks, ranging from learning behaviors [33] to language modeling [12], [15], [16], [34]. Recent work has also applied such methods to improve the alignment of text-to-image models. Human preferences are typically gathered at scale by asking annotators to compare generations, and a reward model is trained (for example, by fine-tuning a vision-language model such as CLIP [35] or BLIP [36]) to produce scalar rewards well-aligned with the human data [19], [21]. The reward model is used to improve text-to-image model quality by fine-tuning a pre-trained generative model [6], [21], [22].

Improving general text-to-image alignment: I roughly categorize the alignment methods for improving T2I alignment into two classes depending on if they involve training. For training-involved methods, several works use Reinforcement Learning from Human Feedback (RLHF) based on human rankings to maximize a reward and improve faithful generation [6], [22], [37], which is the method I am aiming to. In a similar vein, Picka-Pic is a dataset of prompts and preferences that is used to train a CLIP-based scoring function [19]. StyleDrop trains adapters to synthesize of images that follow a specific style [38], and T2I-Adapter trains adapters to improve the control for the color and structure of the generation results [39]. DreamBooth and HyperDreamBooth improve personalized generation [40], [41], and they have inspired more efficient methods such as SVDiff [42]. Being orthogonal to training-involved methods, there is a body of work on training-free methods that make inference time adjustments to the model to improve alignment, such such as SynGen and StructuralDiffusion [7], [43]–[45].

Human feedback datasets: There are multiple datasets of human feedback on text-to-image generation, including AGIQA-1K [46], Human Preference Dataset (HPD) [20], ImageReward [21], Pick-a-Pic [19], AGIQA-3K [39], and Human Preference Dataset v2 (HPD v2) [47]. In this work, I use the ImageReward datasets that selects prompts and images from the DiffusionDB dataset [48], and collects for them image preference ratings via crowd workers.

III Basic knowledge

1 Stable Diffusion

In this section, I will describe basic knowledge for text-to-image generation of diffusion models.

1.1 Diffusion models

I consider the use of *denoising diffusion probabilistic models (DDPMs)* as stochastic Markov chains with Gaussian noises [28] for image generation and draw notation and problem formulation from [28]. Let q_0 be the data distribution, i.e., $x_0 \sim q_0(x_0), x_0 \in \mathbb{R}^n$. A DDPM approximates q_0 with a parameterized model of the form $p_\theta(x_0) = \int p_\theta(x_{0:T})\, dx_{1:T}$, where $p_\theta(x_{0:T}) = p_T(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)$ and the reverse process is a Markov chain with the following dynamics:

$$p_T(x_T) = \mathcal{N}(0, I), \quad p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(\mu_\theta(x_t, t), \Sigma_t) \tag{1}$$

A unique characteristic of DDPMs is the exploitation of an approximate posterior $q(x_{1:T} \mid x_0)$, known as the forward or diffusion process, which itself is a Markov chain that adds Gaussian noise to the data according to a variance schedule $\beta_1, \ldots, \beta_T$:

$$q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1}), \quad q(x_t \mid x_{t-1}) = \mathcal{N}\left(\sqrt{1-\beta_t} x_{t-1}, \beta_t I\right) \tag{2}$$

Let $\alpha_t = 1 - \beta_t, \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, and $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$. Ho et al. [28] adopt the parameterization $\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$.

Training a DDPM is performed by optimizing a variational bound on the negative log-likelihood $\mathbb{E}[-\log p_\theta(x_0)]$, which is equivalent to optimizing:

$$\mathbb{E}_q\left[\sum_{t=1}^{T} \mathrm{KL}\left(q(x_{t-1} \mid x_t, x_0) \,\|\, p_\theta(x_{t-1} \mid x_t)\right)\right] \tag{3}$$

Note that the variance sequence $(\beta_t)_{t=1}^T \in (0,1)^T$ is chosen such that $\bar{\alpha}_T \approx 0$, and thus, $q(x_T \mid x_0) \approx \mathcal{N}(0, I)$. The covariance matrix Σ_t in (1) is often set to either $\beta_t I$ or $\tilde{\beta}_t I$, which is not trainable. Unlike the original DDPM, I use a latent diffusion model [3], so x_t's are latents.

1.2 Text-to-image diffusion models

Diffusion models are especially well-suited to conditional data generation, as required by text-to-image models: one can plug in a classifier as guidance function [29], or can directly train the diffusion model's conditional distribution with classifier-free guidance [49].

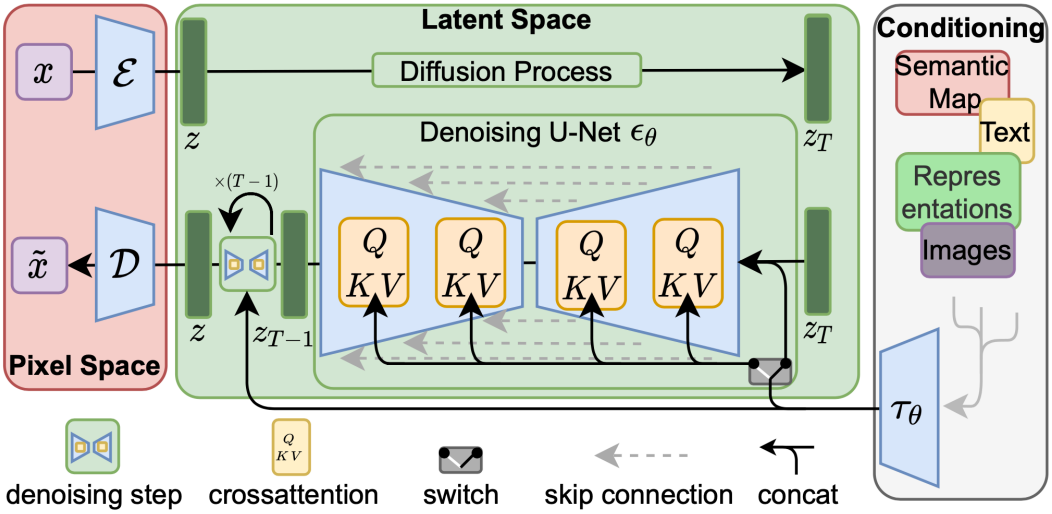

Figure 2: *The architecture of latent diffusion model.*
(Image source: [3])

Given text prompt $z \sim p(z)$, let $q(x_0 \mid z)$ be the data distribution conditioned on z. This induces a joint distribution $p(x_0, z)$. During training, the same noising process q is used regardless of input z, and both the unconditional denoising model $\epsilon_\theta(x_t, t)$ and a conditional denoising model $\epsilon_\theta(x_t, t, z)$ are learned. For data sampling, let $\bar{\epsilon}_\theta = w\epsilon_\theta(x_t, t, z) + (1-w)\epsilon_\theta(x_t, t)$, where $w \geqslant 1$ is the guidance scale. At test time, given a text prompt z, the model generates conditional data according to $p_\theta(x_0 \mid z)$.

In this project, I use Stable Diffusion v1.5 [3] as baseline generative model.

2 Reinforcement Learning from Human Feedback

Reinforcement learning from human feedback (RLHF) [11] is a machine learning approach that combines reinforcement learning techniques, such as rewards and comparisons, with human guidance to train an artificial intelligence (AI) agent to align with human preference.

RLHF training is done in three phases:

- **Initial phase:** The first phase involves selecting an existing model as the main model to determine and label correct behavior. Using a pre-trained model is a timesaver due to the amount of data required for training.
- **Human feedback:** After training the initial model, human testers provide input on performance. Human trainers provide a quality or accuracy score to various model-generated outputs. The system then evaluates its performance based on human feedback to create rewards for reinforcement learning.
- **Reinforcement learning:** The reward model is fine-tuned with outputs from the main model and receives a quality score from testers. The main model uses this feedback to improve its performance on future tasks.

RLHF is an iterative process because collecting human feedback and refining the model with reinforcement learning is repeated for continuous improvement.

However, scaling the process to train bigger, more sophisticated models, because this process depends on human feedback, it can be very time-consuming and resource-intensive. So another approach is that I can train a reward function based on a large amount of human feedback to predict automatically human preference.

3 CLIP

CLIP (Contrastive Language-Image Pre-Training) [35] is a neural network trained on a variety of (image, text) pairs to learn representations for images and text in a joint embedding space. Images and their captions are close together in this space, while unrelated images and captions are further apart.

CLIP jointly trains a text encoder and an image feature extractor over the pretraining task that predicts which caption goes with which image.

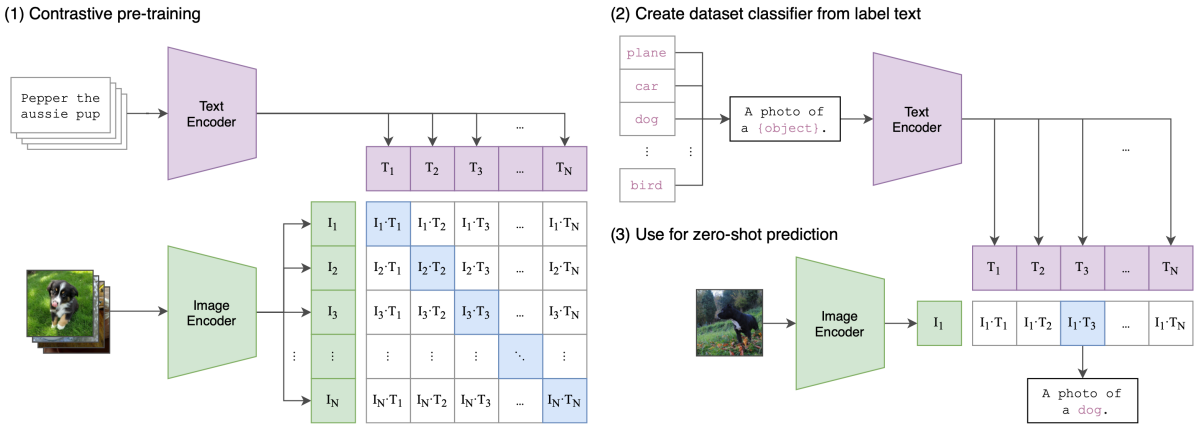

Figure 3: *Illustration of CLIP contrastive pre-training over text-image pairs.* (Image source: [35])

Given a batch of N (image, text) pairs, CLIP computes the dense cosine similarity matrix between all $N \times N$ possible (image, text) candidates within this batch. The text and image encoders are jointly trained to maximize the similarity between N correct pairs of (image, text) associations while minimizing the similarity for $N(N-1)$ incorrect pairs via a symmetric cross entropy loss over the dense matrix.

Compared to other methods for learning good visual representation, what makes CLIP really special is "the appreciation of using natural language as a training signal". It does demand access to supervised dataset in which we know which text matches which image. It is trained on 400 million (text, image) pairs, collected from the Internet. The query list contains all the words occurring at least 100 times in the English version of Wikipedia.

CLIP produces good visual representation that can meaningfully transfer to many computer vision benchmark datasets, achieving results competitive with supervised baseline. This allows the model to perform tasks requiring computer vision and natural language understanding capabilities.

4 LoRA Technique

LoRA (Low-Rank Adaptation of Large Language Models) [50] is a popular and lightweight training technique that significantly reduces the number of trainable parameters. It works by inserting a smaller number of new weights into the model and only these are trained while freezing the original weights.

A neural network contains many dense layers which perform matrix multiplication. The weight matrices in these layers typically have full-rank. For a pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$, we constrain its update by representing the latter with a low-rank decomposition $W_0 + \Delta W = W_0 + BA$, where $B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times k}$, and the rank $r \ll \min(d, k)$. During training, LoRA does not directly train the parameters in ΔW but trains the parameters in A and B. Therefore, the number of trained parameters will be much lower. There are only $(d + k) \times r$ parameters for training in A and B instead of $d \times k$ parameters for training in ΔW. This technique is illustrated in Figure 4. Where A is initialized with a random Gaussian distribution and B is initialized with a zero matrix.

Training with LoRA possesses several distinct advantages:

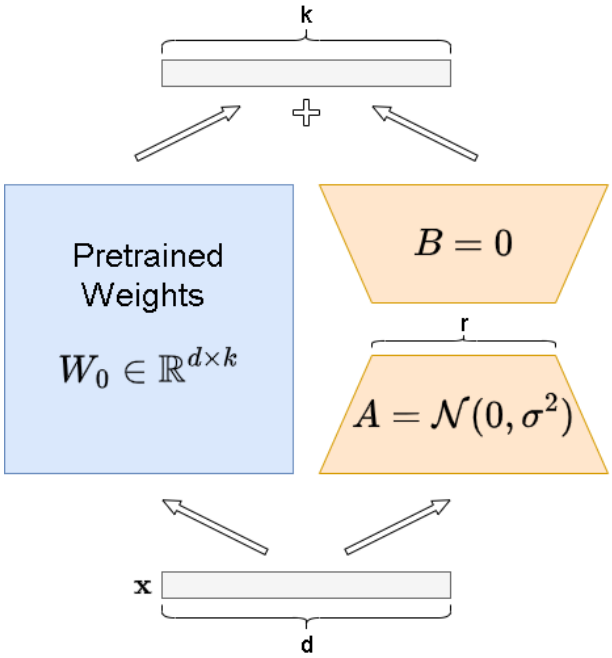

Figure 4: *Illustration of LoRA technique*

- **Computational Efficiency:** Fine-tuning the entire model can be computationally expensive, especially when dealing with huge models with millions or billions of parameters. LoRA reduces the computational cost by working with low-rank matrices, making it more feasible for resource-constrained environments. It focuses on the optimal use of computational resources, such as CPU processing power, GPU capabilities, and memory by decreasing the count of parameters that should be adjusted or trained.
- **Knowledge Preservation:** LoRA retains the general knowledge captured during pre-training, which is essential for applications where the model's broad understanding is beneficial. Knowledge preservation can be a key motivation for using LoRA. Instead of completely retraining or fine-tuning a model from scratch, which might result in a loss of valuable pre-trained knowledge, LoRA allows you to adapt the model while minimizing the loss of this knowledge.
- **Reduced Catastrophic Forgetting:** When fine-tuning a pre-trained model on a new task, there is a risk that the model might overfit the new data and lose some of the knowledge it gained during pre-training. This is known as catastrophic forgetting. The model's weights are updated primarily for the new task, and it might lose its knowledge of previous tasks. LoRA can potentially mitigate catastrophic forgetting by keeping pre-trained weights frozen so they are not changed during fine-tuning.
- **Portability:** Due to its reduced number of parameters that are trained and original weights frozen, the LoRA model is compact and mobile. The extent to which the rank of weight matrices is reduced affects the final model size. A higher rank reduction will result in a smaller model. In any case, LoRA models weigh less than a fully fine-tuned model. That enables a user, for example, to keep a variety of models for different styles to generate images without filling up their local storage.

The performance of LoRA models may be comparable or slightly degraded compared to fully fine-tuned models. However, all substantial advantages of LoRA models such as reduced processing memory, hard disk storage space, and preservation of pre-trained knowledge resulting in decreased catastrophic forgetting make us using LoRA in this work.

IV Method

In this work, I use reinforcement learning to fine-tune text-to-image models. So, first, I train a reward function to predict human preference, score image-text alignment and quality of image. Then, I fine-tune

the text-to-image model by optimizing the expected reward of a diffusion model's image output to better align it with human feedback. I also incorporate Kullback–Leibler (KL) divergence with respect to the pre-trained model as regularization.

1 Training reward model

Admittedly, human evaluation is after all the touchstone for human preference for synthesized images; but it is limited by labor costs and hard to scale up. I aim to model human preference based on annotations, which can lead to a virtual evaluator free from dependence on humans.

First, to measure image-text alignment and image quality, I use CLIP Model to extract image and text features, combine them into embeddings. Then I train a reward function $r_\phi(x, z)$ (parameterized by ϕ) that maps the CLIP embeddings to a scalar value for preference comparison (Figure 5).

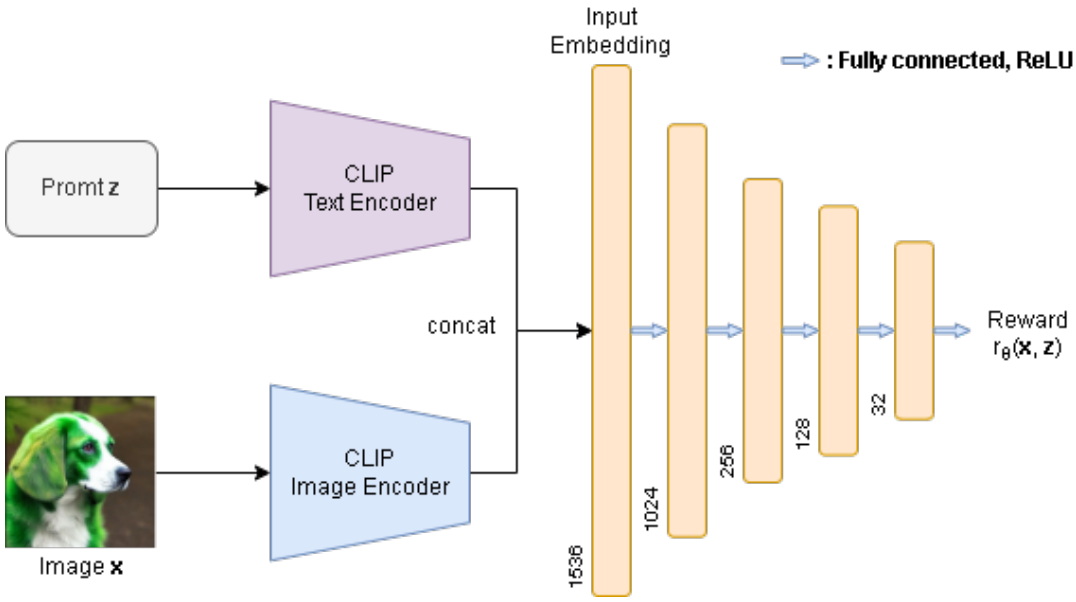

Figure 5: *Reward model architecture*

Similar to reward model training for language model of previous works [12], [15], I formulate the preference annotations as rankings. In dataset, I have $k \in [4, 9]$ images ranked for the same prompt z (the best to the worst are denoted as $x_1, x_2, \ldots, x_k$) and get at most C_k^2 comparison pairs if no ties between two images. For each comparison, if x_i is better and x_j is worse, the loss function can be formulated as:

$$\text{loss}(\theta) = -\mathbb{E}_{(z,x_i,x_j)\sim\mathcal{D}}\left[\log\left(\sigma\left(r_\theta\left(x_i, z\right) - r_\theta\left(x_j, z\right)\right)\right)\right] \tag{4}$$

where $r_\theta(x, z)$ is a scalar value of reward model for prompt z and generated image x.

2 Fine-tuning text-to-image models

After having learned reward function r_θ, I use it to fine-tune the text-to-image model with reinforcement learning. After generating images, the model uses feedbacks from reward model to improve its performance on future tasks by minimizing loss function in Eq. 5. This process is illustrated in Figure 6. For computational efficiency, I just fine-tune the denoising U-Net of stable diffusion model.

In the observation, along denoising steps (i.e., 40 in my case), the Stable Diffusion model can generate high quality images after 30-40 denoising steps. So, I conclude that my reward model can score for generations x_0 after 30 steps of denoising, unnecessarily the final step, could serve as reliable feedback for improving models.

Refer to Reward Feedback Learning (ReFL) of Xu et al. [21], I choose an algorithm to directly fine-tune text-to-image by viewing the scores of reward model as human preference losses to back-propagate gradients to a randomly-picked step t (in this case $t \in [30, 40)$) in the denoising process. The reason for

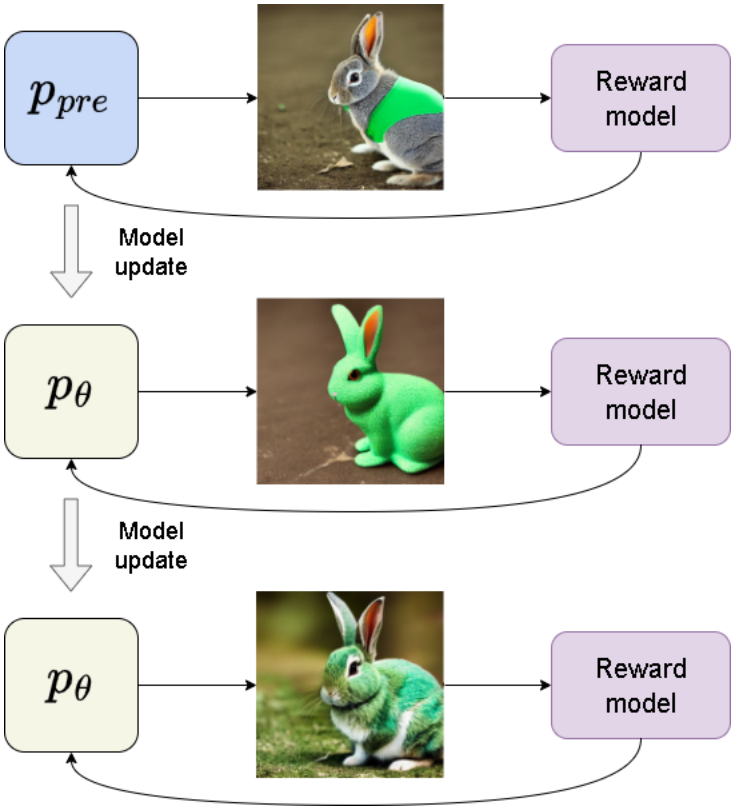

Figure 6: *Illustration of RL fine-tuning.*

the random selection of t instead of using the last step is that, if only the gradient of the last denoising step is retained, the training is proved very unstable and the results are bad.

In practice, the risk of fine-tuning purely based on the reward model learned from human or AI feedback is that the model may overfit to the reward and discount the "skill" of the initial diffusion model to a greater degree than warranted. To avoid this phenomenon, similar to [12], [15], I incorporate Kullback–Leibler (KL) divergence with respect to the pre-trained model as regularization, treating this as an implicit reward that ensures the updated model is not too far from the original one.

The final loss form is written as:

$$\mathcal{L} = \mathbb{E}_{p(z)}\mathbb{E}_{t\sim[0,10)}\mathbb{E}_{p_\theta(x_t|z)}\left[-\alpha r_\theta\left(x_t, z\right) + \beta\,\mathrm{KL}\left(p_\theta\left(x_t \mid x_{t+1}, z\right) \| p_{\mathrm{pre}}\left(x_t \mid x_{t+1}, z\right)\right)\right] \tag{5}$$

where α, β are the reward and KL weights, respectively, θ denotes the parameters of the text-to-image model.

V Experiment & Result

1 Reward model

Environment: I implemented training reward model on environment provided by Google Colab (with 1x 15GB NVIDIA Tesla T4 GPU, 1x Intel Xeon 2.0Ghz CPU, 12.7GB RAM).

Dataset: I used ImageRewardDB [21], which contains 8,878 prompts and 136,892 pairs of image comparisons. This dataset get prompts and images from the DiffusionDB dataset [48], and collects image preference ratings via crowd workers.

Model: I use ViT-L/14 CLIP model [35] to extract image and text embeddings and train a MLP using these embeddings as input. Specifically, I use five-layer MLPs with 1024, 256, 128, 32 hidden dimensions respectively (Figure 5). I use ReLUs for the activation function between layers, and I use the linear activation for output. To avoid overfitting, I use Dropout on input layer ($rate^{[0]} = 0.2$) and first two hidden layers ($rate^{[1]} = 0.2, rate^{[2]} = 0.1$)

Training: I train reward model using Adam [51] with $\beta_1 = 0.9, \beta_2 = 0.99, \epsilon = 1e-8$. The model is trained for a total 20 epochs with initial learning rate $1e-5$ and batch size 256. To improve training

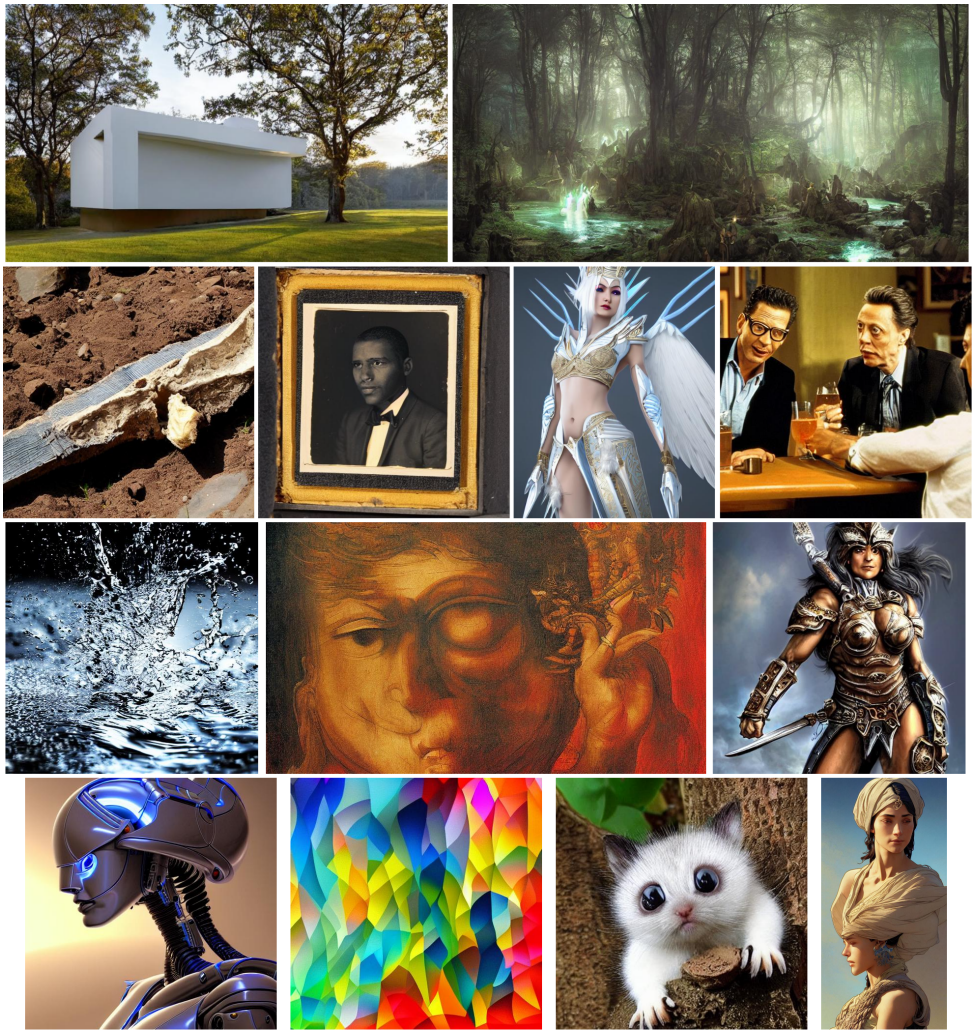

Figure 7: *Some samples from ImageRewardDB.*

performance and and reduced processing memory, I freeze CLIP layers' parameters save state that shows the best performance on valididation set.

Evaluation metric and result:

Preference Accuracy is the correctness of a scorer choosing the same one from two different images of one prompt with a human. Table 1 show results of my reward model and comparison methods on human preference prediction. Preference accuracy is from the test set of 466 prompts (6,399 comparisons). My reward model outperforms CLIP Score, Aethetic Score and BLIP Score and is 2.34% less than ImageReward. The preference accuracy of my reward model reaches up to 62.82%, which is 12.82% more than 50% (random).

Table 1: *Results of my reward model and comparison methods on human preference prediction.*

Model	Preference Acc.
CLIP Score	54.82
Aethetic Score	57.35
BLIP Score	57.76
ImageReward	**65.14**
Mine	62.82

2 Fine-tuning text-to-image model

Environment: I fine-tune the model on environment provided by Kaggle (with 1x 16GB NVIDIA Tesla P100 GPU, 1x Intel Xeon 2.20 GHz CPU, 32GB RAM).

Prompt set: I use prompt set for fine-tune models from ImageReward [21], which consists 10K prompts sampled from DiffusionDB [48].

Model: As my baseline generative model, I use Stable Diffusion v1.5 [3], which has been pre-trained on large image-text datasets [52], [53] with 1.45B parameters. For compute-efficient fine-tuning, I use Low-Rank Adaption (LoRA) [50], which freezes the parameters of the pre-trained model and introduces low-rank trainable weights. I apply LoRA to the U-Net [54] module with lora rank 4 and only update the added weights.

Training: I fine-tune the model using AdamW [55] with default hyper-parameters in AdamW. About other hyperparameters, I set learning rate = 1e-5, batch size m = 1, $\alpha = 100$, $\beta = 0.01$, timesteps $T = 40$. The model have been fine-tuned in half-precision for 10 epochs, equivalent to more than 300 hours.

Evaluation metrics and result:

For evaluation, I sample 200 prompts from test set of ImageReward, which consists 466 prompts. Then I generate 5 images from each prompt and report the average scores of all images. For scoring, I use my trained reward function. However, to be fair and objective, I use some additional metrics. Here, I use both ImageReward and the aesthetic predictor [53], which is trained to predict the aesthetic aspects of generated images. The evaluation result is shown in Figure 8.

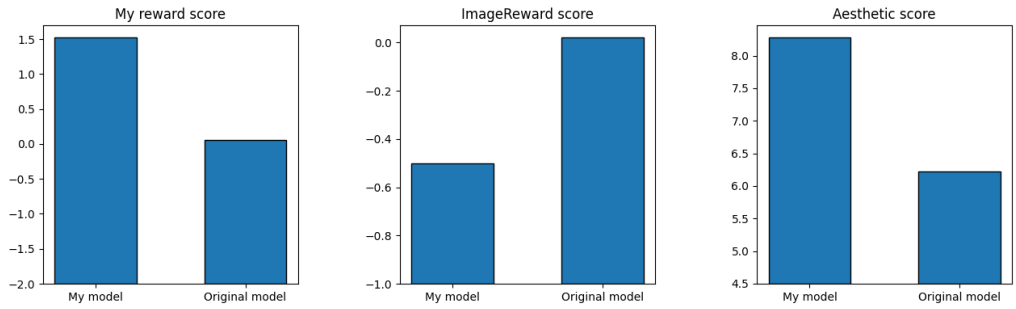

Figure 8: *My reward scores, ImageReward and Aesthetic scores are averaged over 1000 samples from each model.*

As shown in Figure 8, my model enjoy higher my reward score and aesthetic score than original model when evaluated with the same prompt set but averaged ImageReward of my model is lower than original model. This proves that if we have a good evaluating image-text alignment method, we can completely improve text-to-image model to generate better images that is meticulously conform to input prompt (higher scores).

However, in my experiment, as we can observe in Figure 9, my fine-tuned models generates images that are not really good. I expect that this is because:

1. Reward model is still simple, not good enough to evaluate image-text alignment.
2. The hyperparameters have not been optimized.
3. The fine-tuning method is still naive and need to be improved (e.g. loss function).

VI Conclusion

In this work, I have completed researching the necessary knowledge for my project and I have also proposed a simple fine-tuning text-to-image models method with artificial intelligence feedback. Although experiment result is not good as expect, it is possible that the fine-tuned model generates better images if we have better evaluating image-text alignment method. In addition, because of insufficient resources, I have not experimented many other algorithms. So I hope I can improve these in my thesis.

Limitations and future directions: There are several limitations and future directions:

- *More diverse and large human dataset:* The diversity and size of human data affect the performance of reward model. Consequently, if we have more diverse and large dataset, we can train better reward models.

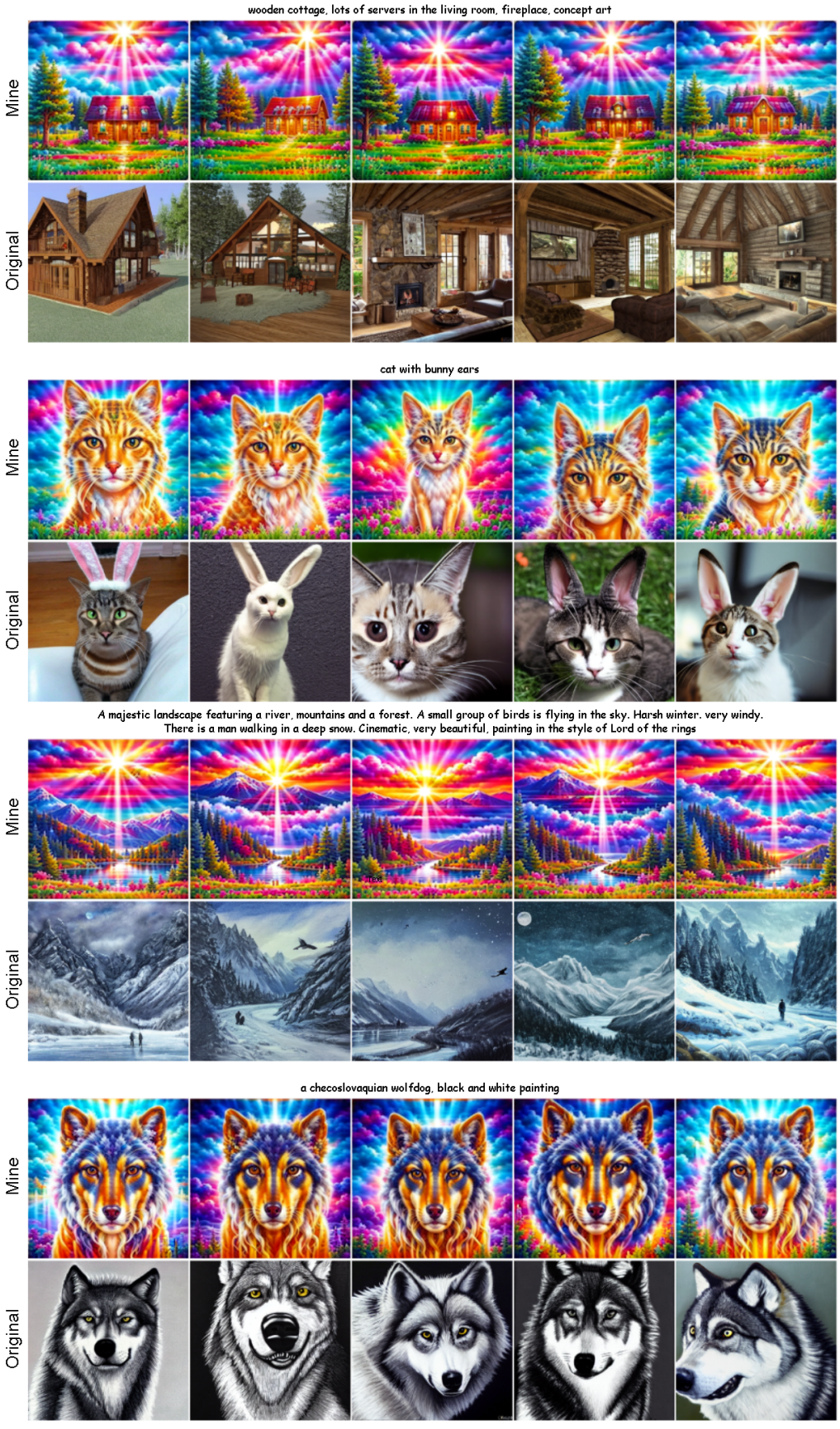

Figure 9: *Some examples of my fine-tuned model and original model.*

- *Different reward model architecture:* In this work, the architecture of reward model is still simple. So we can expect that improving the architecture of reward model will enhance preference accuracy of reward model.
- *Improving reinforcement learning algorithms:* For fine-tuning the text-to-image model, I try to maximize the expected reward of the generated images given the prompt distribution. However, this can lead to overfitting to the reward function. For this reason, I will try to improve fine-tuning algorithms.

Plans for my thesis: For the future plans for my thesis, I will try to improve my reward model to evaluate the generated images more accurately as well as try to improve the model tuning algorithm. Specifically, I may try with other loss functions, e.g. reward-weighted log likelihood. I believe that my project will achieve better results in the upcoming thesis.

References

[1] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, vol. 1, no. 2, p. 3, 2022.

[2] A. Ramesh, M. Pavlov, G. Goh, *et al.*, "Zero-shot text-to-image generation," in *International Conference on Machine Learning*, PMLR, 2021, pp. 8821–8831.

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[4] J. Yu, Y. Xu, J. Y. Koh, *et al.*, "Scaling autoregressive models for content-rich text-to-image generation," *arXiv preprint arXiv:2206.10789*, vol. 2, no. 3, p. 5, 2022.

[5] C. Saharia, W. Chan, S. Saxena, *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 479–36 494, 2022.

[6] K. Lee, H. Liu, M. Ryu, *et al.*, "Aligning text-to-image models using human feedback," *arXiv preprint arXiv:2302.12192*, 2023.

[7] W. Feng, X. He, T.-J. Fu, *et al.*, "Training-free structured diffusion guidance for compositional text-to-image synthesis," *arXiv preprint arXiv:2212.05032*, 2022.

[8] N. Liu, S. Li, Y. Du, A. Torralba, and J. B. Tenenbaum, "Compositional visual generation with composable diffusion models," in *European Conference on Computer Vision*, Springer, 2022, pp. 423–439.

[9] R. Liu, D. Garrette, C. Saharia, *et al.*, "Character-aware models improve visual text rendering," *arXiv preprint arXiv:2212.10562*, 2022.

[10] V. Petsiuk, A. E. Siemenn, S. Surbehera, *et al.*, "Human evaluation of text-to-image models on a multi-task benchmark," *arXiv preprint arXiv:2211.12112*, 2022.

[11] D. M. Ziegler, N. Stiennon, J. Wu, *et al.*, "Fine-tuning language models from human preferences," *arXiv preprint arXiv:1909.08593*, 2019.

[12] N. Stiennon, L. Ouyang, J. Wu, *et al.*, "Learning to summarize with human feedback," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3008–3021, 2020.

[13] J. Wu, L. Ouyang, D. M. Ziegler, *et al.*, "Recursively summarizing books with human feedback," *arXiv preprint arXiv:2109.10862*, 2021.

[14] R. Nakano, J. Hilton, S. Balaji, *et al.*, "Webgpt: Browser-assisted question-answering with human feedback," *arXiv preprint arXiv:2112.09332*, 2021.

[15] L. Ouyang, J. Wu, X. Jiang, *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[16] Y. Bai, A. Jones, K. Ndousse, *et al.*, "Training a helpful and harmless assistant with reinforcement learning from human feedback," *arXiv preprint arXiv:2204.05862*, 2022.

[17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[18] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[19] Y. Kirstain, A. Polyak, U. Singer, S. Matiana, J. Penna, and O. Levy, "Pick-a-pic: An open dataset of user preferences for text-to-image generation," *arXiv preprint arXiv:2305.01569*, 2023.

[20] X. Wu, K. Sun, F. Zhu, R. Zhao, and H. Li, "Better aligning text-to-image models with human preference," *arXiv preprint arXiv:2303.14420*, 2023.

[21] J. Xu, X. Liu, Y. Wu, *et al.*, "Imagereward: Learning and evaluating human preferences for text-to-image generation," *arXiv preprint arXiv:2304.05977*, 2023.

[22] Y. Fan, O. Watkins, Y. Du, *et al.*, "Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models," *arXiv preprint arXiv:2305.16381*, 2023.

[23] J. Sun, D. Fu, Y. Hu, *et al.*, "Dreamsync: Aligning text-to-image generation with image understanding feedback," *arXiv preprint arXiv:2311.17946*, 2023.

[24] E. Mansimov, E. Parisotto, J. L. Ba, and R. Salakhutdinov, "Generating images from captions with attention," *arXiv preprint arXiv:1511.02793*, 2015.

[25] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," in *International conference on machine learning*, PMLR, 2015, pp. 1462–1471.

[26] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*, PMLR, 2015, pp. 2256–2265.

[27] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *Advances in neural information processing systems*, vol. 32, 2019.

[28] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[29] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.

[30] A. Nichol, P. Dhariwal, A. Ramesh, *et al.*, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," *arXiv preprint arXiv:2112.10741*, 2021.

[31] Z. Feng, Z. Zhang, X. Yu, *et al.*, "Ernie-vilg 2.0: Improving text-to-image diffusion model with knowledge-enhanced mixture-of-denoising-experts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10 135–10 145.

[32] H. Zhang, W. Yin, Y. Fang, *et al.*, "Ernie-vilg: Unified generative pre-training for bidirectional vision-language generation," *arXiv preprint arXiv:2112.15283*, 2021.

[33] K. Lee, L. Smith, and P. Abbeel, "Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training," *arXiv preprint arXiv:2106.05091*, 2021.

[34] H. Liu, C. Sferrazza, and P. Abbeel, "Languages are rewards: Hindsight finetuning using human feedback," *arXiv preprint arXiv:2302.02676*, 2023.

[35] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.

[36] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *International Conference on Machine Learning*, PMLR, 2022, pp. 12 888–12 900.

[37] S. Karthik, K. Roth, M. Mancini, and Z. Akata, "If at first you don't succeed, try, try again: Faithful diffusion-based text-to-image generation by selection," *arXiv preprint arXiv:2305.13308*, 2023.

[38] K. Sohn, N. Ruiz, K. Lee, *et al.*, "Styledrop: Text-to-image generation in any style," *arXiv preprint arXiv:2306.00983*, 2023.

[39] C. Mou, X. Wang, L. Xie, *et al.*, "T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models," *arXiv preprint arXiv:2302.08453*, 2023.

[40] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22 500–22 510.

[41] N. Ruiz, Y. Li, V. Jampani, *et al.*, "Hyperdreambooth: Hypernetworks for fast personalization of text-to-image models," *arXiv preprint arXiv:2307.06949*, 2023.

[42] L. Han, Y. Li, H. Zhang, P. Milanfar, D. Metaxas, and F. Yang, "Svdiff: Compact parameter space for diffusion fine-tuning," *arXiv preprint arXiv:2303.11305*, 2023.

[43] D. Epstein, A. Jabri, B. Poole, A. A. Efros, and A. Holynski, "Diffusion self-guidance for controllable image generation," *arXiv preprint arXiv:2306.00986*, 2023.

[44] S. Hong, G. Lee, W. Jang, and S. Kim, "Improving sample quality of diffusion models using self-attention guidance," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 7462–7471.

[45] R. Rassin, E. Hirsch, D. Glickman, S. Ravfogel, Y. Goldberg, and G. Chechik, "Linguistic binding in diffusion models: Enhancing attribute correspondence through attention map alignment," *arXiv preprint arXiv:2306.08877*, 2023.

[46] Z. Zhang, C. Li, W. Sun, X. Liu, X. Min, and G. Zhai, "A perceptual quality assessment exploration for aigc images," *arXiv preprint arXiv:2303.12618*, 2023.

[47] X. Wu, Y. Hao, K. Sun, *et al.*, "Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis," *arXiv preprint arXiv:2306.09341*, 2023.

[48] Z. J. Wang, E. Montoya, D. Munechika, H. Yang, B. Hoover, and D. H. Chau, "Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models," *arXiv preprint arXiv:2210.14896*, 2022.

[49] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[50] E. J. Hu, Y. Shen, P. Wallis, *et al.*, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[52] C. Schuhmann, R. Vencu, R. Beaumont, *et al.*, "Laion-400m: Open dataset of clip-filtered 400 million image-text pairs," *arXiv preprint arXiv:2111.02114*, 2021.

[53] C. Schuhmann, R. Beaumont, R. Vencu, *et al.*, "Laion-5b: An open large-scale dataset for training next generation image-text models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 25 278–25 294, 2022.

[54] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Springer, 2015, pp. 234–241.

[55] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.