TextCraftor: Your Text Encoder Can be Image Quality Controller

Yanyu Li[1,2] Xian Liu[1] Anil Kag[1] Ju Hu[1] Yerlan Idelbayev[1]
Dhritiman Sagar[1] Yanzhi Wang[2] Sergey Tulyakov[1] Jian Ren[1]
[1]Snap Inc. [2]Northeastern University

Figure 1. **Example generated images.** For each prompt, we show images generated from three different models, which are SDv1.5, TextCraftor, TextCraftor + UNet, listed from left to right. The random seed is fixed for all generation results.

Abstract

Diffusion-based text-to-image generative models, e.g.*, Stable Diffusion, have revolutionized the field of content generation, enabling significant advancements in areas like image editing and video synthesis. Despite their formidable capabilities, these models are not without their limitations. It is still challenging to synthesize an image that aligns well with the input text, and multiple runs with carefully crafted prompts are required to achieve satisfactory results. To mitigate these limitations, numerous studies have endeavored to fine-tune the pre-trained diffusion models,* i.e.*, UNet, utilizing various technologies. Yet, amidst these efforts, a pivotal question of text-to-image diffusion model training has remained largely unexplored:* **Is it possible and feasible to fine-tune the text encoder to improve the performance of text-to-image diffusion models?** *Our findings reveal that, instead of replacing the CLIP text encoder used in Stable Diffusion with other large language models, we can enhance it through our proposed fine-tuning approach, TextCraftor, leading to substantial improvements in quantitative benchmarks and human assessments. Interestingly, our technique also empowers controllable image generation through the interpolation of different text encoders fine-tuned with various rewards. We also demonstrate that TextCraftor is orthogonal to UNet finetuning, and can be combined to further improve generative quality.*

1. Introduction

Recent breakthroughs in text-to-image diffusion models have brought about a revolution in content generation [10, 17, 27, 40, 51]. Among these models, the open-sourced Stable Diffusion (SD) has emerged as the *de facto* choice for a wide range of applications, including image editing, super-resolution, and video synthesis [4, 18, 25, 29, 31, 42, 44, 47, 59]. Though trained on large-scale datasets, SD still holds two major challenges. *First*, it often produces images that do not align well with the provided prompts [5, 57]. *Second*, generating visually pleasing images frequently requires multiple runs with different random seeds and manual prompt engineering [13, 53]. To address the *first* challenge, prior studies explore the substitution of the CLIP text encoder [36] used in SD with other large language models like T5 [7, 43]. Nevertheless, the large T5 model has an order of magnitude more parameters than CLIP, resulting in additional storage and computation overhead. In tackling the *second* challenge, existing works fine-tune the pre-trained UNet from SD on paired image-caption datasets with reward functions [8, 34, 56]. Nonetheless, models trained on constrained datasets may still struggle to generate high-quality images for unseen prompts.

Stepping back and considering the pipeline of text-to-image generation, the text encoder and UNet should *both* significantly influence the quality of the synthesized images. Despite substantial progress in enhancing the UNet model [15, 46], limited attention has been paid to improving the text encoder. This work aims to answer a pivotal question: *Can fine-tuning a pre-trained text encoder used in the generative model enhance performance, resulting in better image quality and improved text-image alignment?*

To address this challenge, we propose *TextCraftor*, an end-to-end fine-tuning technique to enhance the pre-trained text encoder. Instead of relying on paired text-image datasets, we demonstrate that reward functions (*e.g.*, models trained to automatically assess the image quality like aesthetics model [1], or text-image alignment assessment models [23, 54]) can be used to improve text-encoder in a differentiable manner. By only necessitating text prompts during training, *TextCraftor* enables the on-the-fly synthesis of training images and alleviates the burden of storing and loading large-scale image datasets. We summarize our findings and contributions as follows:

- We demonstrate that for a well-trained text-to-image diffusion model, fine-tuning text encoder is a buried gem, and can lead to significant improvements in image quality and text-image alignment (as in Fig. 1 & 3). Compared with using larger text encoders, *e.g.*, SDXL, *TextCraftor* does not introduce extra computation and storage overhead. Compared with prompt engineering, *TextCraftor* reduces the risks of generating irrelevant content.
- We introduce an effective and stable text encoder fine-tuning pipeline supervised by public reward functions. The proposed alignment constraint preserves the capability and generality of the large-scale CLIP-pretrained text encoder, making *TextCraftor* the first generic reward fine-tuning paradigm among concurrent arts. Comprehensive evaluations on public benchmarks and human assessments demonstrate the superiority of *TextCraftor*.
- We show that the textual embedding from different fine-tuned and original text encoders can be interpolated to achieve more diverse and controllable style generation. Additionally, *TextCraftor* is orthogonal to UNet finetuning. We further show quality improvements by subsequently fine-tuning UNet with the improved text encoder.

2. Related Works

Text-to-Image Diffusion Models. Recent efforts in the synthesis of high-quality, high-resolution images from natural language inputs have showcased substantial progress [2, 40]. Diverse investigations have been conducted to improve model performance by employing various network architectures and training pipelines, such as GAN-based approaches [20], auto-regressive models [30, 58], and diffusion models [17, 21, 48, 50, 51]. Since the introduction of the Stable Diffusion models and their state-of-the-art performance in image generation and editing tasks, they have emerged as the predominant choice [40]. Nevertheless, they exhibit certain limitations. For instance, the generated images may not align well with the provided text prompts [57]. Furthermore, achieving high-quality images may necessitate extensive prompt engineering and multiple runs with different random seeds [13, 53]. To address these challenges, one potential improvement involves replacing the pre-trained CLIP text-encoder [36] in the Stable Diffusion model with T5 [7] and fine-tuning the model using high-quality paired data [9, 43]. However, it is crucial to note that such an approach incurs a substantial training cost. Training the Stable Diffusion model alone from scratch demands considerable resources, equivalent to $6,250$ A100 GPUs days [5]. This work improves pre-trained text-to-image models while significantly reducing computation costs.

Automated Performance Assessment of Text-to-Image Models. Assessing the performance of text-to-image models has been a challenging problem. Early methods use automatic metrics like FID to gauge image quality and CLIP scores to assess text-image alignment [37, 38]. However, subsequent studies have indicated that these scores exhibit limited correlation with human perception [33]. To address such discrepancies, recent research has delved into training models specifically designed for evaluating image quality for text-to-image models. Examples include ImageReward [56], PickScore [23], and human preference scores [54, 55], which leverage human-annotated images to train the quality estimation models. In our work, we leverage these models, along with an image aesthetics model [1], as reward functions for enhancing visual quality and text-image alignment for the text-to-image diffusion models.

Fine-tuning Diffusion Models with Rewards. In response to the inherent limitations of pre-trained diffusion models, various strategies have been proposed to elevate generation quality, focusing on aspects like image color, composition, and background [11, 24]. One direction utilizes reinforcement learning to fine-tune the diffusion model [3, 12]. Another area fine-tunes the diffusion models with reward function in a differentiable manner [56]. Following this trend, later studies extend the pipeline to trainable LoRA weights [19] with the text-to-image models [8, 34]. In our work, we delve into the novel exploration of fine-tuning the text-encoder using reward functions in a differentiable manner, a dimension that has not been previously explored.

Improving Textual Representation. Another avenue of research focuses on enhancing user-provided text to generate images of enhanced quality. Researchers use large language models, such as LLAMA [52], to refine or optimize text prompts [14, 35, 60]. By improving the quality of prompts, the text-to-image model can synthesize higher-quality images. However, the utilization of additional language models introduces increased computational and storage demands. This study demonstrates that by fine-tuning the text encoder, the model can gain a more nuanced understanding of the given text prompts, obviating the need for additional language models and their associated overhead.

3. Method

3.1. Preliminaries of Latent Diffusion Models

Latent Diffusion Models. Diffusion models convert the real data distribution *e.g.*, images, into a noisy distribution, *e.g.*, Gaussian distribution, and can reverse such a process to for randomly sampling [48]. To reduce the computation cost, *e.g.*, the number of denoising steps, latent diffusion model (LDM) proposes to conduct the denoising process in the latent space [40] using a UNet [17, 41], where real data is encoded through variational autoencoder (VAE) [22, 39]. The latent is then decoded into an image during inference time. LDM demonstrates promising results for text-conditioned image generation. Trained with large-scale text-image paired datasets [45], a series of LDM models, namely, Stable Diffusion [40], are obtained. The text prompts are processed by a pre-trained text encoder, which is the one from CLIP [36] used by Stable Diffusion, to obtain textual embedding as the condition for image generation. In this work, we use the Stable Diffusion as the baseline model to conduct most of our experiments, as it is widely adopted in the community for various tasks.

Formally, let ($\mathbf{x}$, $\mathbf{p}$) be the real-image and prompt data pair (for notation simplicity, $\mathbf{x}$ also represents the data encoded by VAE) drawn from the distribution $p_{\text{data}}(\mathbf{x}, \mathbf{p})$, $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\cdot)$ be the diffusion model with parameters $\boldsymbol{\theta}$, $\mathcal{T}_{\boldsymbol{\varphi}}(\cdot)$ be the text encoder parameterized by $\boldsymbol{\varphi}$, training the text-to-image LDM under the objective of noise prediction can be formulated as follows [17, 48, 51]:

$$\min_{\boldsymbol{\theta}} \; \mathbb{E}_{t\sim U[0,1],(\mathbf{x},\mathbf{p})\sim p_{\text{data}}(\mathbf{x},\mathbf{p}),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})} \; ||\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(t, \mathbf{z}_t, \mathbf{c}) - \boldsymbol{\epsilon}||_2^2, \quad (1)$$

where $\boldsymbol{\epsilon}$ is the ground-truth noise; t is the time step; $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$ is the noised sample with α_t represents the signal and σ_t represents the noise, that both decided by the scheduler; and $\mathbf{c}$ is the textual embedding such that $\mathbf{c} = \mathcal{T}_{\boldsymbol{\varphi}}(\mathbf{p})$.

During the training of SD models, the weights of text encoder $\mathcal{T}$ are fixed. However, the text encoder from CLIP model is optimized through the contrastive objective between text and images. Therefore, it does not necessarily learn the semantic meaning of the prompt, resulting the generated image might not align well with the given prompt using such a text encoder. In Sec. 3.2, we introduce the technique of improving the text encoder without using the text and image contrastive pre-training in CLIP [36].

Denoising Scheduler – DDIM. After a text-to-image diffusion model is trained, we can sample Gaussian noises for the same text prompt using numerous samplers, such as DDIM [49], that iteratively samples from t to its previous step t' with the following denoising process, until t becomes 0:

$$\mathbf{z}_{t'} = \alpha_{t'} \frac{\mathbf{z}_t - \sigma_t \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(t, \mathbf{z}_t, \mathbf{c})}{\alpha_t} + \sigma_{t'} \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(t, \mathbf{z}_t, \mathbf{c}). \quad (2)$$

Classifier-Free Guidance. One effective approach to improving the generation quality during the sampling stage is the classifier-free guidance (CFG) [16]. By adjusting the guidance scale w in CFG, we can further balance the trade-off between the fidelity and the text-image alignment of the synthesized image. Specifically, for the process of text-conditioned image generation, by letting $\varnothing$ denote the null text input, classifier-free guidance can be defined as follows:

$$\hat{\boldsymbol{\epsilon}} = w \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(t, \mathbf{z}_t, \mathbf{c}) - (w - 1) \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(t, \mathbf{z}_t, \varnothing). \quad (3)$$

3.2. Text Encoder Fine-tuning with Reward Propagation

We introduce and experiment with two techniques for fine-tuning the text encoder by reward guidance.

3.2.1 Directly Fine-tuning with Reward

Recall that for a normal training process of diffusion models, we sample from real data and random noise to perform forward diffusion: $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$, upon which the denoising UNet, $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\cdot)$, makes its (noise) prediction. Therefore, instead of calculating $\mathbf{z}_{t'}$ as in Eqn. 2, we can alternatively predict the original data as follows [49],

$$\hat{\mathbf{x}} = \frac{\mathbf{z}_t - \sigma_t \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(t, \mathbf{z}_t, \mathcal{T}_{\boldsymbol{\varphi}}(\mathbf{p}))}{\alpha_t}, \quad (4)$$

where $\hat{\mathbf{x}}$ is the estimated real sample, which is an image for the text-to-image diffusion model. Our formulation works for both pixel-space and latent-space diffusion models, where in latent diffusion, $\hat{\mathbf{x}}$ is actually post-processed by the VAE decoder before feeding into reward models. Since the decoding process is also differentiable, for simplicity, we omit this process in formulations and simply refer $\hat{\mathbf{x}}$ as the predicted image. With $\hat{\mathbf{x}}$ in hand, we are able to utilize public reward models, denoted as $\mathcal{R}$, to assess the quality of the generated image. Therefore, to improve the text encoder used in the diffusion model, we can optimize its weights, *i.e.*, $\boldsymbol{\varphi}$ in $\mathcal{T}$, with the learning objective as maximizing the quality scores predicted by reward models.

More specifically, we employ both image-based reward model $\mathcal{R}(\hat{\mathbf{x}})$, *i.e.*, Aesthetic score predictor [1], and

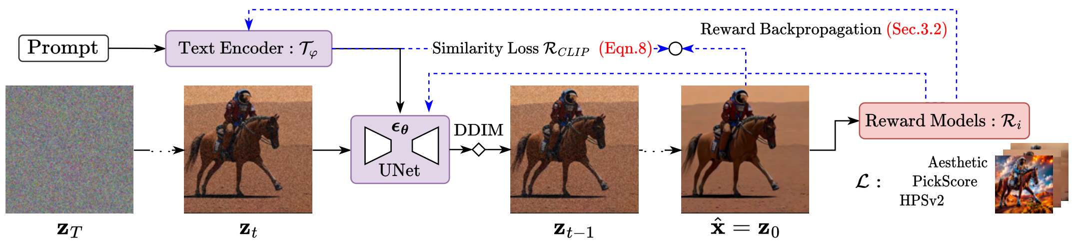

Figure 2. **Overview of TextCraftor**, an end-to-end text encoder fine-tuning paradigm based on prompt data and reward functions. The text embedding is forwarded into the DDIM denoising chain to obtain the output image and compute the reward loss, then we backward to update the parameters of the text encoder (and optionally UNet) by maximizing the reward.

text-image alignment-based reward models $\mathcal{R}(\hat{\mathbf{x}}, \mathbf{p})$, *i.e.*, HPSV2 [54] and PickScore [23]. Consequently, the loss function for maximizing the reward scores can be defined as follows,

$$\begin{aligned}\mathcal{L}(\varphi) &= -\mathcal{R}(\hat{\mathbf{x}}, \cdot/\mathbf{p}) \\ &= -\mathcal{R}(\frac{\mathbf{z}_t - \sigma_t \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(t, \mathbf{z}_t, \mathcal{T}_\varphi(\mathbf{p}))}{\alpha_t}, \cdot/\mathbf{p}).\end{aligned} \tag{5}$$

Note that when optimizing Eqn. 5, the weights for all reward models and the UNet model are fixed, while only the weights in the CLIP text encoder are modified.

Discussion. Clearly, directly fine-tuning shares a similar training regime with regular training of diffusion models, where we are ready to employ text-image paired data $(\mathbf{x}, \mathbf{p})$ and predict reward by converting predicted noise into the predicted real data $\hat{\mathbf{x}}$. However, considering the very beginning (noisy) timesteps, the estimated $\hat{\mathbf{x}}$ can be *inaccurate* and *less reliable*, making the predicted reward less meaningful. Instead of utilizing $\hat{\mathbf{x}}$, Liu *et al*. [26] propose to fine-tune the reward models to enable a noisy latent ($\mathbf{z}_t$) aware score prediction, which is out of the scope of this work. For the best flexibility and sustainability of our method, we only investigate publicly available reward models, thus we directly employ $\hat{\mathbf{x}}$ prediction. We discuss the performance of direct finetuning in Section. 4.

3.2.2 Prompt-Based Fine-tuning

As an alternative way to overcome the problem of the inaccurate $\hat{\mathbf{x}}$ prediction, given a specific text prompt $\mathbf{p}$ and an initial noise $\mathbf{z}_T$, we can iteratively solve the denoising process in Eqn. 2 to get $\hat{\mathbf{x}} = \mathbf{z}_0$, which can then be substituted to Eqn. 5 to compute the reward scores. Consequently, we are able to precisely predict $\hat{\mathbf{x}}$, and also eliminate the need for paired text-image data and perform the reward fine-tuning with *only prompts* and a pre-defined denoising schedule, *i.e.*, 25-steps DDIM in our experiments. Since each timestep in the training process is differentiable, the gradient to update φ in $\mathcal{T}$ can be calculated through

Algorithm 1 Prompt-Based Reward Finetuning

Require: Pretrained UNet $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}$; pretrained text encoder $\mathcal{T}_\varphi$; prompt set: $\mathbb{P}\{\mathbf{p}\}$.
Ensure: $\mathcal{T}_\varphi$ (optionally $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}$ if fine-tuning UNet) converges and maximizes $\mathcal{L}_{total}$.
$\rightarrow$ **Perform text encoder fine-tuning**.
Freeze UNet $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}$ and reward models $\mathcal{R}_i$, activate $\mathcal{T}_\varphi$.
while $\mathcal{L}_{total}$ not converged **do**
  Sample $\mathbf{p}$ from $\mathbb{P}$; $t = T$
  while $t > 0$ **do**
    $\mathbf{z}_{t-1} \leftarrow \alpha_{t'} \frac{\mathbf{z}_t - \sigma_t \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(t, \mathbf{z}_t, \mathcal{T}_\varphi(\mathbf{p}))}{\alpha_t} + \sigma_{t'} \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(t, \mathbf{z}_t, \mathcal{T}_\varphi(\mathbf{p}))$
  end while
  $\hat{\mathbf{x}} \leftarrow \mathbf{z}_0$
  $\mathcal{L}_{total} \leftarrow -\sum_i \gamma_i \mathcal{R}_i(\hat{\mathbf{x}}, \cdot/\mathbf{p})$.
  Backward $\mathcal{L}_{total}$ and update $\mathcal{T}_\varphi$ for last K steps.
end while
$\rightarrow$ **Perform UNet finetuning**.
Freeze $\mathcal{T}_\varphi$ and reward models $\mathcal{R}_i$, activate UNet $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}$.
Repeat the above reward training until converge.

chain rule as follows,

$$\frac{\partial \mathcal{L}}{\partial \varphi} = -\frac{\partial \mathcal{R}}{\partial \hat{\mathbf{x}}} \cdot \prod_{t=0}^{t} \frac{\partial [\alpha_{t'} \frac{\mathbf{z}_t - \sigma_t \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(t, \mathbf{z}_t, \mathcal{T}_\varphi(\mathbf{p}))}{\alpha_t} + \sigma_{t'} \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(t, \mathbf{z}_t, \mathcal{T}_\varphi(\mathbf{p}))]}{\partial \mathcal{T}_\varphi(\mathbf{p})} \cdot \frac{\partial \mathcal{T}_\varphi(\mathbf{p})}{\partial \varphi}. \tag{6}$$

It is notable that solving Eqn. 6 is memory infeasible for early (noisy) timesteps, *i.e.*, $t = \{T, T-1, ...\}$, as the computation graph accumulates in the backward chain. We apply gradient checkpointing [6] to trade memory with computation. Intuitively, the intermediate results are re-calculated on the fly, thus the training can be viewed as solving one step at a time. Though with gradient checkpointing, we can technically train the text encoder with respect to each timestep, early steps still suffer from gradient explosion and vanishing problems in the long-lasting accumulation [8]. We provide a detailed analysis of step selection in Section. 4.2. The proposed prompt-based reward finetuning is further illustrated in Fig. 2 and Alg. 1.

3.3. Loss Function

We investigate and report the results of using multiple reward functions, where the reward losses $\mathcal{L}_{total}$ can be weighted by γ and linearly combined as follows,

$$\mathcal{L}_{total} = \sum_i \mathcal{L}_i = -\sum_i \gamma_i \mathcal{R}_i(\hat{\mathbf{x}}, \cdot/\mathbf{p}). \tag{7}$$

Intuitively, we can arbitrarily combine different reward functions with various weights. However, as shown in Fig. 6, some reward functions are by nature limited in terms of their capability and training scale. As a result, fine-tuning with only one reward can result in catastrophic forgetting and mode collapse.

To address this issue, recent works [3, 56] mostly rely on careful tuning, including focusing on a specific subdomain, *e.g*., human and animals [34], and early stopping [8]. Unfortunately, this is not a valid approach in the generic and large-scale scope. In this work, we aim at enhancing generic models and eliminating human expertise and surveillance.

To achieve this, we set CLIP space similarity as an always-online constraint as follows,

$$\mathcal{R}_{CLIP} = \texttt{cosine-sim}(\mathcal{I}(\hat{\mathbf{x}}), \mathcal{T}_\varphi(\mathbf{p})), \tag{8}$$

and ensure $\gamma_{\text{CLIP}} > 0$ in Eqn. 7. Specifically, we maximize the cosine similarity between the textual embeddings and image embeddings. The textual embedding is obtained in forward propagation, while the image embedding is calculated by sending the predicted image $\hat{\mathbf{x}}$ to the image encoder $\mathcal{I}$ of CLIP. The original text encoder $\mathcal{T}_\varphi$ is pre-trained in large-scale contrastive learning paired with the image encoder $\mathcal{I}$ [36]. As a result, the CLIP constraint preserves the coherence of the fine-tuned text embedding and the original image domain, ensuring capability and generalization.

3.4. UNet Fine-tuning with Reward Propagation

The proposed fine-tuning approach for text encoder is orthogonal to UNet reward fine-tuning [8, 34], meaning that the text-encoder and UNet can be optimized under similar learning objectives to further improve performance. Note that our fine-tuned text encoder can seamlessly fit the pre-trained UNet in Stable Diffusion, and can be used for other downstream tasks besides text-to-image generation. To preserve this characteristic and avoid domain shifting, we fine-tune the UNet by freezing the finetuned text encoder $\mathcal{T}_\varphi$. The learning objective for UNet is similar as Eqn. 6, where we optimize parameters $\boldsymbol{\theta}$ of $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}$, instead of $\boldsymbol{\varphi}$.

4. Experiments

Reward Functions. We use image-based aesthetic predictor [1], text-image alignment-based CLIP predictors, (*i.e*., Human Preference Score v2 (HPSv2) [54] and PickScore [23]), and CLIP model [36]. We adopt the improved (v2) version of the aesthetic predictor that is trained on $176,000$ image-rating pairs. The predictor estimates a quality score ranging from 1 to 10, where larger scores indicate higher quality. HPSv2 is a preference prediction model trained on a large-scale well-annotated dataset of human choices, with 798K preference annotations and 420K images. Similarly, PickScore [23] is a popular human preference predictor trained with over half a million samples.

Training Datasets. We perform training on OpenPrompt[1] dataset, which includes more than 10M high quality prompts for text-to-image generation. For direct finetuning, we use the public LAION-2B dataset with conventional pre-processing, *i.e*., filter out NSFW data, resize and crop images to 512^2px, and use Aesthetics> 5.0 images.

Experimental Settings. We conduct experiments with the latest PyTorch [32] and HuggingFace Diffusers[2]. We choose Stable Diffusion v1.5 (SDv1.5) [40] as the baseline model, as it performs well in real-world human assessments with appealing model size and computation than other large diffusion models [33]. We fine-tune the ViT-L text encoder of SDv1.5, which takes 77 tokens as input and outputs an embedding with dimension 768. The fine-tuning is done on 8 NVIDIA A100 nodes with 8 GPUs per node, using AdamW optimizer [28] and a learning rate of 10^{-6}. We set CFG scale to 7.5 in all the experiments.

Comparison Approaches. We compare our method with the following approaches.

- *Pre-trained* text-to-image models that include SDv1.5, SDv2.0, SDXL Base 0.9, and DeepFloyd-XL.
- *Direct fine-tuning* that is described in Sec. 3.2.1.
- *Reinforcement learning-based approach* that optimize the diffusion model using reward functions [3].
- *Prompt engineering*. From the scope of the enhancement of text information, prompt engineering [13, 53] can be considered as a counterpart of our approach. By extending and enriching the input prompt with more detailed instructions, *e.g*., using words like 4K, photorealistic, ultra sharp, etc., the output image quality could be greatly boosted. However, prompt engineering requires case-by-case human tuning, which is not appealing in real-world applications. Automatic engineering method[3] employs text generation models to enhance the prompt, while the semantic coherence might not be guaranteed. We experiment and compare with automatic prompt engineering on both quantitative and qualitative evaluations.

Quantitative Results. We report the results with different training settings (the CLIP constraint is utilized under all the settings of our approach) on two datasets:

- We report zero-shot evaluation results for the score of

[1] https://github.com/krea-ai/open-prompts
[2] https://github.com/huggingface/diffusers
[3] https://huggingface.co/daspartho/prompt-extend

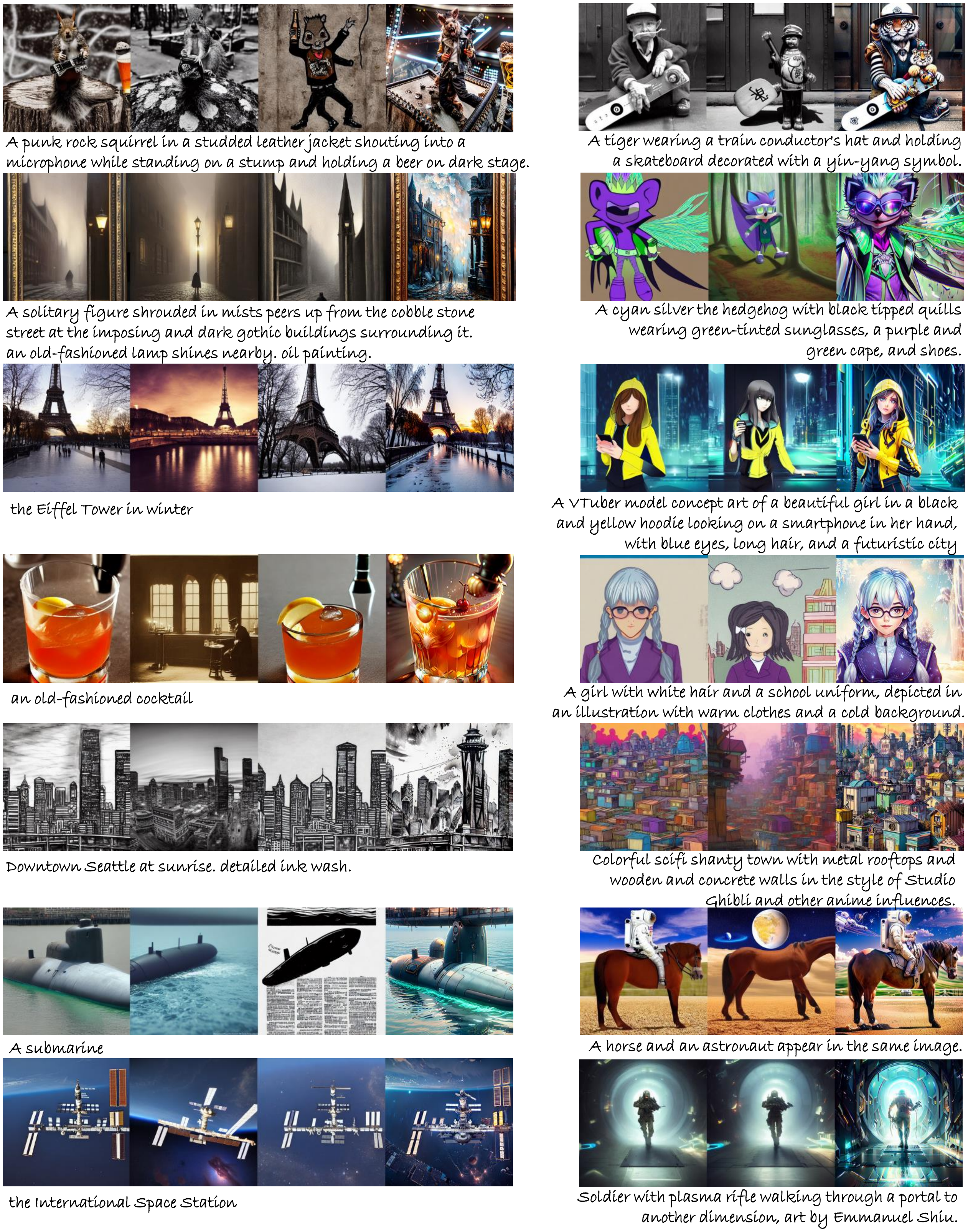

Figure 3. **Qualitative visualizations.** ***Left***: generated images on Parti-Prompts, in the order of SDv1.5, prompt engineering, DDPO, and TextCraftor. ***Right***: examples from HPSv2, ordered as SDv1.5, prompt engineering, and TextCraftor.

three rewards on Parti-Prompts [58], which contains 1632 prompts with various categories, in Tab. 1. We show experiments using a single reward function, *e.g.*, Aesthetics, and the combination of all reward functions, *i.e.*, denoted as All. We also fine-tune the UNet by freezing the fine-tuned text-encoder (TextCraftor + UNet in Tab. 1). We evaluate different methods by forwarding the generated images (or the given prompt) to reward functions to obtain scores, where higher scores indicate better performance.
- We report zero-shot results on the HPSv2 benchmark set, which contains 4 subdomains of animation, concept art, painting, and photo, with 800 prompts per category. In addition to the zero-shot model trained with combined rewards (denote as All in Tab. 2), we train the model solely with HPSv2 reward to report the best possible scores TextCraftor can achieve on the HPSv2 benchmark.

From the results, we can draw the following observations:

- Compared to the pre-trained text-to-image models, *i.e.*, SDv1.5 and SDv2.0, our TextCraftor achieves significantly higher scores, *i.e.*, Aesthetics, PickScore, and HPSv2, compared to the baseline SDv1.5. More interestingly, TextCraftor outperforms SDXL Base 0.9. and DeepFloyd-XL, which have much larger UNet and text encoder.
- Direct fine-tuning (described in Sec. 3.2.1) can not provide reliable performance improvements.
- Compared to prompt engineering, TextCraftor obtains better performance, without necessitating human effort and ambiguity. We notice that the incurred additional information in the text prompt leads to lower alignment scores.
- Compared to previous state-of-the-art DDPO [3] that performs reward fine-tuning on UNet, we show that *TextCraftor + UNet* obtains better metrics by a large margin. It is notable that DDPO is fine-tuned on subdomains, *e.g.*, animals and humans, with early stopping, limiting its capability to generalize for unseen prompts. The proposed TextCraftor is currently the first large-scale and generic reward-finetuned model.
- Lastly, fine-tuning the UNet can further improve the performance, proving that TextCraftor is orthogonal to UNet fine-tuning and can be combined to achieve significantly better performance.

Qualitative Results. We demonstrate the generative quality of TextCraftor in Fig. 1 and 3. Images are generated with the same noise seed for direct and fair comparisons. We show that with TextCraftor, the generation quality is greatly boosted compared to SDv1.5. Additionally, compared to prompt engineering, TextCraftor exhibits more reliable text-image alignment and rarely generates additional or irrelevant objects. Compared to DDPO [3], the proposed TextCraftor resolves the problem of mode collapse and catastrophic forgetting by employing text-image similarity as the constraint reward. We also show that fine-tuning the UNet models upon the TextCraftor enhanced text encoder can further boost the generation quality. From the visualizations, we observe that the reward fine-tuned models tend to generate more artistic, sharp, and colorful styles, which results from the preference of the reward models. When stronger and better reward predictors emerge in the future, TextCraftor can be seamlessly extended to obtain even better performance. Lastly, we provide a comprehensive human evaluation in Tab. 3, proving the users prefer the images synthesized by TextCraftor.

Table 1. **Comparison results on Parti-Prompts [58].** We perform TextCraftor fine-tuning on individual reward functions, including Aesthetics, PisckScore, and HPSv2, and the combination of all rewards to form a more generic model.

Parti-1632	Reward	Aesthetics	PickScore	HPSv2
SDXL Base 0.9	-	5.7144	20.466	0.2783
SDv2.0	-	5.1675	18.893	0.2723
SDv1.5	-	5.2634	18.834	0.2703
DDPO [3]	Aesthetic	5.1424	18.790	0.2641
DDPO [3]	Alignment	5.2620	18.707	0.2676
Prompt Engineering	-	5.7062	17.311	0.2599
Direct Fine-tune (Sec. 3.2.1)	All	5.2880	18.750	0.2701
TextCraftor	Aesthetics	5.5212	18.956	0.2670
TextCraftor	PickScore	5.2662	19.023	0.2641
TextCraftor	HPSv2	5.4506	18.922	0.2800
TextCraftor (Text)	All	5.8800	19.157	0.2805
TextCraftor (UNet)	All	6.0062	19.281	0.2867
TextCraftor (Text+UNet)	All	6.4166	19.479	0.2900

Table 2. **Comparison results on HPSv2 benchmark [54].** In addition to the generic model, we report TextCraftor fine-tuned solely on HPSv2 reward, denoted as TextCraftor (HPSv2).

HPS-v2	Animation	Concept Art	Painting	Photo	**Average**
DeepFloyd-XL	0.2764	0.2683	0.2686	0.2775	0.2727
SDXL Base 0.9	0.2842	0.2763	0.2760	0.2729	0.2773
SDv2.0	0.2748	0.2689	0.2686	0.2746	0.2717
SDv1.5	0.2721	0.2653	0.2653	0.2723	0.2688
TextCraftor (HPSv2)	0.2938	0.2919	0.2930	0.2851	0.2910
TextCraftor + UNet (HPSv2)	0.3026	0.2995	0.3005	0.2907	0.2983
TextCraftor (All)	0.2829	0.2800	0.2797	0.2801	0.2807
TextCraftor + UNet (All)	0.2885	0.2845	0.2851	0.2807	0.2847

Table 3. **Human evaluation** on 1632 Parti-Prompts [58]. Human annotators are given two images generated from different approaches and asked to choose the one that has better image quality and text-image alignment. Our approach obtains better human preference over all compared methods.

Comparison Methods	SDv1.5	SDv2.0	SDXL Base 0.9	Prompt Eng.	DDPO Align.	DDPO Aes.
Our Win Rate	71.7%	81.7%	59.7%	81.3%	56.7%	66.2%

4.1. Controllable Generation

Instead of adjusting reward weights γ_i in Eqn. 7, we can alternatively train dedicated text encoders optimized for each reward, and mix-and-match them in the inference phase for flexible and controllable generation.

Interpolation. We demonstrate that, besides quality en-

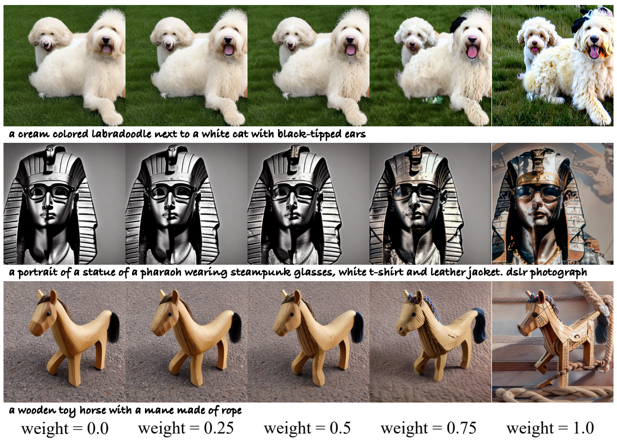

Figure 4. **Interpolation** between original text embedding (weight 0.0) and the one from TextCraftor (weight 1.0) , demonstrating controllable generation. *From top to bottom row*: TextCraftor using HPSv2, PickScore, and Aesthetics as reward models.

Figure 5. **Style mixing.** Text encoders fine-tuned from different reward models can collaborate and serve as style mixing. The weights listed at the bottom are used for combining text embedding from {origin, Aesthetics, PickScore, HPSv2}, respectively.

hancements, TextCraftor can be weighted and interpolated with original text embeddings to control the generative strength. As in Fig. 4, with the increasing weights of enhanced text embeddings, the generated image gradually transforms into the reward-enhanced style. **Style Mixing.** We also show that different reward-finetuned models can collaborate together to form style mixing, as in Fig. 5.

4.2. Ablation Analysis

Rewards and CLIP Constraint. We observe that simply relying on some reward functions might cause mode collapse problems. As in Fig. 6, training solely on Aesthetics score or PickScore obtains exceptional rewards, but the model loses its generality and tends to generate a specific

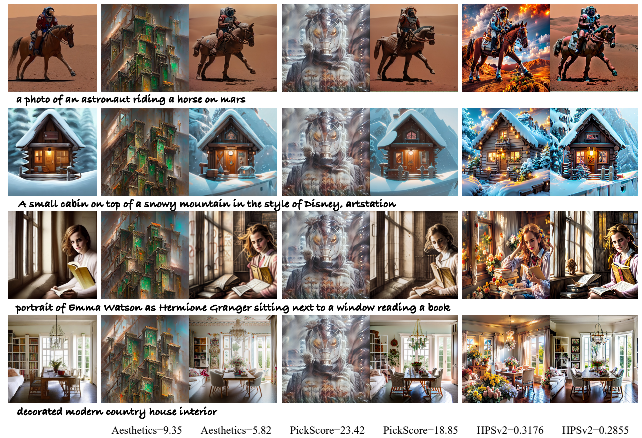

Figure 6. **Ablation on reward models and the effect of CLIP constraint.** The *leftmost* column shows original images. Their averaged Aesthetics, PickScore, and HPSv2 scores are 5.49, 18.19, and 0.2672, respectively. For the following columns, we show the synthesized images *without* and *with* CLIP constraint using different reward models. The reward scores are listed at the bottom.

image that the reward model prefers. To conclude the root cause, not all reward models are pre-trained with large-scale fine-labeled data, thus lacking the capability to justify various prompts and scenarios. We see that HPSv2 shows better generality. Nevertheless, the CLIP constraint prevents the model from collapsing in all three reward regimes, while with reliable improvements in the corresponding scores.

We include more ablation studies on denoising scheduler and steps, reward weights, training and testing steps for TextCraftor , and discussions on training cost and data in supplementary material.

5. Conclusion

In this work, we propose TextCraftor, a stable and powerful framework to fine-tune the pre-trained text encoder to improve the text-to-image generation. With only prompt dataset and pre-defined reward functions, TextCraftor can significantly enhance the generative quality compared to the pre-trained text-to-image models, reinforcement learning-based approach, and prompt engineering. To stabilize the reward fine-tuning process and avoid mode collapse, we introduce a novel similarity-constrained paradigm. We demonstrate the superior advantages of TextCraftor in different datasets, automatic metrics, and human evaluation. Moreover, we can fine-tune the UNet model in our reward pipeline to further improve synthesized images. Given the superiority of our approach, an interesting future direction is to explore encoding the style from reward functions into specific tokens of the text encoder.

References

[1] Clip+mlp aesthetic score predictor. https://github.com/christophschuhmann/improved-aesthetic-predictor#clipmlp-aesthetic-score-predictor, 2022. 2, 3, 5

[2] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 2

[3] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023. 2, 5, 7

[4] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 1

[5] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023. 1, 2

[6] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016. 4

[7] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022. 1, 2

[8] Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023. 2, 4, 5

[9] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, et al. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*, 2023. 2

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 1

[11] Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023. 2

[12] 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. *arXiv preprint arXiv:2305.16381*, 2023. 2

[13] Jindong Gu, Zhen Han, Shuo Chen, Ahmad Beirami, Bailan He, Gengyuan Zhang, Ruotong Liao, Yao Qin, Volker Tresp, and Philip Torr. A systematic survey of prompt engineering on vision-language foundation models. *arXiv preprint arXiv:2307.12980*, 2023. 1, 2, 5

[14] Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. Optimizing prompts for text-to-image generation. *arXiv preprint arXiv:2212.09611*, 2022. 3

[15] Yingqing He, Shaoshu Yang, Haoxin Chen, Xiaodong Cun, Menghan Xia, Yong Zhang, Xintao Wang, Ran He, Qifeng Chen, and Ying Shan. Scalecrafter: Tuning-free higher-resolution visual generation with diffusion models. *arXiv preprint arXiv:2310.07702*, 2023. 2

[16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 3

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1, 2, 3

[18] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 1

[19] 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. *arXiv preprint arXiv:2106.09685*, 2021. 2

[20] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10124–10134, 2023. 2

[21] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022. 2

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[23] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *arXiv preprint arXiv:2305.01569*, 2023. 2, 4, 5

[24] 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. *arXiv preprint arXiv:2302.12192*, 2023. 2

[25] Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. *arXiv preprint arXiv:2306.00980*, 2023. 1

[26] Xihui Liu, Dong Huk Park, Samaneh Azadi, Gong Zhang, Arman Chopikyan, Yuxiao Hu, Humphrey Shi, Anna Rohrbach, and Trevor Darrell. More control for free! image synthesis with semantic diffusion guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 289–299, 2023. 4

[27] Xian Liu, Jian Ren, Aliaksandr Siarohin, Ivan Skorokhodov, Yanyu Li, Dahua Lin, Xihui Liu, Ziwei Liu, and Sergey Tulyakov. Hyperhuman: Hyper-realistic human generation with latent structural diffusion. *arXiv preprint arXiv:2310.08579*, 2023. 1

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[29] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 1

[30] Jacob Menick and Nal Kalchbrenner. Generating high fidelity images with subscale pixel networks and multidimensional upscaling. *arXiv preprint arXiv:1812.01608*, 2018. 2

[31] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 1

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[33] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 2, 5

[34] Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. *arXiv preprint arXiv:2310.03739*, 2023. 2, 5

[35] Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with" gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*, 2023. 3

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 5

[37] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 2

[38] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[39] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014. 3

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1, 2, 3, 5

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICAI*, 2015. 3

[42] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. 1

[43] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1, 2

[44] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1

[45] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022. 3

[46] Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. Freeu: Free lunch in diffusion u-net. *arXiv preprint arXiv:2309.11497*, 2023. 2

[47] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 1

[48] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 2, 3

[49] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3

[50] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 2

[51] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1, 2, 3

[52] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 3

[53] Sam Witteveen and Martin Andrews. Investigating prompt engineering in diffusion models. *arXiv preprint arXiv:2211.15462*, 2022. 1, 2, 5

[54] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. 2, 4, 5, 7

[55] Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2096–2105, 2023. 2

[56] 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. 2, 5

[57] Zeyue Xue, Guanglu Song, Qiushan Guo, Boxiao Liu, Zhuofan Zong, Yu Liu, and Ping Luo. Raphael: Text-to-image generation via large mixture of diffusion paths. *arXiv preprint arXiv:2305.18295*, 2023. 1, 2

[58] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022. 2, 7

[59] Zhixing Zhang, Ligong Han, Arnab Ghosh, Dimitris N Metaxas, and Jian Ren. Sine: Single image editing with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6027–6037, 2023. 1

[60] Shanshan Zhong, Zhongzhan Huang, Weushao Wen, Jinghui Qin, and Liang Lin. Sur-adapter: Enhancing text-to-image pre-trained diffusion models with large language models. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 567–578, 2023. 3