# *DiffChat*: Learning to Chat with Text-to-Image Synthesis Models for Interactive Image Creation

**Jiapeng Wang**[1]*, **Chengyu Wang**[2]†, **Tingfeng Cao**[1], **Jun Huang**[2], **Lianwen Jin**[1]†

[1] South China University of Technology, China

[2] Alibaba Group, China

{eejpwang, setingfengcao}@mail.scut.edu.cn, eelwjin@scut.edu.cn

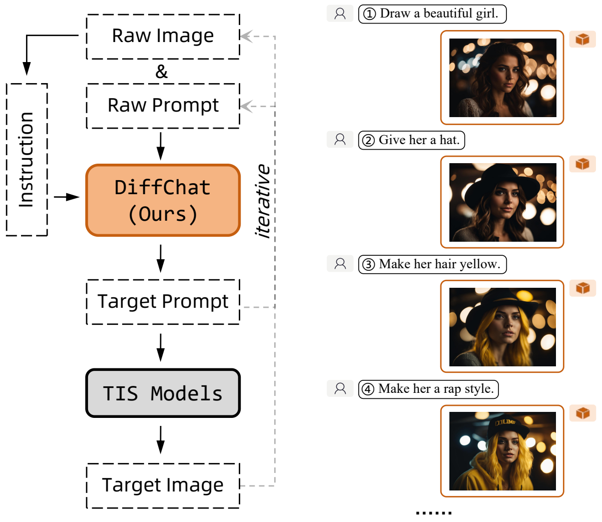
{chengyu.wcy, huangjun.hj}@alibaba-inc.com

## Abstract

We present *DiffChat*, a novel method to align Large Language Models (LLMs) to "chat" with prompt-as-input Text-to-Image Synthesis (TIS) models (e.g., Stable Diffusion) for interactive image creation. Given a raw prompt/image and a user-specified instruction, *DiffChat* can effectively make appropriate modifications and generate the target prompt, which can be leveraged to create the target image of high quality. To achieve this, we first collect an instruction-following prompt engineering dataset named InstructPE for the supervised training of *DiffChat*. Next, we propose a reinforcement learning framework with the feedback of three core criteria for image creation, i.e., aesthetics, user preference and content integrity. It involves an action-space dynamic modification technique to obtain more relevant positive samples and harder negative samples during the off-policy sampling. Content integrity is also introduced into the value estimation function for further improvement of produced images. Our method can exhibit superior performance than baseline models and strong competitors based on both automatic and human evaluations, which fully demonstrates its effectiveness. [1]

## 1 Introduction

In recent years, large-scale deep generative models have emerged as powerful tools for generating contents across various modalities. One of the most remarkably developed and extensively adopted applications is Text-to-Image Synthesis (TIS), which aims to create realistic images with texts as inputs (*prompts*). Large pre-trained TIS models (Ramesh et al., 2021, 2022; Rombach et al., 2022; Saharia et al., 2022; Zhang and Agrawala, 2023; Avrahami et al., 2023; Wang et al., 2023a) have achieved significant improvement to enable users to create images of unprecedentedly high quality, even without art expertise.



(a) (b)

Figure 1: (a) The pipeline of our *DiffChat* collaborating with off-the-shelf TIS models for interactive image iteration. (b) A simple example of *DiffChat* following instructions to interact with TIS models (Stable Diffusion XL here) for interactive image creation. Note that *DiffChat* is capable of automatic prompt refinement and re-writing through "chats" and can be applied to a variety of TIS models.

However, for non-experts, coming up with appropriate and accurate prompts required by TIS models is by no means an easy task. Different language expressions with the same semantics or slightly minor revisions can often result in multiple variations of image generation, which means it is full of uncertainty to write such prompts that meet the user's requirements (Oppenlaender, 2022; Liu and Chilton, 2022). Furthermore, when non-experts wish to create images with specific needs, they often need to iteratively conduct uncertainty trials and errors for prompt refinement, leading to significant losses of time and computing resources. The

*Contribution during internship at Alibaba Group.

†Co-corresponding authors.

[1] InstructPE is available at EasyNLP (Wang et al., 2022). URL: https://github.com/alibaba/EasyNLP

*Findings of the Association for Computational Linguistics: ACL 2024*, pages 8826–8840
August 11-16, 2024 ©2024 Association for Computational Linguistics

capabilities of TIS models are also under-utilized in this case, due to the poorly optimized prompts.

To address these issues, in this paper, we propose a novel framework named *DiffChat*, which can follow user-specified instructions to interact with TIS models for image creation, as shown in Fig. 1. It avoids the tedious attempts of prompt crafting and re-writing mentioned above, making users feel as simple as "chatting" with these TIS models. Specifically, we first collect an **Instruct**ion-following **P**rompt **E**ngineering dataset named **InstructPE** using an automatic data collection pipeline based on existing datasets and AI models. Next, we utilize InstructPE to align off-the-shelf LLMs to adapt to our task with supervised fine-tuning. Finally, we propose an enhanced reinforcement learning framework with three user-concerned criteria for image creation: (1) ***A**esthetics* which represents the aesthetic evaluation of created images; (2) ***P**reference* that indicates the user's preference for specified images relative to other ones; and (3) ***C**ontent integrity* to evaluate whether the creations contain core contents as complete as possible. Thus, *DiffChat* can be trained to pursue positive ***APC*** feedback without any manual labeling. To further enhance the sample quality during training, we also propose an improved sampling strategy based on Action-space Dynamic Modification (ADM). For positive samples, we restrict the generation of tokens with low information quantity to improve the overall quality; and for negative samples, we also partially mask or replace the key information to simulate hard samples, allowing the model to fully learn from these errors. Additionally, we develop the Value estimation function of vanilla PPO (Schulman et al., 2017) method with the consideration of Content Integrity (VCI), to achieve a more accurate perception of the current state during optimization.

By using prompts as intermediaries, *DiffChat* allows users to easily interact and collaborate with TIS models for image creation through chatting. It is worth noting that, recent related research works (Morita et al., 2023a,b; Brooks et al., 2023; Couairon et al., 2023; Zhang et al., 2023; Sun et al., 2023; Koh et al., 2023) mainly focus on the model structure design. They typically adopt instructions, images, and additional information as inputs to generate edited images using an end-to-end network. Different from these works, our method pays more attention to the preliminary automatic prompt writing procedure, which can be used in conjunction with these methods or various widely-used TIS models such as the series of Stable Diffusion (SD) models. It does not need to re-train with the development of TIS models to make extra costs, with its user-friendliness and generalization abilities fully manifested.

Experimental results based on both automatic and human evaluations demonstrate that our method exhibits greater performance than baseline models and competitors. In summary, the main contributions of this paper are listed as follows:

- We propose *DiffChat* to collaborate with TIS models for interactive image creation. It is easy to use and applicable to a wide range of TIS models. Surpassing baselines and competitors also indicates its effectiveness.
- We release a new prompt engineering dataset named InstructPE with 234,786 train and 5,582 test samples for supervised fine-tuning. We further conduct feedback on aesthetics, preference, and content integrity during reinforcement learning. ADM and VCI are also introduced for improved off-policy sampling and state value estimation, respectively.
- The public availability of InstructPE is expected to greatly promote future research on TIS based on instruction-based user-agent interaction.

## 2 Related Work

### 2.1 Text-to-Image Synthesis (TIS) Models

TIS is a multi-modal task involving the generation of images based on textual conditioning. In early years, prevalent research works for TIS primarily relied on the concept of generative adversarial network (GAN) as expounded by (Goodfellow et al., 2014) and (Reed et al., 2016). Recently, diffusion-based models (Ho et al., 2020; Sohl-Dickstein et al., 2015) have emerged as the epitome of excellence in image synthesis endeavors. DALLE-2 (Ramesh et al., 2022) employs a CLIP (Radford et al., 2021a) text embedding to generate an image embedding via a prior network, which is then utilized by a diffusion decoder for image generation. Imagen (Saharia et al., 2022) turns to utilize T5-XXL (Raffel et al., 2020) to produce the text embedding. Moreover, Stable Diffusion (Rombach et al., 2022) trains diffusion models in latent space utilizing a pre-trained auto-encoder.

The qualities of the images generated by these methods are greatly contingent upon the given text prompts. When users are not satisfied with the current results or wish to make modifications, our proposed *DiffChat* serves as a powerful tool to facilitate interactive creation more easily.

### 2.2 Instruction-Following Image Creation

Traditional image editing models mainly targeted a single editing task such as style transfer (Gatys et al., 2016a,b) or translation between image domains (Huang et al., 2018; Isola et al., 2017). With the advent of CLIP and recent diffusion-based models, now users can guide image editing with text instructions (Brooks et al., 2023; Couairon et al., 2023; Zhang et al., 2023; Morita et al., 2023a,b). (Morita et al., 2023a) focuses on the text-relevant content for manipulation and a super-resolution technique is applied. InstructPix2Pix (Brooks et al., 2023) releases an instruction-following image editing dataset and trains an end-to-end diffusion model. These studies mainly focus on the design of the image generation model structure, and thus *DiffChat* can collaborate with the existing models for better user experience. Lately, GILL (Koh et al., 2023) and Emu (Sun et al., 2023) have introduced diffusion decoders into LLMs for end-to-end image generation. One issue with doing so is that whenever a better image decoder appears, the entire model needs to be re-trained and adapted, which often leads to unignorable costs.

Another alternative route is to design the forms and features of input texts (Hertz et al., 2022; Wang et al., 2023b; Kim et al., 2023; Wei et al., 2023). Prompt-to-Prompt (Hertz et al., 2022) designs several rules to control the text-to-image cross-attention for image editing. However, it cannot be directly performed using instructions. InstructEdit (Wang et al., 2023b) directly utilizes BLIP-2 (Li et al., 2023) and ChatGPT (OpenAI, 2023) to generate the original and target captions. Yet, these texts still differ significantly from high-quality prompts used in real-world scenarios. BeautifulPrompt (Cao et al., 2023) generates high-quality prompts by a language model but is not intended for image editing and conversations. PromptMagician (Feng et al., 2023) constructs a prompt recommendation model that identifies related prompt keywords for recommendations. However, it is unable to explicitly follow user-defined instructions and the recommendations are also restricted by the database. PRedItOR (Ravi et al., 2023) uses simplified target prompts to edit images but our *DiffChat* directly uses user-defined instructions. DialogPaint (Wei et al., 2023) aims to train a language model to create instructions from conversations. Nevertheless, it is not as direct and effective as generating the target prompt by *DiffChat*, which can be immediately used for image creation in collaboration with off-the-shelf TIS models such as the Stable Diffusion series.

## 3 Methodology

The overall framework of our method is composed of three main steps: data collection of InstructPE in Fig. 2, supervised fine-tuning over *DiffChat* and enhanced reinforcement learning with APC feedback in Fig. 3. We first construct the InstructPE dataset from the raw data of InstructPix2Pix (Brooks et al., 2023) with prompt beautification and prompt engineering. Next, *DiffChat* is fine-tuned with supervised learning. Finally, an enhanced PPO-based (Schulman et al., 2017) reinforcement learning process is performed with aesthetics, preference, and content integrity criteria. The detailed explanations of each step are as follows.

### 3.1 Data Collection of InstructPE

The goal of the *DiffChat* model is to generate the target prompts for interactive image creation given the raw prompts/images and the user-specified instruction. To achieve it, we first need to build a highly-correlated dataset. InstructPix2Pix (Brooks et al., 2023) has conducted a <*raw prompt*, *instruction*, *target prompt*> format dataset using a fine-tuned GPT-3 (Brown et al., 2020) model. Specifically, it collects a relatively small dataset of editing triplets: input captions, edit instructions, output captions, and fine-tunes GPT-3 for the purpose. The input captions are sampled from the LAION-Aesthetics V2 6.5+ (Schuhmann et al., 2022) dataset, and instructions and output captions are manually written. After this, a large number of new input captions are fed to the trained GPT-3 to generate instructions and output captions, resulting in the final 454,445 examples. However, these input and output captions are simplified and user-friendly, which differ greatly from the effective and high-quality model-friendly prompts in practical applications with detailed descriptions and tags (examples can be found in Appendix A.1).

Realizing this, we first create a prompt beautification (PB) model to solve the problem. We collect a large amount of real-world high-quality

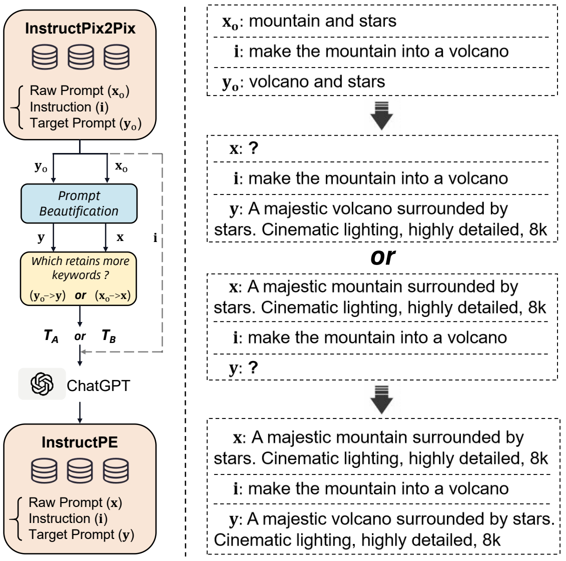

InstructPix2Pix
Raw Prompt ($\mathbf{x_o}$)
Instruction ($\mathbf{i}$)
Target Prompt ($\mathbf{y_o}$)
$\mathbf{y_o}$
$\mathbf{x_o}$
Prompt Beautification
$\mathbf{y}$
$\mathbf{x}$
$\mathbf{i}$
Which retains more keywords ?
($\mathbf{y_o}$->$\mathbf{y}$) or ($\mathbf{x_o}$->$\mathbf{x}$)
$T_A$ or $T_B$
ChatGPT
InstructPE
Raw Prompt ($\mathbf{x}$)
Instruction ($\mathbf{i}$)
Target Prompt ($\mathbf{y}$)
$\mathbf{x_o}$: mountain and stars
$\mathbf{i}$: make the mountain into a volcano
$\mathbf{y_o}$: volcano and stars
$\mathbf{x}$: ?
$\mathbf{i}$: make the mountain into a volcano
$\mathbf{y}$: A majestic volcano surrounded by stars. Cinematic lighting, highly detailed, 8k
$T_A$
or
$\mathbf{x}$: A majestic mountain surrounded by stars. Cinematic lighting, highly detailed, 8k
$\mathbf{i}$: make the mountain into a volcano
$\mathbf{y}$: ?
$T_B$
$\mathbf{x}$: A majestic mountain surrounded by stars. Cinematic lighting, highly detailed, 8k
$\mathbf{i}$: make the mountain into a volcano
$\mathbf{y}$: A majestic volcano surrounded by stars. Cinematic lighting, highly detailed, 8k

Figure 2: Data collection process of **InstructPE**.

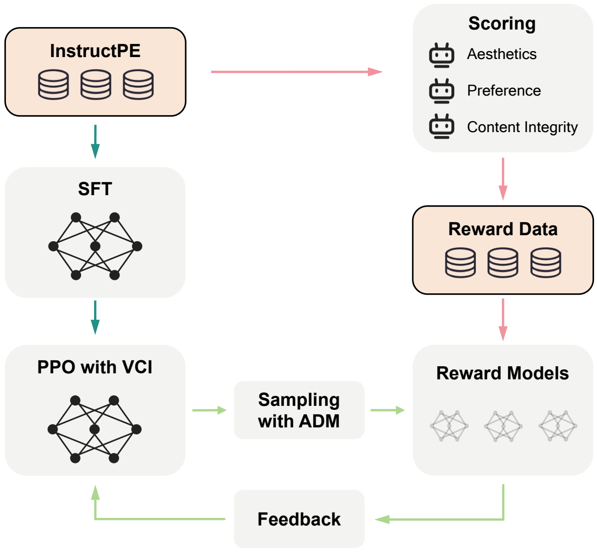


Figure 3: The training procedure of *DiffChat*.

prompts from DiffusionDB[2], MagicPrompt[3], and the Civitai website[4]. Next, we ask ChatGPT to summarize these high-quality prompts into simplified user-friendly prompts[5] (such as the captions in the InstructPix2Pix data as shown in Fig. 2). Through this approach, we have obtained a plentiful supply of <*simplified*, *high-quality*> prompt pairs, which will be used to fine-tune a BLOOM-1.1B (Scao et al., 2022) model as our PB model.

After obtaining the PB model, the collection process of InstructPE is allowed to begin, as shown in Fig. 2. The raw prompt $\mathbf{x_o}$ and target prompt $\mathbf{y_o}$ of InstructPix2Pix are separately sent to our PB model to generate the beautified $\mathbf{x}$ and $\mathbf{y}$. Next, due to the inevitable risk of missing keywords during the generation process of the PB model, we decide whether to use template $T_A$ or $T_B$ for the next step of interaction with ChatGPT based on which group ($\mathbf{y_o}$ –> $\mathbf{y}$) or ($\mathbf{x_o}$ –> $\mathbf{x}$) retains more keywords (the keywords extraction process is shown as the function in Line 10 of Lst. 1 in Appendix A.4). The reason behind this operation is that we hope to maintain consistency between the modified prompts and the original prompts as much as possible. For example, if ($\mathbf{y_o}$ –> $\mathbf{y}$) retains more keywords than ($\mathbf{x_o}$ –> $\mathbf{x}$), we will set $\mathbf{y}$ as the known reference and let ChatGPT generate $\mathbf{x}$, and vice versa. Then, given $\mathbf{x}$ or $\mathbf{y}$ and the instruction $\mathbf{i}$, we ask ChatGPT to write another one ($\mathbf{y}$ or $\mathbf{x}$) with prompt engineering[6]. Finally, our InstructPE dataset is organized as ($\mathbf{x}$, $\mathbf{i}$, $\mathbf{y}$).

Moreover, post-processing is also involved. Non-English and NSFW (not safe for work) examples are first filtered out. Next, we utilize the scoring models that will be discussed in Sec. 3.3 to filter out the low-quality examples. We finally collect 234,786 triplets as our training set and 5,582 triplets as the testing set.

### 3.2 Supervised Fine-Tuning

Given the InstructPE dataset with triplets $D = \{(\mathbf{x}, \mathbf{i}, \mathbf{y})\}$ containing input prompts $\mathbf{x}$, instructions $\mathbf{i}$, and target prompts $\mathbf{y}$, we fine-tune a decoder-only language model to output each high-quality prompt of tokens $\mathbf{y} = \{y_1, ..., y_t\}$, where $t$ is the length of $\mathbf{y}$. We use the auto-regressive language modeling objective to maximize the following likelihood (Radford et al., 2019):

$$\mathcal{L}^{SFT} = -\sum_t \log P(y_t \mid \mathcal{T}(\mathbf{x}, \mathbf{i}), y_1, ..., y_{t-1}), \tag{1}$$

where $\mathcal{T}$ is a template for organizing $\mathbf{x}$ and $\mathbf{i}$ into a prefix sentence[7].

[2] https://huggingface.co/datasets/poloclub/diffusiondb

[3] https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts

[4] https://civitai.com

[5] The detailed prompts and examples to ask ChatGPT for prompt beautification can be found in Appendix A.2.

[6] The detailed prompts and examples to ask ChatGPT for InstructPE can be found in Appendix A.3.

[7] We set the template $\mathcal{T}$ as “Instruction: Give a description of the image and a modification to generate a drawing prompt.\nInput: {$\mathbf{x}$}\nModification: {$\mathbf{i}$}\nOutput:”.

### 3.3 Reinforcement Learning with Feedback

As the collected dataset inevitably contains noises, e.g., the target prompts do not strictly follow the corresponding input prompts and instructions, the performance of the supervising trained model can be unsatisfactory. To make further development, we aim to follow (Ouyang et al., 2022) to perform the task using reinforcement learning leveraging the proximal policy optimization (PPO) (Schulman et al., 2017) algorithm. Yet before that, we propose the following improvements to adapt it to our task.

**Reward Models** The agent model needs to obtain reward feedback from the environment to update its policy in the desired direction. Focusing on our tasks, rewards must reflect the aspects that users care about the results of interactive image creation. In this regard, we design three user-concerned criteria: (1) *Aesthetics*. It represents the aesthetic evaluation of the created images. (2) *Preference*. It indicates the user's preference for the specified image relative to other ones. (3) *Content integrity*. It evaluates the completeness of the key contents of input prompts and instructions contained in the creations. However, even though using human feedback to meet these standards may often bring promising results (Ouyang et al., 2022), it requires extensive and tedious labor efforts. Instead, Bai et al. (2022) proposes to use AI models to instruct the training of LLMs. Inspired by this, we also aim to use off-the-shelf AI models along with self-designed heuristic rules to automatically score our generated results, thus avoiding the cost of expensive human labeling. Specifically, the aesthetic score (Schuhmann et al., 2022) and PickScore (Kirstain et al., 2023) are considered as our *aesthetics* and *preference* criteria, respectively. Moreover, we design the *content integrity* score: Given $(\mathbf{x_o}, \mathbf{i}, \mathbf{y_o})$ as references, it first heuristically extracts keywords from $\mathbf{y_o}$ and identifies the highlighted ones. Then it determines whether to reward based on whether the content integrity of $\mathbf{y}$ reaches a threshold. A code example in Python style is shown in Appendix A.4.

**Action-space Dynamic Modification (ADM)** The action spaces involved in language generation often far surpass the capabilities of most discrete action spaces in traditional designs (Mnih et al., 2015; Hessel et al., 2018). For instance, GPT-3 (Brown et al., 2020) and T5 (Raffel et al., 2020) models have vocabulary sizes of 50K and 32K respectively. During vanilla off-policy data sampling $\pi^{\mathrm{o}}$, it randomly selects the next token $y_t$ based on the probability distribution over the entire action/vocabulary space $\mathcal{Y}$:

$$y_t \sim \pi^{\mathrm{o}}(\cdot \mid P(\mathcal{Y} \mid \mathcal{T}(\mathbf{x}, \mathbf{i}), y_1, ..., y_{t-1})). \quad (2)$$

In this regard, every token with a non-zero probability has a chance of being selected. However, the large size of the action space is a fundamental reason for the instability of sample qualities. To tackle this, we introduce action-space dynamic modification (ADM) to simultaneously refine both positive and negative samples. For positive ones, we exclude tokens with less information quantity from the action space $\mathcal{Y}$ to form $\widehat{\mathcal{Y}}_t^+$ during each sampling step $t$:

$$y_t \sim \pi^{+}(\cdot \mid P(\widehat{\mathcal{Y}}_t^+ \mid \mathcal{T}(\mathbf{x}, \mathbf{i}), y_1, ..., y_{t-1})), \quad (3)$$

where achieving $\widehat{\mathcal{Y}}_t^+$ involves employing locally typical sampling (Meister et al., 2023) with probability $p$, which restricts tokens to the smallest set while ensuring the sum of their probabilities surpasses the specified probability parameter $p$. For negative samples, we conduct that:

$$y_t \sim \pi^{-}(\cdot \mid P(\widehat{\mathcal{Y}}_t^- \mid \mathcal{T}(\mathbf{x}, \mathbf{i}), y_1, ..., y_{t-1})), \quad (4)$$

where we randomly select keywords (from the results of the function in Line 10 of Lst. 1 in Appendix A.4) for each of a small proportion of (with a 4% probability) target prompts and then remove (with a 50% probability) or modify (with a 50% probability) it to create $\widehat{\mathcal{Y}}_t^-$. For the modification, given the original keyword, we select a keyword with the same part of speech but not in its synonym dictionary (Line 36 of Lst. 1) in the training set. In this way, we can simulate the real omission or replacement errors to enable targeted optimization.

**Value Estimation with Content Integrity (VCI)** Advanced policy gradient methods (Schulman et al., 2015a, 2017) introduce the advantage function $A$ to measure the extent to which an action $a_t$ is better or worse than the policy's average action in a particular state $s_t$. Generalized advantage estimation (GAE) (Schulman et al., 2015b) is widely adopted to calculate $A$ as:

$$A(s_t, a_t) = \sum_l (\gamma\lambda)^l \delta_{t+l}, \quad (5)$$

$$\text{where} \quad \delta_t = r_t + \gamma V(s_{t+1}) - V(s_t). \quad (6)$$

Here, $l$ is the trajectory length, $\lambda$ and $\gamma$ are trade-off and discount parameters. $r_t$ is the reward in $t$-th

| Method | PickScore ↑ | Aes. Score ↑ | HPS ↑ | CLIP-S ↑ | D-CLIP-S ↑ | CI Score ↑ | Avg. Rank. ↓ |
|---|---|---|---|---|---|---|---|
| ChatGPT | 19.569 | 6.394 | 20.822 | 29.354 | 15.802 | **87.496** | 2.500 |
| InstructPix2Pix | 19.346 | 5.726 | 19.036 | 22.670 | **17.327** | - | 3.400 |
| *DiffChat* (SFT only) | 19.571 | 6.392 | 20.836 | 29.350 | 16.596 | 85.089 | 2.667 |
| *DiffChat* (full imp.) | **19.584** | **6.416** | **20.851** | **29.397** | 16.787 | 87.314 | **1.333** |

Table 1: Average automatic evaluation results on the InstructPE testing set with different SD models. More detailed results are shown in Appendix A.7. Avg. Rank. is calculated as the average ranking value under each score. Aes. Score: the aesthetic score. CLIP-S: CLIP score. D-CLIP-S: directional CLIP similarity. SFT only: only conducting supervised fine-tuning. Full imp.: Full implementation.

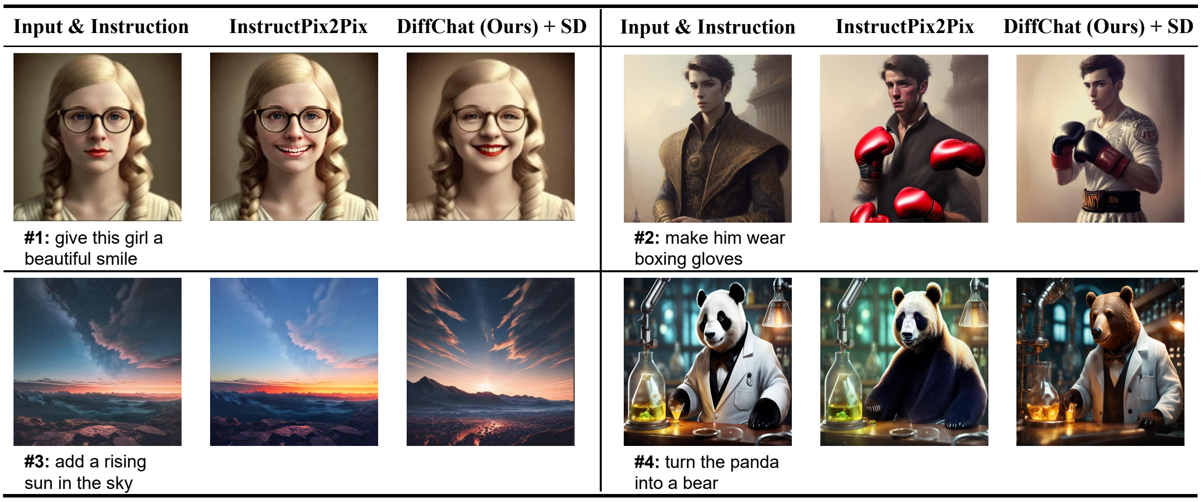

Input & Instruction
InstructPix2Pix
DiffChat (Ours) + SD
Input & Instruction
InstructPix2Pix
DiffChat (Ours) + SD
#1: give this girl a beautiful smile
#2: make him wear boxing gloves
#3: add a rising sun in the sky
#4: turn the panda into a bear

Figure 4: Qualitative results of InstructPix2Pix and *DiffChat* + SD for instruction-following image creation.

step. $V$ is the value function to comprehensively evaluate the current state. In order to help it better perceive the current text generation progress, we propose to add content integrity to compute the value for reinforcement learning:

$$\widehat{V}(s_t) = V(s_t) + \alpha \cdot \text{CI\_score}(s_t). \quad (7)$$

$\alpha$ is a trade-off hyper-parameter. CI_score has been introduced and defined in Lst. 1, and $s_t$ is formulated as $(\mathbf{x_o}, \mathbf{i}, \mathbf{y_o}, \mathbf{y}_{\sim t} = \{y_1, ..., y_t\})$.

## 4 Experiments

### 4.1 Implementation Details

**Training Settings** We use the pre-trained checkpoint of BLOOM (Scao et al., 2022) 1.1B parameters with 24 transformer layers as the backbone of *DiffChat*. Note that, choosing this relatively small version is to ensure the high inference efficiency to support real-world applications. Our method is independent of the selection of specific models and we find that the 1.1B model is sufficient to achieve satisfactory performance. The BFLOAT16 format is leveraged to save GPU memory and speed up training. All the experiments are implemented in PyTorch and run on a single server with NVIDIA Tesla A100 GPUs. More detailed parameters are shown in Appendix A.5.

**Evaluation Protocols** Systematically evaluating the goodness of a prompt engineering model is a challenging task. One of the most straightforward methods is to evaluate the images generated by the prompts that models produce. We use Stable Diffusion 1.5[8], Deliberate[9], Dreamlike[10], Realistic[11], and Stable Diffusion XL 1.0[12] with fixed seeds to generate images and calculate PickScore (Kirstain et al., 2023), the aesthetic score (Schuhmann et al., 2022), HPS (Wu et al., 2023), CLIP score (Radford et al., 2021b), directional CLIP similarity (Gal et al., 2022) and our CI Score before thresholding for the images and the corresponding prompts. Furthermore, we also conduct human evaluations on 100 randomly selected examples from the testing set and 100 randomly user-written examples. Given the raw images and instructions, we ask human ex-

[8] https://huggingface.co/runwayml/stable-diffusion-v1-5

[9] https://huggingface.co/XpucT/Deliberate

[10] https://huggingface.co/dreamlike-art/dreamlike-photoreal-2.0

[11] https://huggingface.co/SG161222/Realistic_Vision_V1.4

[12] https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0

| # | Raw Prompt | Instruction | DiffChat w/o. $\pi^-$ | DiffChat |
|---|---|---|---|---|
| 1 | A digital painting of a landscape of ..., highly detailed | make it a painting | A digital painting of a landscape of ..., highly detailed | A painting of a landscape of ..., highly detailed |
| 2 | Photograph of a wedding. ... prewedding photorealistic | have a vintage feel | Photograph of a vintage wedding. ... prewedding, photorealistic | Photograph of a vintage wedding. ... prewedding, with a vintage feel |

(a) Ablation study for ADM $\pi^-$.

| # | Raw Prompt | Instruction | DiffChat w/o. VCI | DiffChat |
|---|---|---|---|---|
| 1 | study of fair hair beauty ..., full hd | turn the woman into a cat | study of fair hair beauty ..., full hd, cat | study of fair hair cat ..., full hd |
| 2 | A wildflower ... in washington D. c, by Greg Rutkowski | have it be in the snow | A wildflower ... in washington D. c, by Greg Rutkowski, in the snow | A wildflower ... in washington D. c, covered in snow, by Greg Rutkowski |

(b) Ablation study for VCI.

Table 2: Ablation study results for ADM $\pi^-$ and VCI.

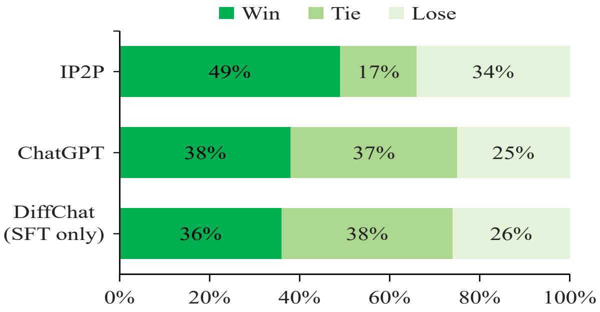


Figure 5: Results of human preference evaluation (i.e., Win/Tie/Lose rates of our method against others). IP2P is short for InstructPix2Pix.

perts to pick the most desirable target images[13] generated by the different methods and report the win rates of *DiffChat*.

## 4.2 Overall Performance

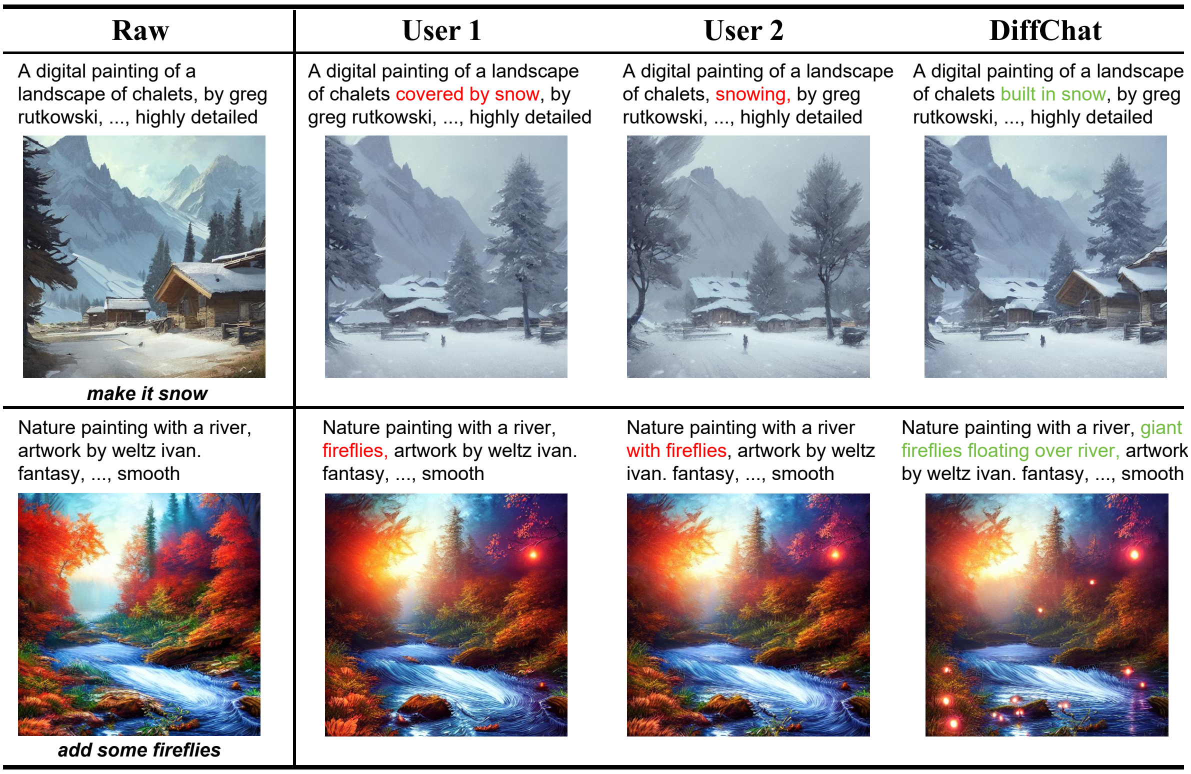
**Competitors** We consider two strong competitors: ChatGPT (OpenAI, 2023) and InstructPix2Pix (Brooks et al., 2023). ChatGPT is almost the most powerful general-purpose LLM with astonishing few-shot learning abilities. Here, it serves as a prompt modifier given the raw prompt and instruction. InstructPix2Pix is a popular end-to-end instruction-following image editing model trained from self-collected data.

From Tab. 1 with the average automatic evaluation results on different SD models, our method consistently achieves competitive or superior performances in most scores. As D-CLIP-S reflects how much the change in text prompts agrees with the change in the images, InstructPix2Pix which directly edits the concerned parts and maintains the overall structure of the image naturally reaches the highest score. We can have an ahead look at example #2 in Fig. 4: InstructPix2Pix mainly adds boxing gloves in the lower local area with an unnatural presentation. On the contrary, our method can add boxing gloves and change it to a boxing posture to fully demonstrate "wear boxing gloves". However, in this case, its D-CLIP-S is still lower than InstructPix2Pix's. CI score mainly reflects the completeness of keywords contained in the target prompt. Although ChatGPT can achieve a relatively high score, it leads to even less beautiful creations. More analyses are shown in Appendix A.7.

As shown in Fig. 5, the human evaluation experiment also indicates the superiority of our approach. *DiffChat* has the highest positive recognition rate among evaluators compared with other models.

Fig. 4 presents the qualitative results generated by InstructPix2Pix and *DiffChat* + SD. For example, given the instruction "give this girl a beautiful smile" in #1, InstructPix2Pix mainly focuses on modifying local areas of the mouth, while ignoring other parts of the face. On the contrary, *DiffChat* + SD can achieve muscle modifications in the eyes, cheeks, and mouth areas, resulting in more natural and beautiful image creations. In #3, InstructPix2Pix mainly increases the overall brightness of the image to express "a rising sun", yet *DiffChat* + SD achieves the addition of half a dazzling sun on the horizon to reflect the rising process. From these examples, we can find that *DiffChat* + SD can complete more aesthetically pleasing creations based on instructions with only minor out-of-concern details changing. Compared with models that directly edit images such as InstructPix2Pix, it can avoid the collapse of local areas in the creation.

[13] The user interface is shown in Appendix A.6.



| Raw | User 1 | User 2 | DiffChat |
|---|---|---|---|
| A digital painting of a landscape of chalets, by greg rutkowski, ..., highly detailed | A digital painting of a landscape of chalets covered by snow, by greg rutkowski, ..., highly detailed | A digital painting of a landscape of chalets, snowing, by greg rutkowski, ..., highly detailed | A digital painting of a landscape of chalets built in snow, by greg rutkowski, ..., highly detailed |
| ***make it snow*** | | | |
| Nature painting with a river, artwork by weltz ivan. fantasy, ..., smooth | Nature painting with a river, fireflies, artwork by weltz ivan. fantasy, ..., smooth | Nature painting with a river with fireflies, artwork by weltz ivan. fantasy, ..., smooth | Nature painting with a river, giant fireflies floating over river, artwork by weltz ivan. fantasy, ..., smooth |
| ***add some fireflies*** | | | |

Figure 6: Comparison between *DiffChat* and human prompt writers given the same inputs and instructions.

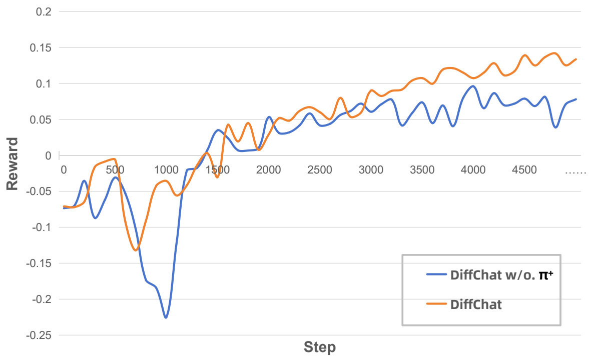


Figure 7: Ablation study for ADM $\pi^+$.

### 4.3 Detailed Analysis

**Ablation Study** From Fig. 7 we can find that ADM $\pi^+$ improves the training stability during optimization and achieves steady higher reward feedback. Furthermore, Tab. 2 (a) shows the improvement brought by ADM $\pi^-$. It helps the model better train with hard negative samples and try to avoid issues such as incorrect replacement and omissions in real applications. From Tab. 2 (b), we can also infer that the vanilla model tends to simply insert the content that needs to be replaced or added at the end of the raw prompts. With the help of VCI, *DiffChat* can be corrected to prefer generating key information content earlier to alleviate it.

**Does *DiffChat* Perform Better than Human Prompt Writers?** We further explore whether *DiffChat* can more effectively bring better image creation experience than users themselves who write prompts completely. For example in Fig. 6 #2, given the raw image of a river, if the users want to *add some fireflies*, they may modify the prompt as "..., fireflies, ..." or "..., with fireflies, ...". However, the resulting images only add a firefly in the top right corner. On the contrary, the modification made by *DiffChat* can lead to a better image creation which is more in line with user expectations.

**Is *DiffChat* Transferable across Different TIS Models?** To verify the transferability of *DiffChat* across different TIS models, we also consider other diffusion-style popular models such as Deliberate, Dreamlike, and Realistic (see them in Appendix A.8). Since our pipeline utilizes prompts as intermediaries for interactive image creation, its flexibility and generalization are guaranteed. More qualitative examples are shown in Appendix A.8.

## 5 Conclusion

In this paper, we propose *DiffChat* to follow user-specified instructions to interact with TIS models for image creation. We collect and release the **InstructPE** dataset for training instruction-following prompt engineering models. A reinforcement learning framework with aesthetics, preference, and content integrity feedback is introduced to align supervising fine-tuned LLMs. Action-space dynamic masking and value estimation with content integrity are also involved for further improvement. Extensive experimental results show that *DiffChat* outperforms competitors in terms of both automatic and human evaluation.

## Limitations

Although *DiffChat* can produce prompts of aesthetically pleasing images given instructions, limited by the training data, it has the risk of ignoring minor parts of the information in the original prompts. Furthermore, since *DiffChat* is guided by the APC feedback as introduced in Sec.3.3 during reinforcement learning, the choices of specific implementation approaches will affect the upper bound of the model performance. These improvements are left to our subsequent work.

## Ethical Considerations

The techniques for training the *DiffChat* model presented in this work are fully methodological, thereby there are no direct negative social impacts of our method. Additionally, we have filtered out NSFW prompts from our training data to ensure that the generated contents are suitable for public distribution. However, given the inherent challenges in controlling the generative process, there is a slight possibility (though improbable) for our model to produce toxic contents. We advise users to refrain from utilizing *DiffChat* intentionally to generate offensive or inappropriate images, and emphasize the need for responsible consideration of potential risks for online deployment.

## Acknowledgements

This research is supported in part by National Natural Science Foundation of China (Grant No.: 62441604, 61936003). This work is partially supported by Alibaba Cloud Group, through Research Talent Program with South China University of Technology.

## References

Omri Avrahami, Ohad Fried, and Dani Lischinski. 2023. Blended latent diffusion. *TOG*, 42(4):1–11.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. 2023. InstructPix2Pix: Learning to follow image editing instructions. In *CVPR*, pages 18392–18402.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *NeurIPS*, 33:1877–1901.

Tingfeng Cao, Chengyu Wang, Bingyan Liu, Ziheng Wu, Jinhui Zhu, and Jun Huang. 2023. Beautifulprompt: Towards automatic prompt engineering for text-to-image synthesis. In *EMNLP*, pages 1–11.

Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. 2023. DiffEdit: Diffusion-based semantic image editing with mask guidance. In *ICLR*.

Yingchaojie Feng, Xingbo Wang, Kam Kwai Wong, Sijia Wang, Yuhong Lu, Minfeng Zhu, Baicheng Wang, and Wei Chen. 2023. Promptmagician: Interactive prompt engineering for text-to-image creation. *IEEE Transactions on Visualization and Computer Graphics*.

Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. 2022. StyleGAN-NADA: CLIP-guided domain adaptation of image generators. *TOG*, 41(4):1–13.

Leon Gatys, Alexander Ecker, and Matthias Bethge. 2016a. A neural algorithm of artistic style. *Journal of Vision*, 16(12):326–326.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016b. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *NeurIPS*, 27.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. 2022. Prompt-to-prompt image editing with cross-attention control. In *ICLR*.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. 2018. Rainbow: Combining improvements in deep reinforcement learning. In *AAAI*, volume 32.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851.

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. 2018. Multimodal unsupervised image-to-image translation. In *ECCV*, pages 172–189.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134.

Sunwoo Kim, Wooseok Jang, Hyunsu Kim, Junho Kim, Yunjey Choi, Seungryong Kim, and Gayeong Lee. 2023. User-friendly image editing with minimal text input: Leveraging captioning and injection techniques. *arXiv preprint arXiv:2306.02717*.

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

Jing Yu Koh, Daniel Fried, and Ruslan Salakhutdinov. 2023. Generating images with multimodal language models. *arXiv preprint arXiv:2305.17216*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Vivian Liu and Lydia B Chilton. 2022. Design guidelines for prompt engineering text-to-image generative models. In *CHI*, pages 1–23.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Ryugo Morita, Zhiqiang Zhang, Man M Ho, and Jinjia Zhou. 2023a. Interactive image manipulation with complex text instructions. In *WACV*, pages 1053–1062.

Ryugo Morita, Zhiqiang Zhang, and Jinjia Zhou. 2023b. BATINeT: Background-aware text to image synthesis and manipulation network. In *ICIP*, pages 765–769.

OpenAI. 2023. ChatGPT. https://openai.com/chatgpt.

Jonas Oppenlaender. 2022. A taxonomy of prompt modifiers for text-to-image generation. *arXiv preprint arXiv:2204.13988*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *NeurIPS*, 35:27730–27744.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021a. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021b. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(1):5485–5551.

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

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *ICML*, pages 8821–8831.

Hareesh Ravi, Sachin Kelkar, Midhun Harikumar, and Ajinkya Kale. 2023. Preditor: Text guided image editing with diffusion prior. *arXiv preprint arXiv:2302.07979*.

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative adversarial text to image synthesis. In *ICML*, pages 1060–1069.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695.

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. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 35:36479–36494.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5B: An open large-scale dataset for training next generation image-text models. *NeurIPS*, 35:25278–25294.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015a. Trust region policy optimization. In *ICML*, pages 1889–1897.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015b. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265.

Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. 2023. Generative pretraining in multimodality. *arXiv preprint arXiv:2307.05222*.

Chengyu Wang, Zhongjie Duan, Bingyan Liu, Xinyi Zou, Cen Chen, Kui Jia, and Jun Huang. 2023a. Paidiffusion: Constructing and serving a family of open chinese diffusion models for text-to-image synthesis on the cloud. *CoRR*, abs/2309.05534.

Chengyu Wang, Minghui Qiu, Taolin Zhang, Tingting Liu, Lei Li, Jianing Wang, Ming Wang, Jun Huang, and Wei Lin. 2022. Easynlp: A comprehensive and easy-to-use toolkit for natural language processing. In *EMNLP*, pages 22–29.

Qian Wang, Biao Zhang, Michael Birsak, and Peter Wonka. 2023b. InstructEdit: Improving automatic masks for diffusion-based image editing with user instructions. *arXiv preprint arXiv:2305.18047*.

Jingxuan Wei, Shiyu Wu, Xin Jiang, and Yequan Wang. 2023. DialogPaint: A dialog-based image editing model. *arXiv preprint arXiv:2303.10073*.

Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. 2023. Better aligning text-to-image models with human preference. *arXiv preprint arXiv:2303.14420*.

Lvmin Zhang and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*.

Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, et al. 2023. HIVE: Harnessing human feedback for instructional visual editing. *arXiv preprint arXiv:2303.09618*.

# A Appendix

## A.1 Examples of User-Friendly and Model-Friendly Prompts

As shown in Fig. 8, the user-friendly prompt is usually simplified and short, while the model-friendly prompt in practical usage is detailed and long with several descriptions and tags. They can lead to image creations with completely different qualities.

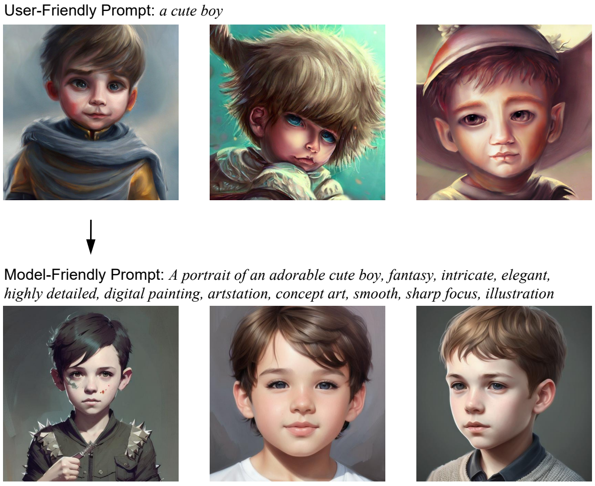

User-Friendly Prompt: *a cute boy*
Model-Friendly Prompt: *A portrait of an adorable cute boy, fantasy, intricate, elegant, highly detailed, digital painting, artstation, concept art, smooth, sharp focus, illustration*

Figure 8: Example of user-friendly and model-friendly prompts.

## A.2 Asking ChatGPT for Prompt Beautification

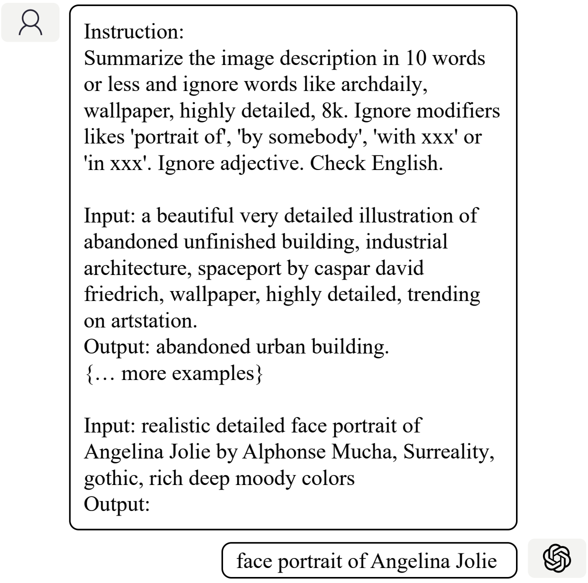
Fig. 9 shows an example of asking ChatGPT to generate data for prompt beautification.

## A.3 Asking ChatGPT for InstructPE

Fig. 10 shows an example of asking ChatGPT to generate data for **InstructPE**.

## A.4 Content Integrity Calculation

A code example in Python style for the calculation of content integrity score is shown in Lst. 1. Some details are abbreviated for better readability.



Instruction:
Summarize the image description in 10 words or less and ignore words like archdaily, wallpaper, highly detailed, 8k. Ignore modifiers likes 'portrait of', 'by somebody', 'with xxx' or 'in xxx'. Ignore adjective. Check English.

Input: a beautiful very detailed illustration of abandoned unfinished building, industrial architecture, spaceport by caspar david friedrich, wallpaper, highly detailed, trending on artstation.
Output: abandoned urban building.
{... more examples}

Input: realistic detailed face portrait of Angelina Jolie by Alphonse Mucha, Surreality, gothic, rich deep moody colors
Output:

face portrait of Angelina Jolie

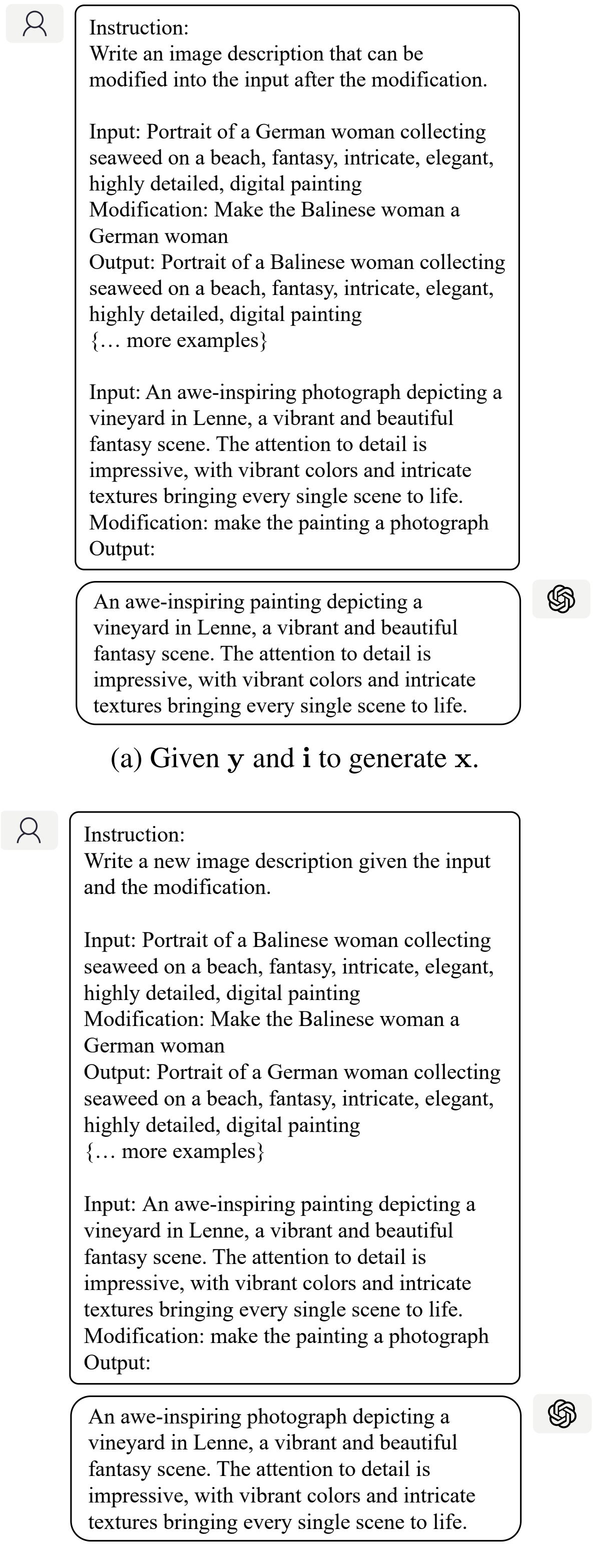
Figure 9: Asking ChatGPT for generating prompt beautification data.

```python
# Pseudocode of CI Score Calculation
import nltk

# Word lemmatization.
def lemmatize(words):
    # "wnl" is a word lemmatization model referenced from https://www.nltk.org/_modules/nltk/stem/wordnet.html.
    return [wnl.lemmatize(w[0], w[1]) for w in words]

# Keywords extraction.
def extract_keywords(text):
    words = nltk.pos_tag(split_to_words(text))
    # "candidates" contains parts of speech that are identified as keywords.
    words = [word for word in words if word[1] in candidates]
    return list(set(lemmatize(words)))

# Input:
# x_o: Raw Prompt of InstructPix2Pix.
#   i: Instruction of InstructPix2Pix.
# y_o: Target Prompt of InstructPix2Pix.
#   y: Target Prompt of InstructPE.
def CI_score(x_o, i, y_o, y, thres=0.7):
    # Split prompt into words and conduct POS tagging.
    y_words = nltk.pos_tag(nltk.split_to_words(y))
    # Word lemmatization.
    y_words = lemmatize(y_words)
    # Extract keywords of x_o, i, and y_o.
    x_o_keywords = extract_keywords(x_o)
    i_keywords   = extract_keywords(i)
    y_o_keywords = extract_keywords(y_o)
    # Identify highlighted words.
```



Instruction:
Write an image description that can be modified into the input after the modification.

Input: Portrait of a German woman collecting seaweed on a beach, fantasy, intricate, elegant, highly detailed, digital painting
Modification: Make the Balinese woman a German woman
Output: Portrait of a Balinese woman collecting seaweed on a beach, fantasy, intricate, elegant, highly detailed, digital painting
{... more examples}

Input: An awe-inspiring photograph depicting a vineyard in Lenne, a vibrant and beautiful fantasy scene. The attention to detail is impressive, with vibrant colors and intricate textures bringing every single scene to life.
Modification: make the painting a photograph
Output:

An awe-inspiring painting depicting a vineyard in Lenne, a vibrant and beautiful fantasy scene. The attention to detail is impressive, with vibrant colors and intricate textures bringing every single scene to life.

(a) Given $\mathbf{y}$ and $\mathbf{i}$ to generate $\mathbf{x}$.

Instruction:
Write a new image description given the input and the modification.

Input: Portrait of a Balinese woman collecting seaweed on a beach, fantasy, intricate, elegant, highly detailed, digital painting
Modification: Make the Balinese woman a German woman
Output: Portrait of a German woman collecting seaweed on a beach, fantasy, intricate, elegant, highly detailed, digital painting
{... more examples}

Input: An awe-inspiring painting depicting a vineyard in Lenne, a vibrant and beautiful fantasy scene. The attention to detail is impressive, with vibrant colors and intricate textures bringing every single scene to life.
Modification: make the painting a photograph
Output:

An awe-inspiring photograph depicting a vineyard in Lenne, a vibrant and beautiful fantasy scene. The attention to detail is impressive, with vibrant colors and intricate textures bringing every single scene to life.

(b) Given $\mathbf{x}$ and $\mathbf{i}$ to generate $\mathbf{y}$.

Figure 10: Asking ChatGPT for generating **InstructPE** data.

```
    highlighted = [q for q in y_o_keywords if q not in x_o_keywords and q in i_keywords]
    # Start calculation.
    cnt = 0
    for y in y_o_keywords:
      # Retrieve in synonym dictionary.
      syns = synonym_dict(y)
      for s in syns:
        if s in y_words:
          if s in highlighted:
            # Give highlighted words greater weight.
              cnt += 2
          else:
              cnt += 1
          break
    CI_score = min(cnt/len(y_o_keywords), 1.)

    # Thresholding.
    return CI_score if CI_score >= thres else 0
```

Listing 1: Pseudocode of content integrity score calculation.

## A.5 More Training Details

Most of the detailed training parameters are listed in Tab. 3. Moreover, for the reinforcement learning, we set the initial KL coefficient as 0.05, $\gamma$ as 0.99, $\lambda$ as 0.95, $p$ as 0.97, and $\alpha$ as 0.05, respectively. We use the AdamW optimizer with eps 1e-8 and $(\beta_1, \beta_2)$ = (0.9, 0.95). Cosine annealing schedule is adopted. Clipping range for PPO policy loss is set as 0.2. Value loss scale w.r.t policy loss is set as 0.5.

| Parameters | SFT | RMs | RL |
|---|---|---|---|
| Epoch | 3 | 1 | 5 |
| Batch Size | 64 | 64 | 128 |
| Maximum Length | 384 | 384 | 384 |
| Learning Rate | 2e-5 | 5e-6 | 5e-6 |
| Unfreezing Layers | All | All | Last 8 layers |
| Weight Decay | 0 | 1e-3 | 1e-6 |

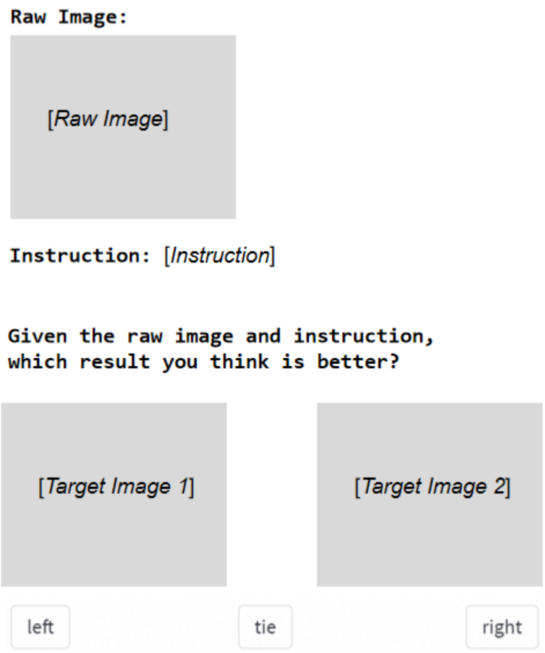
Table 3: Detailed training parameters for SFT (supervised fine-tuning), RMs (reward models), and RL (reinforcement learning) parts.

## A.6 Interface for Human Preference Evaluation

Fig. 11 shows a screenshot of the interface for the human preference evaluation experiment.

## A.7 Detailed Automatic Evaluation Results

In this subsection, we provide the detailed automatic evaluation results as shown in Tab. 4 utilizing different SD models, such as Stable Diffusion 1.5, Deliberate, Dreamlike, Realistic, and Stable Diffusion XL 1.0. It can be found that our method achieves the highest average ranking when collaborating with multiple different SD models. Another noteworthy fact is that InstructPix2Pix is optimized for the D-CLIP-S metric based on Stable Diffusion 1.5 model. When evaluated under other metrics, it performs relatively poorly, and the application on the new Stable Diffusion XL 1.0 leads to significantly poorer results, which also reflects the limitations of its technical route's generalizability.



Raw Image:

Instruction: [Instruction]

Given the raw image and instruction, which result you think is better?

Figure 11: Interface for the human evaluation.

## A.8 Collaborating with various TIS models

Fig. 12 shows some examples of raw images and target images generated with collaboration between *DiffChat* and various Stable Diffusion-style models: Deliberate, Dreamlike, and Realistic. The transferability of *DiffChat* is verified.

| Method | PickScore ↑ | Aes. Score ↑ | HPS ↑ | CLIP-S ↑ | D-CLIP-S ↑ | CI Score ↑ | Avg. Rank. ↓ |
|---|---|---|---|---|---|---|---|
| ChatGPT | 19.338 | 6.145 | 20.123 | **28.837** | 14.594 | **87.496** | 2.500 |
| InstructPix2Pix | 19.235 | 5.917 | 19.430 | 24.580 | **20.818** | - | 3.400 |
| *DiffChat* (SFT only) | 19.339 | 6.149 | 20.129 | 28.769 | 15.532 | 85.089 | 2.500 |
| *DiffChat* (full imp.) | **19.359** | **6.169** | **20.163** | 28.822 | 15.747 | 87.314 | **1.500** |

(a) Stable Diffusion 1.5

| Method | PickScore ↑ | Aes. Score ↑ | HPS ↑ | CLIP-S ↑ | D-CLIP-S ↑ | CI Score ↑ | Avg. Rank. ↓ |
|---|---|---|---|---|---|---|---|
| ChatGPT | 19.602 | 6.501 | 21.010 | 29.692 | 16.863 | **87.496** | 2.333 |
| InstructPix2Pix | 19.349 | 5.977 | 19.672 | 25.175 | **19.309** | - | 3.400 |
| *DiffChat* (SFT only) | 19.602 | 6.487 | **21.042** | 29.686 | 17.592 | 85.089 | 2.500 |
| *DiffChat* (full imp.) | **19.612** | **6.511** | 21.032 | **29.723** | 17.687 | 87.314 | **1.500** |

(b) Dreamlike

| Method | PickScore ↑ | Aes. Score ↑ | HPS ↑ | CLIP-S ↑ | D-CLIP-S ↑ | CI Score ↑ | Avg. Rank. ↓ |
|---|---|---|---|---|---|---|---|
| ChatGPT | 19.494 | 6.254 | 20.586 | 28.863 | 15.258 | **87.496** | 2.500 |
| InstructPix2Pix | 19.313 | 6.052 | 19.784 | 25.072 | **20.790** | - | 3.400 |
| *DiffChat* (SFT only) | 19.500 | 6.249 | 20.589 | 28.857 | 16.018 | 85.089 | 2.667 |
| *DiffChat* (full imp.) | **19.513** | **6.283** | **20.622** | **28.945** | 16.242 | 87.314 | **1.333** |

(c) Realistic

| Method | PickScore ↑ | Aes. Score ↑ | HPS ↑ | CLIP-S ↑ | D-CLIP-S ↑ | CI Score ↑ | Avg. Rank. ↓ |
|---|---|---|---|---|---|---|---|
| ChatGPT | 19.590 | 6.312 | 20.755 | 28.982 | 14.994 | **87.496** | 2.500 |
| InstructPix2Pix | 19.363 | 6.082 | 19.913 | 25.209 | **20.602** | - | 3.400 |
| *DiffChat* (SFT only) | 19.590 | 6.311 | 20.760 | 29.051 | 15.842 | 85.089 | 2.500 |
| *DiffChat* (full imp.) | **19.600** | **6.335** | **20.780** | **29.101** | 16.214 | 87.314 | **1.333** |

(d) Deliberate

| Method | PickScore ↑ | Aes. Score ↑ | HPS ↑ | CLIP-S ↑ | D-CLIP-S ↑ | CI Score ↑ | Avg. Rank. ↓ |
|---|---|---|---|---|---|---|---|
| ChatGPT | 19.820 | 6.757 | 21.636 | **30.395** | 17.300 | **87.496** | 2.333 |
| InstructPix2Pix | 19.468 | 4.603 | 16.381 | 13.313 | 5.117 | - | 4.000 |
| *DiffChat* (SFT only) | 19.823 | 6.762 | **21.659** | 30.383 | 17.992 | 85.089 | 2.167 |
| *DiffChat* (full imp.) | **19.836** | **6.781** | 21.654 | 30.392 | **18.040** | 87.314 | **1.500** |

(e) Stable Diffusion XL 1.0

Table 4: Detailed automatic evaluation results on the **InstructPE** testing set with different SD models. Avg. Rank. is calculated as the average ranking value under each score. Aes. Score: the aesthetic score. CLIP-S: CLIP score. D-CLIP-S: directional CLIP similarity. SFT only: only conducting supervised fine-tuning. Full imp.: Full implementation.

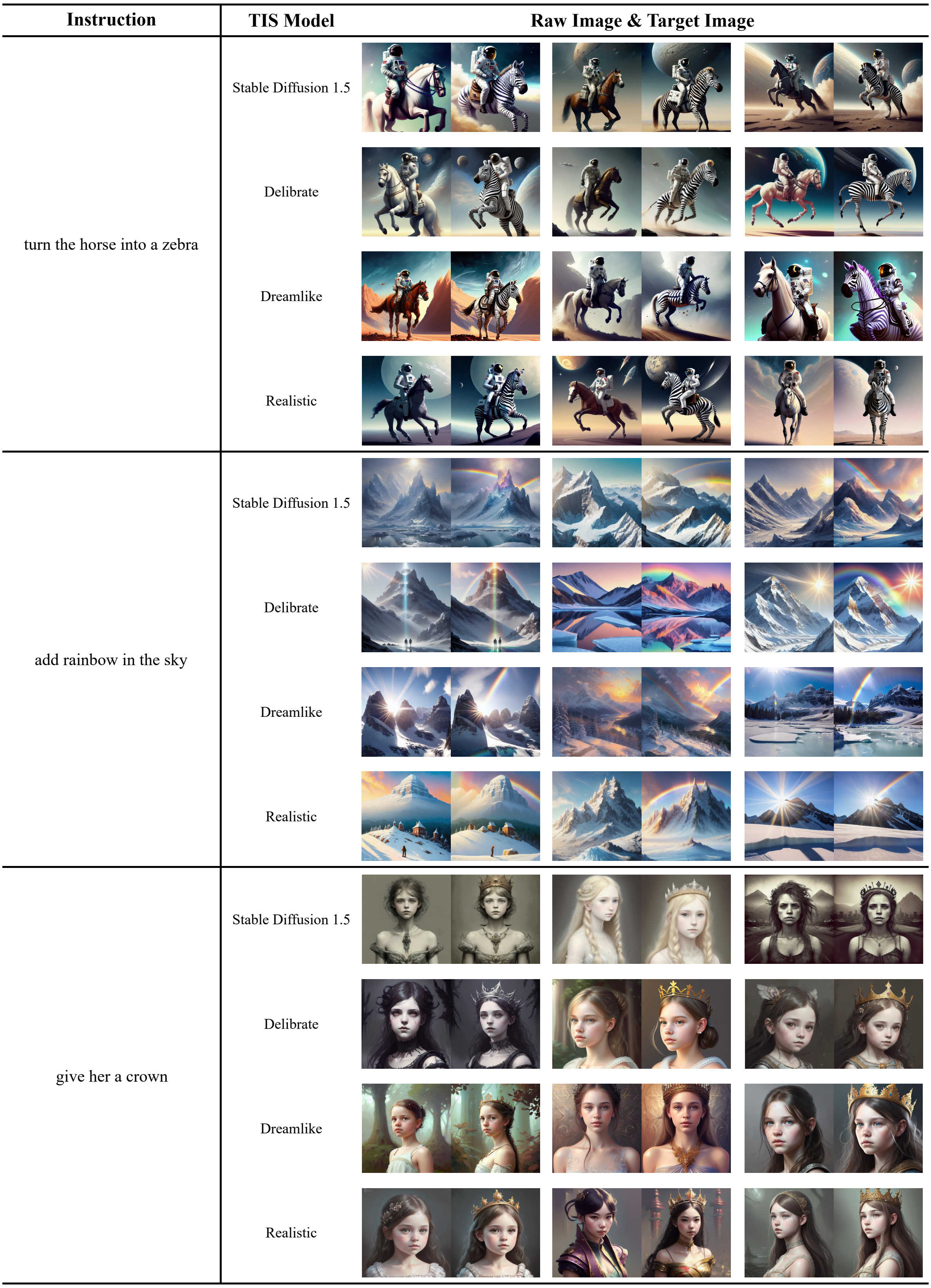


Figure 12: Examples of raw images & target images generated with collaboration between *DiffChat* and various Stable Diffusion-style models.