

# DiffAgent: Fast and Accurate Text-to-Image API Selection with Large Language Model

Lirui Zhao<sup>1,2†</sup> Yue Yang<sup>5,2†</sup> Kaipeng Zhang<sup>2†\*</sup> Wenqi Shao<sup>2†\*</sup>  
 Yuxin Zhang<sup>1</sup> Yu Qiao<sup>2</sup> Ping Luo<sup>2,4</sup> Rongrong Ji<sup>1,3\*</sup>

<sup>1</sup>Key Laboratory of Multimedia Trusted Perception and Efficient Computing,  
 Ministry of Education of China, Xiamen University

<sup>2</sup>OpenGVLab, Shanghai AI Laboratory <sup>4</sup>The University of Hong Kong,

<sup>3</sup>Institute of Artificial Intelligence, Xiamen University <sup>5</sup>Shanghai Jiao Tong University

## Abstract

Text-to-image (T2I) generative models have attracted significant attention and found extensive applications within and beyond academic research. For example, the Civitai community, a platform for T2I innovation, currently hosts an impressive array of 74,492 distinct models. However, this diversity presents a formidable challenge in selecting the most appropriate model and parameters, a process that typically requires numerous trials. Drawing inspiration from the tool usage research of large language models (LLMs), we introduce DiffAgent, an LLM agent designed to screen the accurate selection in seconds via API calls. DiffAgent leverages a novel two-stage training framework, SFTA, enabling it to accurately align T2I API responses with user input in accordance with human preferences. To train and evaluate DiffAgent’s capabilities, we present DABench, a comprehensive dataset encompassing an extensive range of T2I APIs from the community. Our evaluations reveal that DiffAgent not only excels in identifying the appropriate T2I API but also underscores the effectiveness of the SFTA training framework. Codes are available at <https://github.com/OpenGVLab/DiffAgent>.

## 1. Introduction

Recent advancements in text-to-image (T2I) generative models [1, 13, 20, 21, 23] have garnered considerable attention, courtesy of their extraordinary proficiency in following text prompts and the exceptional capability in image generation. As a prominent technique, Stable Diffusion (SD) [21] is widely utilized in various generative applications, and has

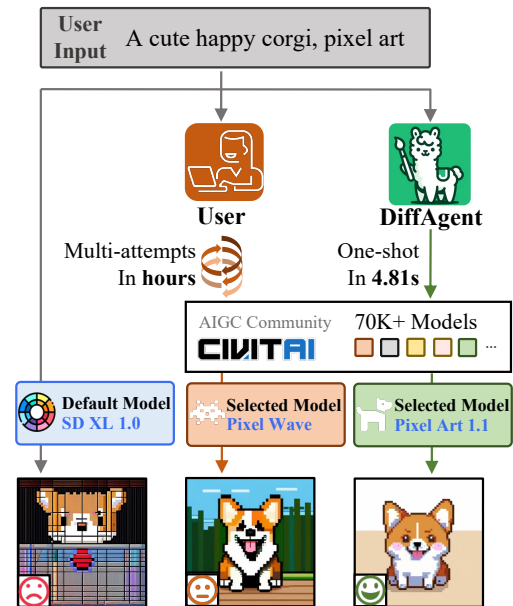


Figure 1. The comparison of different ways to T2I generation.

gained significant traction within the research community and broader adoption.

However, the vanilla SD model falls short in catering to diverse personalized styles such as pixel art. For example, a corgi covered by several messy lines is generated by the SD-XL model [17] with the user’s request of ‘A cute happy corgi, pixel art’, as shown in Fig. 1. Such a limitation has spurred the emergence of various personalized models, which are tailor-made to excel in specific styles. Innovations in lightweight personalization techniques such as Low-Rank Adaption (LoRA) [11] and other approaches [7, 22, 34] have streamlined the process of fine-tuning these models from the original SD framework, making it more cost-efficient. Such advancements foster a wide adoption and influence of customized models in the open-sourced community [4, 6]. For instance, Civitai [4] hosts

\*Corresponding author: rrji@xmu.edu.cn

†Equal contribution \*Project lead

‡This work was done during his internship at Shanghai AI Laboratory.

approximately 74,492 models with various personalized LoRA [11] packages, facilitating itself a vibrant ecosystem for personalized image generation.

Despite the amounts of personalized models, it is hard to obtain the desired images given arbitrary requests from users. Firstly, it often requires multiple attempts to select the appropriate one from massive SD models. Secondly, continuous feedback is needed to modify generation parameters for the final performances. The multiple attempts to select an appropriate model and refinement of input prompts necessitate a meticulous cycle of text-to-image (T2I) generations and continuous feedback adaptation, as shown in Fig. 1. This work targets tackling the challenging problem: *how to quickly screen a good SD model and corresponding parameters for user-specific requests?*

Recent advances in large language models (LLMs) have shown that LLMs fine-tuned with self-instruction data about tool usage exhibit exceptional capability in invoking various tools, serving as powerful agents to call different tools (*e.g.*, APIs) for target tasks. For example, Gorilla [16] can generate proper model API collected from open-sourced model libraries such as HuggingFace [6] to execute various tasks. GPT4Tool [33] enables LLMs to use multimodal tools for tackling a variety of visual problems such as object grounding and segmentation. Motivated by these studies, we turn the challenges of selecting a proper SD model and corresponding parameters for various requests into a process of calling appropriate text-to-image APIs conditioned on the input prompt through LLMs. In this regard, LLMs as API calling tools are expected to comprehend the user’s request and generate a corresponding T2I API for image generation.

To achieve this, we propose LLM as the agent to call text-to-image API involving a stable diffusion model with associated hyperparameters in response to arbitrary user prompts, termed as DiffAgent. DiffAgent incorporates two essential components: an instruction-following dataset called DABench that gathers extensive pairs of user prompts and T2I APIs, and a fine-tuning and alignment training framework for training an LLM as an agent to call a proper T2I API for the user prompt. Compared with previous API calling tools [16, 33], our DiffAgent has two appealing properties. First, DiffAgent can screen the most appropriate T2I generation API for different user requests in the text-to-image task by training the LLM with massive pairs of user prompts and T2I APIs in DABench. While previous approaches [16, 33] provide a limited number of model APIs for every single task. For example, GPT4Tools [33] would always return Stable Diffusion XL as the API when given any text-to-image generation prompts. Second, DiffAgent can generate the T2I API aligned well with human preferences. Other than supervised fine-tuning (SFT) LLM to follow API calling instructions as also done in previous approaches [16, 33], DiffAgent further trains the LLM with

the ranking responses with human feedback (RRHF) algorithm [36], encouraging the LLM to generate responses aligned better with human preference than results from SFT.

Specifically, we first construct, DABench, a large instruction-following dataset for T2I generation API calling by crawling massive pairs of T2I APIs (SD models with corresponding parameters) and user prompts from the Civitai website which is a vibrant community for creators to share various AI-generated art [4]. DABench undergoes rigorous filtering and reconstruction processes. It then incorporates high-quality APIs from two main components of the community, *i.e.* 39,670 and 10,812 pairs of T2I APIs and user prompt for SD 1.5 and SD XL, respectively, ultimately resulting in 50,482 pieces of instruction data for API calls.

By fine-tuning the LLM (instantiated as LLaMA-2-7B [29] in this paper) on the DABench, DiffAgent can generate API calls that correspond to the user prompt. Nevertheless, we find that the supervised fine-tuning on DABench cannot choose the best one in terms of human preference when multiple appropriate T2I APIs are presented for the user prompt. As shown in Fig. 1, for one style (*e.g.*, pixel art), there are several T2I generation APIs that all adopt this style but with different qualities. To address this limitation, we further fine-tune the LLM with the RRHF algorithm [36] by incorporating various image assessment scores aligned with human preference as the reward model function. Our observations indicate that the LLM, fine-tuned using the RRHF, exhibits a decreased occurrence of hallucination errors in its generated APIs (*e.g.*, 20.1% lower than the SFT model). Moreover, the resulting image aligns closely with human preferences. Specifically, when considering the SD 1.5 model architecture, our fine-tuned model demonstrates over a 10% improvement compared to the SFT model in terms of unified metric as presented in Table 2.

This paper’s contributions are summarized as follows:

- We propose DiffAgent, a text-to-image (T2I) agent model that selects appropriate T2I generation information (*i.e.*, T2I API) to free users from tedious attempts and matching for the prompt. The T2I APIs from DiffAgent exhibit a high degree of human preferences and relevance with the prompt.
- We introduced DABench, which is a carefully curated high-quality dataset for T2I API selection. DABench is the first API call dataset designed specifically for T2I domain, containing a large number of callable APIs (a total of 50,482).
- We conduct comprehensive experiments at DiffAgent, encompassing various datasets (DABench, COCO Caption [2], Parti Prompts[35]) and multiple T2I evaluation scores (CLIP Score [9], ImageReward [32], HPS v2 [31]). Our results demonstrate that DiffAgent achieves a substantial performance improvement than using the original T2I model (*e.g.*, 40% in SD 1.5).

## 2. Related Work

**LLMs Tool Usage.** Recent studies [8, 16, 18, 24, 28, 33] have revealed the growing potential of large language models (LLMs) in acquiring proficiency in tool usage and decision-making within intricate environments. [8, 24] enhance the performance of LLMs in targeted domains by using a small set of simple tools (*e.g.*, using calculators for precise mathematical reasoning). Gorilla [16] enables LLMs to respond to API calls by constructing a dataset and performing fine-tuning. ToolLLM [18] further to make LLMs utilize massive APIs. GPT4Tool [33] focuses on using multimodal tools for tackling a variety of visual problems (*e.g.*, segmentation). ToolAlpaca [28] verifies the feasibility of equipping compact LLMs with generalized tool-use capacities. These works have primarily focused on a general range of API calls that only allow for limited and fixed options for a single task. Our work, DiffAgent, aims to select the most suitable API from massive T2I APIs in the text-to-image task. Different further align LLMs with human preferences using the training framework SFTA.

**Text-to-Image Diffusion Models.** Recently, text-to-image (T2I) diffusion models [1, 13, 20], have shown exceptional capability in image generation quality and extraordinary proficiency in following text prompts, under the dual support of large-scale text-image dataset [25] and model optimizations [5, 10, 21]. GLIDE [13] incorporated text conditions into the diffusion model and empirically showed that leveraging classifier guidance leads to visually appealing outcomes. DALLE-2 [20] enhances text-image alignment via CLIP [19] joint feature space. DALLE-3 [1] further improves the prompt following abilities by training on highly descriptive generated image captions. Stable Diffusion [21], a well-established and widely adopted method, has gained substantial traction in the open-source community [4, 6], leading to a large and diverse model library (*e.g.*, 74492 models in Civitai [4]). The cost of generating desired images using specific personalized models also increases especially in such a vast quantity of models with quality varies. DiffAgent can free users from multiple attempts and screen the personalized model and corresponding parameters for matching text prompts in seconds.

**Align Language Models with Human Preferences.** Reinforcement Learning from Human Feedback (RLHF) has been pivotal in aligning LLMs to human preferences, typically employing Proximal Policy Optimization (PPO) algorithm for model optimization [15, 27, 37]. During the optimization process with PPO, the likelihood of a whole generation is required to update LLMs. For LLM agents, however, human feedback is typically available only after acquiring the complete API response and successful function invocation. This implies that 1) the reward model based on human feedback is not capable of getting the likelihood of a whole generation and 2) the receipt of delayed human

feedback necessitates the complete API response for effective invocation. Rank Responses to Align Human Feedback (RRHF) [36] emerges as a notable method, employing reward models to rank multiple responses for language model alignment. This approach facilitates its extension to fine-grained API Agents, maximizing the utility of existing reward models. Nevertheless, it presents challenges such as maintaining the output API’s integrity and evaluating output using the reward model solely on the post-invocation actual results. To address these challenges, we enhance the RRHF method by incorporating evaluation scores into the reward model function, establishing a viable alignment procedure.

## 3. Methodology

Given a text prompt, our objective is to free users from tedious attempts and matching in order to find the appropriate APIs for text-to-image generation (*i.e.*, T2I API selection). We initially introduce DABench for this task, a benchmark derived from Civitai. DABench consists of Instruction-API pairs, each pair including the text prompts and corresponding T2I generation information (*e.g.*, model information and parameters information). We provide an overview of the procedure of the API dataset collection, filtering, and reconstruction as detailed in Sec. 3.1. Subsequently, we introduce DiffAgent, an LLM agent to call T2I APIs in response to arbitrary user prompts. We describe the multi-step training framework, called SFTA, for DiffAgent and the unified metric for evaluating DiffAgent (Sec. 3.2).

### 3.1. DABench

The Civitai community [4] comprises approximately 74,492 models, each typically accompanied by a set of sample images. These images demonstrate the function of the corresponding model, while their prompts show the style and common scenarios that are most suitable for generating. Civitai also exists evaluation mechanisms (download count, rating, *etc.*), statistical results show common preferences of the community. We can acquire hierarchical information, which includes the model’s data and sample images with generation details from uploader, and statistical results from user. Our goal is to collect a high-quality dataset, each data pair consists of Instruction (text prompts), and its API (corresponding model and parameter information). This is achieved through the collection and filtration of model-level data, followed by a reverse mapping with image’s data.

**APIs Collection** We first perform a comprehensive analysis of the composition of Civitai. As shown in Fig. 2, the base model architectures can be categorized into two primary types: SD 1.5 (85%) and SDXL 1.0 (5%). regarding the model types, they consist of LoRA (77%) and Checkpoint (9%). Therefore, we determine the scope of APIs collection to include the above content.

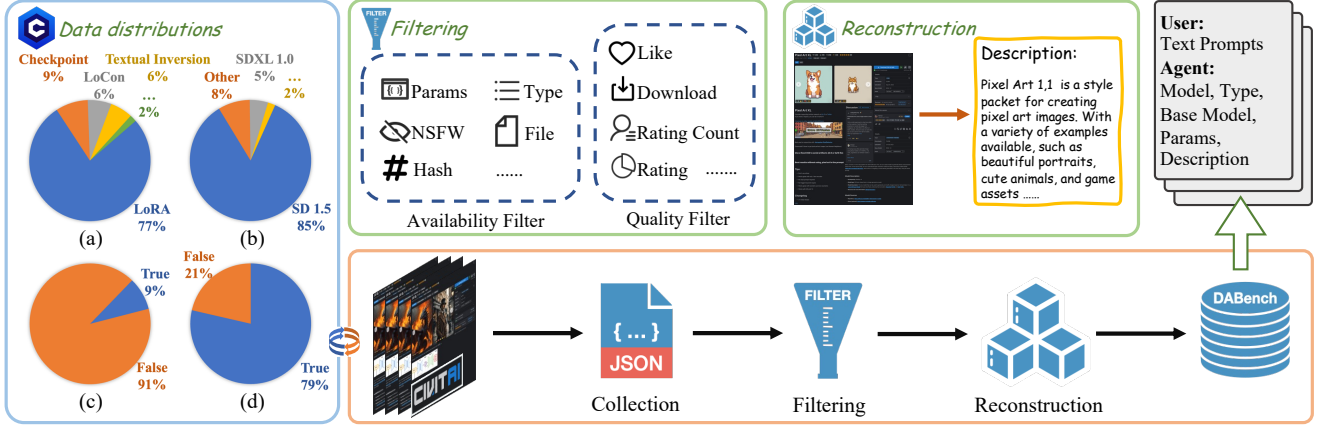


Figure 2. The data collection process of DABench. The left side is the data distributions of the data source, Civitai [4]. These distributions include: (a) Model type. (b) Base model architecture. (c) Not suitable for work (NSFW) content. (d) File availability.

**APIs Filtering** Due to the varying quality of APIs, we employ two multi-attribute filters (availability and quality filters) to choose a wide range of high-quality options.

As depicted in Fig. 2, we initiate the process of verifying the availability of APIs to ensure successful invocation of the collected APIs. The APIs related to not suitable for work (NSFW) content will also be excluded during this stage. As stated in 3.1, the model type (Checkpoint and LoRA) and base model architectures (SDXL 1.0 and SD 1.5) under the scope are reserved and vice versa. During the T2I generation, the LoRA model needs to be used simultaneously with the base model by embedding tags into prompts. However, only the hash of the base model exists in the collected data, and for a LoRA model, there may be different base models present in different sample images. Then we attempt to obtain base model information for LoRA models’ images using the hash and filter out the inaccessible parts. The LoRA tag embedded within their text prompts is subsequently eliminated.

We next perform a quality assessment of these APIs using established evaluation mechanisms (download count, rating count, *etc.*) within the community, showing the common preferences of the users. The process results in the identification of a subset with superior quality

**Description Reconstruction** As an active open-source community, many models lack accurate and relevant descriptions. For instance, the description of the pixel Art model does not include many of the applicable styles for the model itself while we can understand this at a glance through sample images. Hence we employ GPT-3.5-turbo [14] to reconstruct the description of models by incorporating information from both the models themselves and sample images (utilizing their associated prompts) as shown in Fig. 2. This approach aims to improve the consistency between APIs and their corresponding description.

**Dataset composition** We convert the collected APIs’ information into a json object with the following fields: {model, type, base Model, width, height, sampling\_method, sampling\_steps, cfg\_scale, model\_description}. These fields consist of two parts required for T2I generation: model information (model, type, base Model, model\_description) and parameter information (the remaining fields). The collected instructions are also converted into a json object: {prompt, negative\_prompt}.

Finally, DABench includes high-quality APIs from SD 1.5 (39,670 Instruction-API pairs in 5,516 API calls) and SD XL (10,812 in 1,306).

### 3.2. DiffAgent

DiffAgent is an LLM agent fine-tuned through our two-stage training framework, SFTA, to achieve T2I API selection given by text prompt. In the first stage, we employ supervised fine-tuning (SFT) with DABench to train and obtain the preliminary model known as DiffAgent-SFT. In the second stage, we further enhance DiffAgent-SFT through alignment with human preferences as depicted in Fig. 3, to obtain the final DiffAgent model (*i.e.*, DiffAgent-RRHF).

#### 3.2.1 DiffAgent-SFT

We employ the standard fine-tuning to get the DiffAgent-SFT model. For a pair of text prompts  $t$  and API response  $r$ , the cross-entropy loss is used for fine-tuning:

$$L_{sft} = - \sum_t \log P_{\pi}(r_t | t, r_{<t}). \quad (1)$$

DiffAgent-SFT possesses all the necessary conditions for generating images, including text prompts as input along with the corresponding model and parameters as response API. This enables us to generate images directly for the purpose of evaluating the performance of APIs.

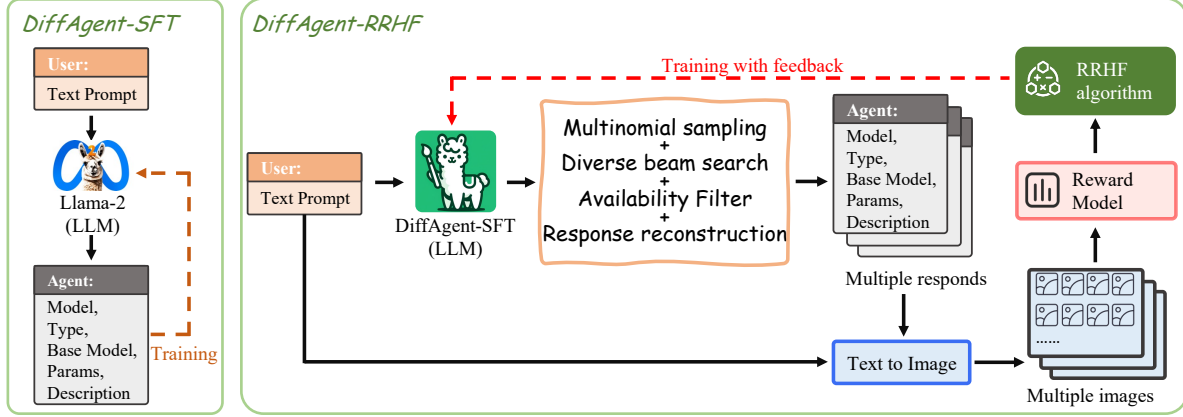


Figure 3. The training framework SFTA to get DiffAgent.

### 3.2.2 DiffAgent-RRHF

We first propose a human preference-related unified metric for evaluating DiffAgent. Then, we introduce an effective training framework to further align DiffAgent-SFT with human preferences.

**Human Preferences Evaluation** In the T2I generation domain, there already exist multiple scoring models or metrics [9, 31, 32], which can be employed for related or human preference evaluation. Given a prompt  $t$  in text prompts and the generated image  $x(r)$  using API response  $r$ , we evaluate DiffAgent using a human preference-related unified metric composed of: *First*, CLIP Score [9] aims to evaluate the correlation between  $t$  and  $x$  using cosine similarity of their respective embedding obtained through CLIP [19]:

$$s_{\theta_c}(t, x(r)) = w * \max(\cos(\text{Enc}_{\text{txt}}(t), \text{Enc}_{\text{img}}(x(r))), 0), \quad (2)$$

where  $w = 2.5$  as stated in [9], and  $\theta_c$  represents the parameters in CLIP (ViT-B for image encoder, 12-layers transformer for text encoder). *Second*, ImageReward [32], a T2I human preference reward model fine-tuned by BLIP [12] (ViT-L for image encoder, 12-layers transformer for image-grounded text encoder), extracts  $t$  and  $x(r)$  features, employs cross attention and an MLP layer to generate a scalar:

$$s_{\theta_i}(t, x(r)) = \text{MLP}(\text{Enc}_{\text{txt}}(t), \text{Enc}_{\text{img}}(x(r))), \quad (3)$$

where  $\theta_i$  represents the parameters of the ImageReward Model. *Third*, Human Preference Score v2 (HPS v2) [31], a scoring model fine-tuned by CLIP-H model, can more accurately predict text-generated images' human preferences:

$$s_{\theta_h}(t, x(r)) = \frac{\text{Enc}_{\text{txt}}(t) \cdot \text{Enc}_{\text{img}}(x(r))}{\tau}, \quad (4)$$

where  $\tau$  represents the learned temperature during HPS v2's fine-tuning process, and  $\theta_h$  denotes its parameters.

Now, we will proceed to the normalization and combination of the scores to obtain a unified metric  $S$  for evaluating APIs. Given a text prompt  $t$  and a set of images  $X(r) = \{x(r)_1, x(r)_2, \dots, x(r)_k\}$ , where each image  $x(r)_i$  generated based on a different random seed to mitigate the influence of random seeds, we compute the scores from three metrics using Eqs. (2) to (4). Each of these scores is normalized to the range  $[0, 1]$ . The final score for the text prompt  $t$  with API response from DiffAgent is obtained by averaging all image's scores:

$$S(t, r) = \frac{1}{k} \sum_{i=1}^k S(t, x(r)_i), \quad (5)$$

s.t.

$$S(t, x(r)_i) = \frac{1}{3} (s_{\theta_c, \text{norm}}(t, x(r)_i) + s_{\theta_i, \text{norm}}(t, x(r)_i) + s_{\theta_h, \text{norm}}(t, x(r)_i)). \quad (6)$$

The Unified Metric  $S(t, r)$  describes the human preference score for an API response  $r$  given a prompt  $t$ . The Eq. (5) serves a dual purpose, as it can be utilized for both API evaluation and as a reward model function in alignment with human preferences.

**Alignment with Human Preferences** The existing Reinforcement Learning from Human Feedback (RLHF) methods [3, 15, 27] is performed using reinforcement learning algorithms (e.g., PPO [26]), which depends on the likelihood of a whole generation to update the model. However, the evaluation way of the agent model necessitates the exclusion of vanilla RLHF methods due to the delayed feedback received only upon API invocation and a complete API response is required during this process.

We propose an extension to the existing RRHF [36] method that can be used for aligning DiffAgent (Fig. 3). For text prompts  $t$ , we employ multinomial sampling to

get a comprehensive API response and utilize diverse beam search [30] to get  $m$  diverse responses from SFT model  $\pi$ . However, it is difficult to ensure the format and validity of multiple different responses. Therefore, we only take the model information part of responses, then proceed to validate and reconstruct the complete API response information. After incorporating the default API response, a list of  $n$  API responses  $R = \{r_1, r_2, \dots, r_n\}$  is generated. Then we actually call APIs to retrieve the images and use Eq. (5) as the scoring function for the reward model to get  $n$  scores for each  $r_i$  with  $S(t, r_i) = s_i$ .

To align with scores  $\{s_i\}_n$ , we use SFT model  $\pi$  to give scores  $p_i$  for each  $r_i$  by:

$$p_i = \frac{\sum_t \log P_\pi(r_{i,t}|t, r_{i,<t})}{\|r_i\|}, \quad (7)$$

where  $p_i$  is conditional log probability (length-normalized) of  $r_i$  under model  $\pi$ . We use the RRHF algorithm to let the model  $\pi$  give larger probabilities for better responses and give smaller probabilities for worse responses. this object is optimized by ranking loss:

$$L_{rank} = \sum_{s_i < s_j} \max(0, p_i - p_j), \quad (8)$$

and a cross-entropy loss like SFT process is added as well:

$$L_{ce} = - \sum_t \log P_\pi(r_{i',t}|t, r_{i',<t}), i' = \arg \max_i s_i. \quad (9)$$

The overall loss is utilized to optimize the DiffAgent-SFT in order to obtain DiffAgent-RRHF:

$$L = L_{rank} + L_{ce}. \quad (10)$$

## 4. Experiment

We performed a comprehensive evaluation of DiffAgent by conducting experiments on three benchmarks: DABench, COCO Caption [2], and Parti Prompts [35]. Our evaluation involved comparing the performance of DiffAgent with other powerful models and investigating the effectiveness of our two-stage training framework. In addition, we show the effectiveness of aligning with human preferences in DiffAgent through comparative experiments. We then conduct a user study to examine DiffAgent’s ability to identify related diverse styles’ APIs, which demonstrated its strong capability. Finally, we showcase visualizations to offer an intuitive assessment and facilitate comparisons.

### 4.1. Experimental Settings

**Training Setup** We transform DABench’s Instruction-API pairs into single-round user-agent dialogues, partitioning them in an 8:1:1 ratio for fine-tuning, aligning with human preferences, and evaluation. For different model architectures (SD 1.5 and SD XL), we employ the SFTA framework to fine-tune LLaMA-2-7b [29], thereby generating the

DiffAgent		SFT	RRHF
SD 1.5	DABench	11.5	<b>9.5</b>
	COCO Caption [2]	26.9	<b>15.5</b>
	Parti Prompts [35]	28.9	<b>20.0</b>
SD XL	DABench	7.4	<b>7.1</b>
	COCO Caption [2]	45.6	<b>36.0</b>
	Parti Prompts [35]	48.1	<b>28.0</b>

Table 1. The hallucination error of DiffAgent in different training stage. DiffAgent-RRHF shows a lower error than DiffAgent-SFT. respective DiffAgents. In the SFTA’s second stage, we assess each response through ten random image generations ( $k = 10$ , as per Eq. (5)).

**Dataset Settings** DiffAgent’s evaluation employs the following datasets: 1) The validation set of DABench. 2) For the COCO Caption and Parti Prompts datasets, we randomly select 1,000 prompts from each for assessment. Additionally, we adapt GPT-3.5-turbo [14] as a text prompt generator, leveraging in-context learning to transform concise prompts into well-structured text prompts.

**Baselines** Our primary comparison involves DiffAgent against the original Stable Diffusion (SD) with a default parameter, maintaining identical architectures. Given that DiffAgent’s T2I APIs incorporate both model and parameter information, we consider combining these separately with the baseline to obtain two additional comparisons. Furthermore, the DiffAgent-SFT, derived from the initial stage of the SFTA process, serves as a significant method in our analysis. As outlined in Sec. 3.2.2, we employ three metrics (CLIP Score [9], ImageReward [32], and HPS v2 [31]) alongside the Unified Metric (Eq. (5)) for evaluation. The Unified Metric requires the normalization of scores across different baselines, a process complicated by potential hallucination issues in DiffAgent’s generated API. Hence, we consider applying the baseline to generate images when the issue of hallucination arises. we standardized the shape of the generated image to be consistent due to the impact of shape on the evaluation. The default parameters are set to {sampling\_method: Euler a, sampling\_steps: 20, cfg\_scale: 7}, which are the most commonly used parameter choices.

## 4.2. Results

### 4.2.1 Hallucination

In Table 1, we present the hallucination error of DiffAgent on various evaluation datasets. It is evident that DiffAgent-RRHF consistently achieves a lower hallucination error compared to DiffAgent-SFT. For example, The hallucination error decreased from 48.1% to 28.0% after optimizing with RRHF on the Parti Prompt. This proves that

		SD XL				SD 1.5			
		CLIP Score ↑	Image Reward ↑	HPS v2 ↑	Unified Metric ↑	CLIP Score ↑	Image Reward ↑	HPS v2 ↑	Unified Metric ↑
DABench	Baseline	83.2	65.1	26.8	49.2	75.4	-73.9	25.6	37.2
	Baseline*	<b>83.4</b>	67.4	26.9	54.2	75.4	-69.7	25.7	39.5
	DiffAgent#	83.0	69.6	26.9	53.0	80.6	-6.7	26.6	67.0
	DiffAgent-SFT	83.2	67.7	27.0	55.8	79.2	-21.0	26.4	59.7
	DiffAgent-RRHF	83.2	<b>71.4</b>	<b>27.0</b>	<b>57.5</b>	<b>80.6</b>	<b>-5.3</b>	<b>26.6</b>	<b>68.2</b>
COCO Caption [2]	Baseline	85.2	71.8	27.4	48.0	78.8	-37.6	26.3	42.3
	Baseline*	<b>85.5</b>	75.3	27.6	53.7	78.8	-33.0	26.4	45.2
	DiffAgent#	85.0	78.9	27.5	52.2	80.3	7.4	26.9	60.8
	DiffAgent-SFT	85.4	79.2	27.6	55.6	79.5	-12.4	26.6	53.5
	DiffAgent-RRHF	85.3	<b>82.7</b>	<b>27.6</b>	<b>57.4</b>	<b>80.3</b>	<b>9.6</b>	<b>26.9</b>	<b>62.2</b>
Parti Prompts [35]	Baseline	86.5	88.7	27.4	49.8	79.9	-30.1	26.1	43.0
	Baseline*	86.6	91.3	27.4	53.6	80.2	-24.6	26.2	46.5
	DiffAgent#	86.4	93.5	27.4	52.5	81.6	12.3	26.7	60.7
	DiffAgent-SFT	86.7	96.4	27.6	56.5	80.3	-0.2	26.5	53.7
	DiffAgent-RRHF	<b>86.8</b>	<b>98.6</b>	<b>27.6</b>	<b>58.8</b>	<b>81.8</b>	<b>13.2</b>	<b>26.7</b>	<b>61.8</b>

Table 2. The evaluation results in SD XL and SD 1.5 model architecture. Baseline\*: Use the baseline model with the parameter information from DiffAgent-RRHF for generation. DiffAgent#: Use the model information from DiffAgent-RRHF with default parameter

the RRHF stage of our framework SFTA can reduce the hallucination error and provide more callable API responses.

#### 4.2.2 Main Experiment

Tab. 2 shows the main results of DiffAgent with different baselines. We report the results of three T2I metrics and the Unified Metric by averaging the scores of three randomly generated images for each T2I API. We can deduce the following conclusions from the results in several aspects.

**Unified Score of DiffAgent** The DiffAgent significantly outperforms the baseline in the Unified Metric in all datasets. Remarkably, DiffAgent-RRHF surpasses the baseline by 8.3 ~ 9.4 in the SD XL architecture and 18.8 ~ 31 in SD 1.5. We find that this improvement is mainly due to the contribution of human preference scores (ImageReward and HPS v2), which indicates that DiffAgent can generate T2I APIs that align well with human preferences. Notably, the models from T2I API are typically specialized for specific styles, resulting in similar outcomes for Clip Score compared with baseline. At the same time, we can observe that under the same architecture, the Unified Scores are very close across different datasets. This proves that this score is a powerful and robust evaluation metric for T2I generation.

**Effectiveness of Different Components** The Baseline\* and DiffAgent# respectively demonstrate the utilization of either parameter or model information from DiffAgent independently, resulting in improved performance compared to the baseline. This shows these two components of T2I API both have a positive effect on original T2I generation.

Furthermore, we observe that the model information from API exhibits a greater improvement for SD 1.5 architecture, whereas, for XL architecture, there is from parameter information. This observation can be attributed to the thriving popularity of models under the SD 1.5 architecture within the community, as depicted in Figure 2.

**Effectiveness of RRHF** We also report the evaluation of DiffAgent-SFT, which represents the results in the first stage of our training framework SFTA, to compare with DiffAgent-RRHF. As shown in Tab. 2, the DiffAgent-RRHF’s performance is higher than the DiffAgent-SFT, especially in human preference scores (*e.g.*, in SD 1.5 architecture, 15.7 improvement in DABench using ImageReward). Notably, DiffAgent-RRHF markedly enhances the unified metric scores by 8.1 ~ 8.7 than DiffAgent-SFT under SD 1.5. This indicates that the RRHF process improves the alignment of responded API with human preferences.

**Inference Time** We evaluate the average time overhead of API generation on DABench, which is 4.81s (single A100 80G), by processing each data independently. This showcases the superior efficient applicability of DiffAgent.

#### 4.3. Human evaluations

We further evaluate our DiffAgent using a user study. Users are provided with two images, one from DiffAgent and one from baseline, along with a corresponding prompt. Users can choose which is better or a tie for the relevance with prompt and image quality (as human preferences). Here, under different model architectures, we sample respective 90 image pairs and prompts evenly across the

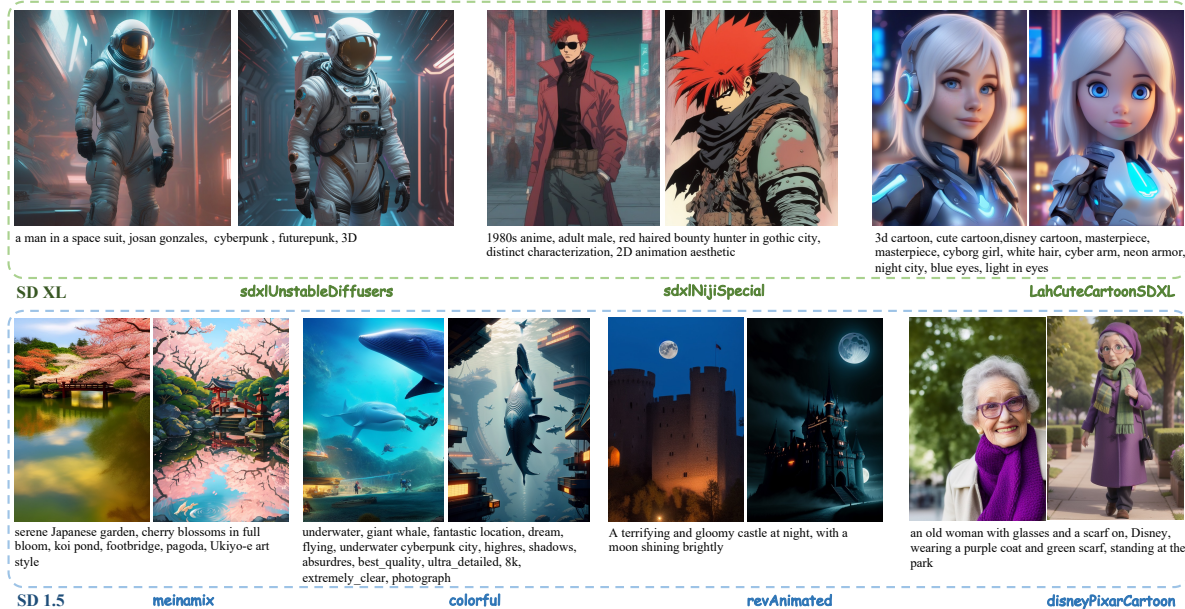


Figure 4. The visualization comparison between the original SD model (left) with DiffAgent’s T2I API (right). The two lines come from the SD XL and SD 1.5 architectures respectively. For each pair, we provide the user prompt and the model name from T2I API.

three evaluation datasets (DABench, COCO Caption, and Parti Prompts) and received evaluations from ten users. As shown in Fig. 5, the user study results align with the unified metric and demonstrate that DiffAgent shows more relevance and human preferences than Baseline. Specifically, DiffAgent achieves a win rate of 78% on relevance and 86% on human preference compared to the baseline in SD 1.5, while for SD XL, it achieves a win rate of 53% and 58%.

#### 4.4. Visualizations

We present a visual comparison between the baseline and T2I API from DiffAgent in Fig. 4. The first row is the results under the SD XL architecture, and the second row is under the SD 1.5. From the comparison, it can be seen that compared to the original one, DiffAgent can find T2I APIs that suit the style of the prompts. For exam-

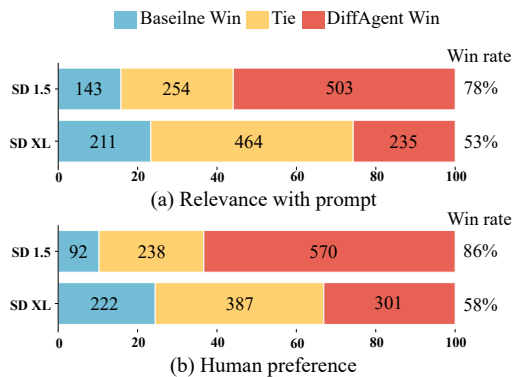


Figure 5. The user study results of DiffAgent. Win rates are calculated without considering tie samples. DiffAgent surpasses the baseline in both relevance with prompt and human preference.

ple, the DiffAgent can capture the Disney style in prompts and responses a T2I API includes the "disneyPixarCartoon" model, a model to make pictures maintain a Disney style.

## 5. Conclusion

Our research is centered on harnessing LLMs as agents to liberate users from constant attempts and adjustments of the generative information in the text-to-image (T2I) domain. This paper introduces DiffAgent, a novel agent model adept at generating T2I APIs (model and parameter information) in response to user specifications. We further present a specialized training framework designed to fine-tune LLMs, enabling them to adeptly select APIs that align well with human preferences. Massive evaluations demonstrate that DiffAgent markedly outperforms existing powerful models in both our collected dataset and two widely used datasets. Notably, DiffAgent exhibits a remarkable proficiency in swiftly identifying the most suitable T2I APIs, expanding the tool usage of LLM in specific domains.

## Acknowledgement

This work was supported by National Science and Technology Major Project (No. 2022ZD0118202), the National Science Fund for Distinguished Young Scholars (No.62025603), the National Natural Science Foundation of China (No. U21B2037, No. U22B2051, No. 62176222, No. 62176223, No. 62176226, No. 62072386, No. 62072387, No. 62072389, No. 62002305 and No. 62272401), and the Natural Science Foundation of Fujian Province of China (No.2021J01002, No.2022J06001).



## References

- [1] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, and Yunxin Jiao. Improving image generation with better captions. 2023. 1, 3
- [2] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 2, 6, 7
- [3] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Neural Information Processing Systems (NeurIPS)*, 2017. 5
- [4] Civitai. Civitai. <https://civitai.com/>, 2022. 1, 2, 3, 4
- [5] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Neural Information Processing Systems (NeurIPS)*, 2021. 3
- [6] Hugging Face. Hugging face. <https://huggingface.co/>, 2016. 1, 2, 3
- [7] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *International Conference on Learning Representations (ICLR)*, 2022. 1
- [8] Shibo Hao, Tianyang Liu, Zhen Wang, and Zhiting Hu. Toolkengpt: Augmenting frozen language models with massive tools via tool embeddings. *arXiv preprint arXiv:2305.11554*, 2023. 3
- [9] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2021. 2, 5, 6
- [10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Neural Information Processing Systems (NeurIPS)*, 2020. 3
- [11] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2021. 1, 2
- [12] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning (ICML)*, 2022. 5
- [13] Alexander Quinn 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. In *International Conference on Machine Learning (ICML)*, 2022. 1, 3
- [14] OpenAI. Openai: Introducing chatgpt. <https://openai.com/blog/chatgpt>, 2022. 4, 6
- [15] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Neural Information Processing Systems (NeurIPS)*, 2022. 3, 5
- [16] Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. Gorilla: Large language model connected with massive apis. *arXiv preprint arXiv:2305.15334*, 2023. 2, 3
- [17] 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. 1
- [18] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023. 3
- [19] 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 (ICML)*, 2021. 3, 5
- [20] 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. 1, 3
- [21] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 3
- [22] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. 1
- [23] 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. In *Neural Information Processing Systems (NeurIPS)*, 2022. 1
- [24] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Neural Information Processing Systems (NeurIPS)*, 2023. 3
- [25] 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. In *Neural Information Processing Systems (NeurIPS)*, 2022. 3
- [26] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 5
- [27] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feed-

- back. In *Neural Information Processing Systems (NeurIPS)*, 2020. [3](#), [5](#)
- [28] Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, and Le Sun. Toolalpaca: Generalized tool learning for language models with 3000 simulated cases. *arXiv preprint arXiv:2306.05301*, 2023. [3](#)
- [29] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. [2](#), [6](#)
- [30] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*, 2016. [6](#)
- [31] 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](#), [5](#), [6](#)
- [32] 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. *arXiv preprint arXiv:2304.05977*, 2023. [2](#), [5](#), [6](#)
- [33] Rui Yang, Lin Song, Yanwei Li, Sijie Zhao, Yixiao Ge, Xiu Li, and Ying Shan. Gpt4tools: Teaching large language model to use tools via self-instruction. *arXiv preprint arXiv:2305.18752*, 2023. [2](#), [3](#)
- [34] Shin-Ying Yeh, Yu-Guan Hsieh, Zhidong Gao, Bernard BW Yang, Giyeong Oh, and Yanmin Gong. Navigating text-to-image customization: From lycoris fine-tuning to model evaluation. *arXiv preprint arXiv:2309.14859*, 2023. [1](#)
- [35] 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](#), [6](#), [7](#)
- [36] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback without tears. In *Neural Information Processing Systems (NeurIPS)*, 2023. [2](#), [3](#), [5](#)
- [37] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019. [3](#)