ChatGen: Automatic Text-to-Image Generation From FreeStyle Chatting

Chengyou Jia[1]*, Changliang Xia[1]*, Zhuohang Dang[1], Weijia Wu[2], Hangwei Qian[3], Minnan Luo[1]†
[1]School of Computer Science and Technology, MOEKLINNS Lab, Xi'an Jiaotong University
[2]National University of Singapore [3]CFAR, A*STAR
cp3jia@stu.xjtu.edu.cn, 202066@stu.xjtu.edu.cn, dangzhuohang@stu.xjtu.edu.cn,
weijiawu96@gmail.com, qian0045@e.ntu.edu.sg, minnluo@xjtu.edu.cn

Abstract

*Despite the significant advancements in text-to-image (T2I) generative models, users often face a trial-and-error challenge in practical scenarios. This challenge arises from the complexity and uncertainty of tedious steps such as crafting suitable prompts, selecting appropriate models, and configuring specific arguments, making users resort to labor-intensive attempts for desired images. This paper proposes **Automatic T2I** generation, which aims to automate these tedious steps, allowing users to simply describe their needs in a freestyle chatting way. To systematically study this problem, we first introduce **ChatGenBench**, a novel benchmark designed for Automatic T2I. It features high-quality paired data with diverse freestyle inputs, enabling comprehensive evaluation of automatic T2I models across all steps. Additionally, recognizing Automatic T2I as a complex multi-step reasoning task, we propose **ChatGen-Evo**, a multi-stage evolution strategy that progressively equips models with essential automation skills. Through extensive evaluation across step-wise accuracy and image quality, ChatGen-Evo significantly enhances performance over various baselines. Our evaluation also uncovers valuable insights for advancing automatic T2I. Our data, code, and models will be publicly available.*

1. Introduction

In recent years, text-to-image (T2I) generative models have attracted considerable attention [1, 23, 31, 32]. Building on advancements in large-scale T2I models such as DALL-E [31] and Stable Diffusion [32], the open-source community has significantly expanded the capabilities of T2I generation. Researchers fine-tune open-source models on specialized datasets, resulting in a diverse selection of task-specific models available on platforms like Civitai [6] and Hugging Face [8]. This variety provides users with a broader range of options to meet customization needs, facilitating the growing adoption of T2I models in real-world applications.

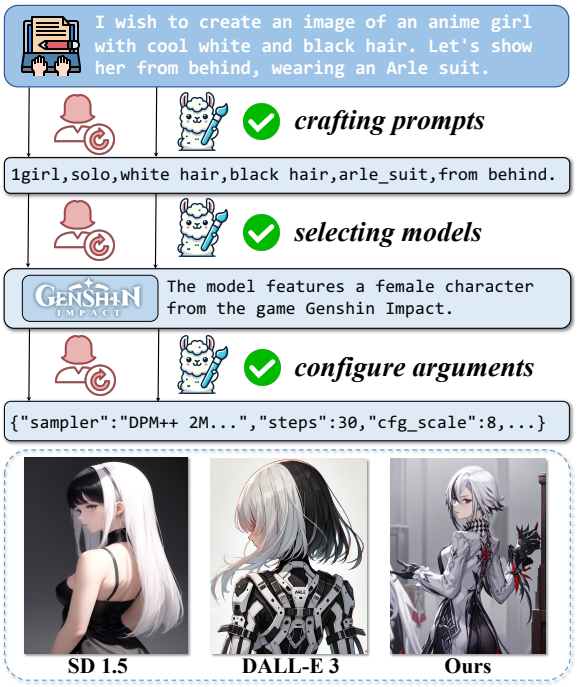

Figure 1. Illustration of tedious steps in T2I. Our method can select an appropriate model with suitable prompts and arguments. *Note: Arle is a character from the game Genshin Impact.*

However, the rapid development of T2I models within the open-source community has also introduced significant challenges for users. When non-experts attempt to create images with specific requirements, they often encounter a trial-and-error process involving several tedious steps. As shown in Figure 1, these steps include **crafting suitable prompts**, **selecting appropriate models**, and **configuring specific model arguments**. The complexity and uncertainty of each step turn the process into an arduous journey, resembling "mice in a maze". In real-world scenarios, this iterative process consumes substantial time and resources

*Equal Contribution.
†Corresponding author.

as users continuously adjust settings to regenerate images. Therefore, we pose the challenging problem: ***can we automate these labor-intensive steps in T2I generation, allowing users to simply describe their needs in a chatting style and receive desired images effortlessly?***

Previous attempts, such as BeautifulPrompt [4, 38] for generating high-quality prompts from low-quality ones and DiffAgent [49] for model selection using large language models, have made some progress in addressing these challenges. However, these methods only seek to automate a specific step in Figure 1, lacking comprehensive research on the automation of entire T2I process. Moreover, they fail to support diverse types of freestyle input, leaving them far from real-world scenarios. To bridge these gaps, we aim to develop the model that can accept arbitrary user input, similar to ChatGPT [24], and automatically generate all necessary components for generation, termed as **Automatic T2I**.

To this end, we first introduce **ChatGenBench**, the benchmark specifically designed for this task. ChatGenBench offers a substantial dataset of high-quality paired data from $6,807$ customized models. Each data pair comprises a user's freestyle chat input, a refined prompt, an appropriate model, and optimal arguments. This comprehensive step-by-step trail enables step-wise evaluation of automatic T2I models, ensuring both quality assessment of the final image output and precise identification of potential automation bottlenecks. Furthermore, ChatGenBench integrates various types of data, enabling testing with multimodal and historical input to simulate practical scenarios.

An intuitive approach to achieve the goal is to build supervised fine-tuning (SFT) data and tune multimodal large language models. However, we argue that automated T2I should be viewed as a complex multi-step reasoning problem. Directly predicting outputs leads the model to focus on simple text mappings, rather than developing the diverse skills needed for robust automation. Inspired by OpenAI's o1 [25], we propose **ChatGen-Evo** with a multi-stage evolution strategy. This approach provides targeted feedback at each stage, progressively equipping models with essential automation skills. First, we construct SFT data to train the model to generate high-quality prompts from freestyle inputs. Next, we augment the model's vocabulary with specialized ModelTokens, enabling effective model selection without affecting other functions. Finally, we guide the model in configuring arguments based on prompts and models selected in previous stages. By decomposing the task into clear stages, the model gains crucial automation skills, delivering outputs aligned with user expectations.

We conduct a comprehensive evaluation of various methods on the novel ChatGenBench to study Automatic T2I. ChatGen-Evo significantly outperforms other methods across all metrics, including both step-wise assessments and image quality evaluations. Extensive quantitative and qualitative results, along with ablation studies, underscore the critical importance of the multi-stage evolution strategy. Additionally, further experiments reveal the impact of different stages on the final T2I output, uncovering valuable insights into the challenges and opportunities in Automatic T2I. We highlight some promising directions, such as exploring the scaling laws of prompt rewriting, improving model selection in few-shot scenarios, and leveraging reasoning approaches to enhance performance. The contributions of this paper are summarized as follows:

- We propose the novel and challenging problem of Automatic T2I generation, aiming to develop the model that can handle user's freestyle chatting and automatically produce all necessary components for image generation.
- We introduce ChatGenBench, a benchmark specifically designed for Automatic T2I. It includes a comprehensive step-by-step trail for step-wise evaluation, supporting multimodal or historical user inputs.
- We present ChatGen-Evo, which adopts a multi-stage evolution strategy. By decomposing the task into distinct stages, ChatGen-Evo enables the model to progressively acquire essential Automatic T2I capabilities.
- Extensive experiments on ChatGenBench not only demonstrate the superiority of ChatGen-Evo but also provide valuable insights for advancing automatic T2I.

2. Related Works

2.1. Text-to-Image Generation

With the advancement of diffusion models [13, 27, 34], text-to-image (T2I) generation [1, 23, 31] has demonstrated exceptional capabilities in high-quality image generation and textual description alignment. Large-scale models such as DALL-E [1] enhance text-image alignment by leveraging the joint feature space of CLIP [30]. Moreover, Stable Diffusion [32], a well-established open-source model, has gained substantial attention. Numerous methods have been developed to fine-tune it or design additional modules for specialized tasks, such as customized image generation [17, 18, 33], layout-to-image generation [14, 19, 20, 47] and image edit [2, 16]. These diverse models have significantly expanded the capabilities of T2I. However, they also present significant learning challenges for non-expert users, underscoring the increasing need for automatic T2I generation.

2.2. LLMs for Text-to-Image Generation

Large language models (LLMs), like ChatGPT [24], Llama [37], have demonstrated impressive capability in language understanding [3] and problem solving [45]. Recently, LLMs have also been applied to image generation. [7, 9, 20, 36, 43] leverage LLMs to generate detailed layout information from complex prompts, enabling the creation of sub-elements and control over their positioning.

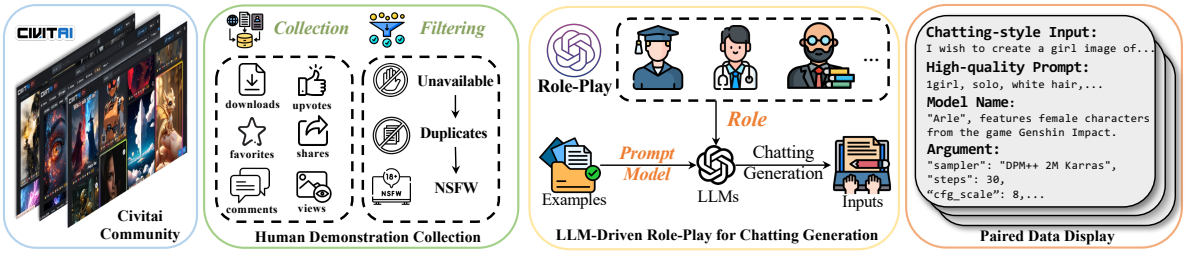

Figure 2. Illustration of the data collection pipeline.

[28, 39, 49] propose employ LLMs for model selection, akin to tool usage [26, 29, 44]. However, the above methods still require specialized prompt inputs and involve complex model usage. Currently, no approach thoroughly considers leveraging LLMs to relieve users from all tedious steps in T2I. Our method aims to address this gap.

3. Methodology

Our goal is to relieve users from tedious steps and automate to produce the desired images from user's freestyle input. To achieve this objective, we first introduce ChatGenBench, a benchmark comprising a large set of user inputs in a chatting style, built upon a foundation of over 6000 personalized models to evaluate automated image generation results. A comprehensive description of data collection, construction, and analysis is provided in Sec. 3.1. Then, we present ChatGen-Evo in Sec. 3.2, which uses a multi-stage evolution strategy to train MLLM for Automatic T2I.

3.1. ChatGenBench: Benchmarking Automatic T2I

For clarity, an example of this data is shown on the right side of Figure 2. The following sections detail the data collection and construction process, which primarily involves **High-Quality Human Demonstration Collection** and using **LLM-driven Role-Play** to simulate user input.

3.1.1 High-Quality Human Demonstration Collection

Civitai [6] is a vibrant community where users share customized AI models for generating high-quality images. Members contribute demonstrations with detailed prompts, model specifications, and arguments, supported by a feedback system that ensures quality validation. These features make Civitai an ideal platform for collecting raw data.

Collection: We start by collecting demonstrations based on established evaluation metrics within Civitai, including download counts, upvotes, and other user feedback. These metrics enable us to collect data that has been validated through community engagement. By focusing on these indicators, we identify a subset of high-quality results.

Filtering: Following the initial collection, we implement a rigorous filtering process to ensure data quality. This involves excluding demonstrations associated with inactive or outdated models, removing duplicates, and filtering out NSFW content. This careful curation refined the dataset to include the most effective demonstrations. Ultimately, this process results in a curated set of 44,881 high-quality human demonstrations across 6,807 unique models.

3.1.2 LLM-Driven Role-Play for Chatting Generation

While the demonstrations collected from communities include the essential procedural information needed for automation, they lack a critical element: freestyle chatting inputs. Such data is not available on open platforms, which has been a key limitation for previous methods [49]. To address this, we propose an LLM-Driven Role-Play strategy for Chatting Generation. As shown in Figure 2, we predefine over 100 roles from everyday life (*e.g.*, students, doctors, professors) and prompt the LLM to simulate these roles, translating each demonstration into freestyle chatting input with tones and habits of the character. This approach significantly enhances data diversity and creates more lifelike inputs. To further enhance reverse synthesis diversity, we incorporate multiple LLMs, adjust diversity arguments, and use BERTScore [48] to filter out overly similar inputs.

Additionally, we define three types of freestyle input formats: single-input, consisting of a single chatting-style sentence; multimodal-input, combining a sentence with an image; and history-input, comprising multiple rounds of dialogue history. These formats effectively simulate how users typically inquire about image generation needs, greatly expanding the practical value of automatic T2I. An example of the single-input prompt is shown in the following. Additional prompts and details are provided in the Appendix.

Prompt: Sing-Input chatting generation

System: You are a professional user experience designer who plays various personas to convert complex and professional content for non-professional users. Please merge the following prompt and model information into a single freestyle query. Remove any obvious details that non-professional users would avoid. Make it similar to what non-professional users may write. The converted single-text query should be colloquial and as brief as possible.
Require: {ROLE}, {PROMPT}, {MODEL}, {Example 1,...,n}
Input: You are playing the {ROLE}. Please generate a single text query based on the following professional {PROMPT} for the {MODEL}. You can refer to following examples: {Example 1,...,n}.

Table 1. Comparison of different methods for benchmarking automatic text-to-image generation.

	TrainSet (# Numbers)	TestSet (# Numbers)	Based Models?	Step-wise Evaluation?	Freestyle User-input?	Multi-modal Support?	History Support?
BeautifulPrompt [4]	143K	2K	Single	✗	✓	✗	✗
DiffChat [38]	234K	5K	Single	✗	✓	✗	✓
DiffusionGPT [28]	-	-	Multi(≈20)	✗	✓	✗	✗
DABench [49]	40K	5K	Multi(5K)	✗	✗	✗	✗
ChatGenBench	256K	14K	Multi(6K)	✓	✓	✓	✓

Table 2. Summary of dataset statistics.

Dataset	Total	Single	M-Modal	History
TrainSet	256,606	147,888	69,548	39,170
TestSet Init	74,364	42,838	20,214	11,312
Benchmark	14,564	11,011	1,668	1,132
Supervised	10,240	8,009	1,099	1,132
Few-Shot	4,324	3,002	569	753

3.1.3 Benchmark Construction

Following the above steps, we generate 330,970 freestyle inputs from 44,881 demonstrations. Considering the large-scale generated data, we don't randomly split a test set for benchmarking. Instead, we carefully select high-quality, non-overlapping samples as the benchmark for evaluation. The selection involves the following steps:

TestSet Split: We split the data based on model origin. For data associated with the same model, we use BERTScore to assess the similarity between samples. From this, we select about 20% most semantically distinct data as initial TestSet in Table 2. This maximizes non-overlap with TrainSet.

TestSet Filtering: Building on the initial TestSet, we perform multiple filtering rounds to ensure the final ChatGenBench's quality. This involves the following steps:

- *Length Filtering*: Remove excessively long entries or those with too many dialogue turns for consistency.
- *Colloquialism Check*: Utilize the Spacy [35] module to filter data for alignment with natural, everyday language.
- *LLM-Based Evaluation*: Employ LLM to assess and select data that best matches the chatting tone.
- *Manual Verification*: Manual verification is conducted to eliminate any inappropriate or irrelevant samples.

Details of each step are provided in the Appendix. Through this rigorous process, we refine the initial 74,364 entries to a final set of 14,564 high-quality, freestyle chatting samples, constituting the novel benchmark.

Setting Division: In practical scenarios, new models often have limited demonstration data, underscoring the importance of evaluations under constrained data. Therefore, we further divide ChatGenBench into two settings—Supervised and Few-shot—based on the availability of samples for each model in the TrainSet. The complete data composition is detailed in Table 2.

3.1.4 Benchmark Analysis

ChatGenBench offers distinct advantages over previous benchmarks, as summarized in Table 1. On the data level, ChatGenBench includes large-scale high-quality data and a broader range of T2I models. Additionally, human demonstrations provide relative ground truth across each step, allowing for step-wise evaluation that pinpoints potential challenges in automation models. ChatGenBench is also the novel benchmark to support multiple input types, making it more aligned with real-world scenarios.

3.2. ChatGen: Achieving Automatic T2I

Our goal is to train the model capable of processing freestyle user inputs and generating the necessary components for image generation (prompt, model, and argument), thereby achieving Automatic T2I. In this section, we first introduce a baseline method, ChatGen-Base. We then analyze the limitations of this approach and subsequently propose ChatGen-Evo, which leverages the multi-stage evolution strategy to enhance performance.

3.2.1 ChatGen-Base with SFT

We first apply supervised fine-tuning (SFT) to develop the ChatGen-Base model as an intuitive baseline. Given a set of freestyle chatting inputs c (which may include text, images, and historical context) and their corresponding outputs comprising prompt p, model m, and argument r, we use the auto-regressive objective to maximize the following loss:

$$L_{sft}^{base} = -\sum_{t} \log P_{\pi}(p, m, r \mid c, *_{<t}). \tag{1}$$

ChatGen-Base satisfies the essential requirements for generating outputs, including handling diverse input modalities and producing corresponding responses. However, even after fine-tuning SOTA open-source MLLMs, the model frequently produces unsatisfactory results.

We attribute this to the complexity of automatic T2I, which requires multi-step reasoning. The quality of generated prompts directly impacts model selection, while model selection serves as a prerequisite for argument configuration. Previous studies [21, 25, 40] have shown that directly

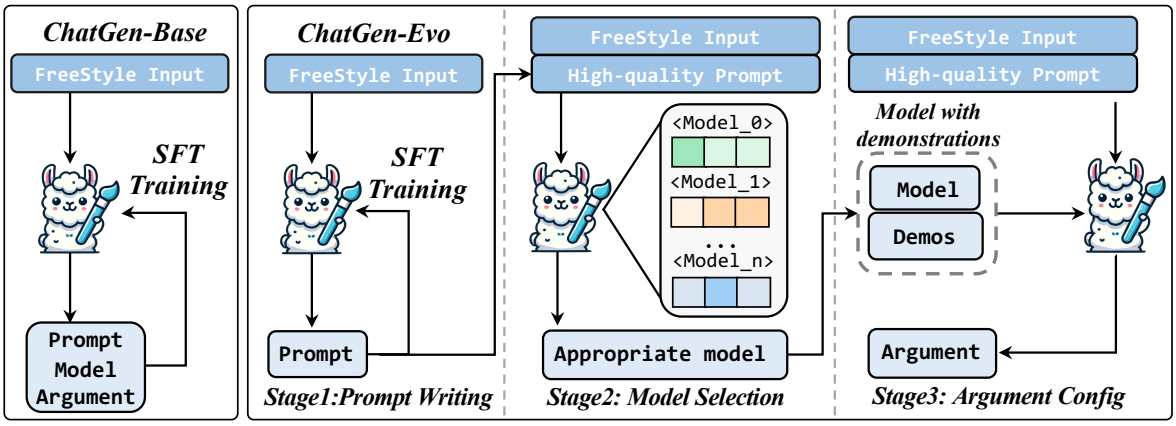

Figure 3. Illustration of the framework for ChatGen-Base and ChatGen-Evo.

predicting answers often fails in multi-step reasoning tasks. Additionally, supervised fine-tuning tends to encourage the model to learn simple text mappings instead of developing skills needed for automation, leading to poor generalization.

3.2.2 ChatGen-Evo

To address the limitations of ChatGen-Base, we propose ChatGen-Evo, which trains MLLM M using a multi-stage evolution strategy. Instead of relying on final outcome supervision as in traditional fine-tuning, the multi-stage evolution strategy in ChatGen-Evo employs stage supervision. By providing more precise feedback at each stage, this approach gradually enables the MLLM to acquire the necessary capabilities for automated T2I. As shown in Figure 3, the training process includes three main stages:

Stage 1: Prompt Writing via SFT. We first train the MLLM using SFT with pairs of freestyle inputs and high-quality prompts. Different from the objective in ChatGen-base, the current stage focuses on a more specific and simplified task instead:

$$L_{sft}^{stage1} = -\sum_{t} \log P_\pi(p \mid c', *_{<t}). \tag{2}$$

Here, c' represents the freestyle input with a prompt prefix that clarifies the task being performed. This prefix helps preserve the MLLM's original capabilities, minimizing catastrophic forgetting [22]. Through this stage, the model learns how to rewrite inputs into effective prompts.

Stage 2: Model Selection via ModelToken. We introduce the ModelToken strategy to equip the model with model selection capabilities without impacting the prompt-writing ability learned in Stage 1. Inspired by token learning [10, 15], ModelToken extends this approach by representing each candidate T2I model as a unique token within the MLLM's vocabulary. Specifically, the model tokens are parameterized as an embedding matrix $W_{\mathcal{M}} \in \mathbb{R}^{|\mathcal{M}| \times d}$ and appended to the original word token matrix $W_\nu \in \mathbb{R}^{|\mathcal{V}| \times d}$.

ModelToken Training: during training, the user input c and prompt p are concatenated as a prefix, with the special model token <Model_i> appended as the ground truth for next-token prediction. The training objective is:

$$L(W_{\mathcal{M}}) = -\log P_\pi(\texttt{<Model\_i>} \mid c, p). \tag{3}$$

Unlike typical next-token prediction training, the embedding matrix $W_{\mathcal{M}}$ represents the only tunable parameters, significantly enhancing training efficiency. With a small parameter size, fewer training samples are required, improving performance in scenarios with limited data.

Inference for Model Selection: Once the embedding matrix is trained, the inference process concatenates the model token and original word token, forming the new language modeling head of the MLLM. In this way, the MLLM predicts the next token with the following probability:

$$P_\pi(m \mid c, p) = \text{softmax}([W_\nu; W_{\mathcal{M}}] \cdot h_{i-1}), \tag{4}$$

where the operation $[;]$ denotes concatenation, and $h_{i-1} \in \mathbb{R}^d$ represents the last hidden state. Once a model token is predicted, the MLLM stops decoding, and the corresponding model m is selected. Additionally, information such as the model's description and demonstrations is loaded for subsequent use. After Stage 2, the model not only retains its prompt-writing skills but also learns to select models.

Stage 3: Argument Configuration via In-Context Learning. After the above two stages, we have obtained the prompt p and model m from the original user input c. The model now needs to generate the appropriate argument configuration to complete the final generation. Due to the careful design of the previous stages, the model's in-context learning ability is maximally preserved. Therefore, we adopt a training-free approach: we guide the MLLM using in-context demonstrations of m, the user's original input c, and the rewritten prompt p. The MLLM can follow the demonstration pattern to complete the parameter configuration for the current user request:

$$a = M(c, p, D(m)), \tag{5}$$

where $D(m)$ represents the set of demonstrations for model m in the training dataset. Thanks to the prior acquisition of the relevant model and prompt in earlier stages, this approach frees up context space, enabling extensive demonstrations. Additionally, the train-free approach avoids interfering with the trained model from the previous two stages.

4. Experiment

We conduct a comprehensive evaluation of various methods on the novel ChatGenBench. First, in section 4.2, we compare ChatGen-Evo with other baseline models, highlighting the efficacy and efficiency of our multi-stage evolution strategy. Next, in section 4.3, we perform extensive ablation studies, uncovering the impact of each step on the final results and providing valuable insights. Finally, we provide visualizations of the generated images in section 4.4.

4.1. Experimental Settings

Training Setups. We adopt InternVL2 [5] as the base MLLM, fully fine-tuning it for both ChatGen-Base and the first stage of ChatGen-Evo. In the second stage of ChatGen-Evo, all model parameters are frozen except for the Model-Token embeddings. We employ the AdamW optimizer with a learning rate of 4e-5 and a weight decay of 1.0 over 5 epochs, maintaining these settings consistently across training stages. All experiments are conducted on 8 A100 GPUs.

Metrics for Step-wise Evaluation: Leveraging the comprehensive process data in ChatGenBench, we introduce the step-wise evaluation metrics to assess the distinct abilities of automatic T2I models in key stages:

- ***Prompt BERTScore:*** To assess prompt rewriting ability, we use BERTScore [48] to compare predicted prompts with high-quality, human-validated prompts. BERTScore leverages pre-trained contextual embeddings to match words in candidate and reference sentences. The metric ranges from 0 to 1, with 1 indicating highly similar meanings and 0 signifying complete dissimilarity.
- ***Selection Accuracy:*** We calculate the accuracy of model selection by comparing the predicted T2I model with the human-validated model.
- ***Argument Accuracy:*** We evaluate argument configuration by calculating the exact match accuracy between the predicted and human-validated arguments. The overall accuracy is obtained by averaging individual arguments.

It is important to note that obtaining the absolute ground truth across all stages is nearly impossible due to the infinite search space. Therefore, we use human-validated high-quality records as **relative ground truth**. While these are not absolute, the comprehensive evaluation of the large-scale benchmark is able to reflect the capabilities of automation models, as confirmed by experimental results.

Image Quality Evaluation: We use HPS v2 [41] and ImageReward [42] metrics to assess the quality of generated images, reflecting alignment with human preferences. Additionally, we employ FID and CLIP Score to evaluate how well the generated images meet user requirements. FID [12] measures the distance between automatically generated images and human-validated high-quality images, while CLIP Score [11] calculates the similarity between the generated images and human-validated prompts. To provide an intuitive and comprehensive measure of image quality, we normalize and combine these four metrics into an aggregated score, **Unified Metric** [49]. Details of each metric and the calculation of Unified Metric are presented in the Appendix.

Baseline Without Fine-Tuning. We further establish a baseline that uses the default in-context learning capabilities of the MLLM for prompt rewriting, along with a single Stable Diffusion model and a fixed set of default parameters. This baseline helps emphasize the significance of prompt rewriting and multi-model utilization.

4.2. Main Experiment

4.2.1 Quantitative Results

Table 3 presents the main quantitative results of ChatGen-Evo compared to different baselines. Overall, ChatGen-Evo significantly outperforms other methods across all metrics, including both step-wise and final image quality evaluations. Specifically, the low performance of the baseline highlights the importance of effective prompt rewriting and multi-model selection, underscoring the necessity of dedicated Automatic T2I methods. Additionally, fine-tuning MLLMs with progressively larger parameter scales yields steady performance improvements. Remarkably, ChatGen-Evo achieves performance comparable to ChatGen-Base at 8B, despite utilizing a significantly smaller parameter scale.

The comparisons across different settings provide further insights. Prompt rewriting ability demonstrates strong generalizability, effectively transferring to rarely seen models. The higher few-shot performance can be attributed to the bias in data distribution between the two settings, as reflected in the "Baseline" results, which reveal a difference of approximately 0.3. In contrast, model selection and parameter configuration show a noticeable decline in performance under few-shot conditions, highlighting their reliance on more training samples. Therefore, exploring the scaling laws of prompt rewriting and enhancing model selection in few-shot scenarios are promising directions.

4.2.2 Human Evaluation

We conduct a user study using pairwise comparisons to further evaluate ChatGen-Base(8B) and ChatGen-Evo(2B). Users are presented with two images generated from the same input: one by ChatGen-Base and the other by

Table 3. The Step-wise and Final evaluation results of different methods on ChatGenBench.

		Step-wise Evaluation			Final Evaluation				
		Prompt BertScore ↑	Selection Acc ↑	Config Acc ↑	FID Score ↓	CLIP Score ↑	HPS v2 ↑	Image Reward ↑	Unified Metric ↑
Supervised	Baseline	0.026	-	-	32.7	64.6	20.2	-34.6	37.3
	ChatGen-Base(2B)	0.184	0.206	0.384	21.3	69.9	23.5	2.4	59.0
	ChatGen-Base(4B)	0.197	0.230	0.490	20.7	70.0	23.4	1.5	58.7
	ChatGen-Base(8B)	0.208	0.264	0.509	20.8	70.7	23.9	4.0	60.7
	ChatGen-Evo (2B)	0.247	0.328	0.537	19.1	72.9	25.1	8.9	65.9
Few-shot	Baseline	0.055	-	-	54.4	63.4	20.0	-40.2	29.7
	ChatGen-Base(2B)	0.221	0.153	0.349	42.8	69.1	23.3	-4.8	51.1
	ChatGen-Base(4B)	0.236	0.171	0.448	41.2	69.4	23.4	-4.3	51.9
	ChatGen-Base(8B)	0.252	0.201	0.462	41.4	70.6	23.4	-3.1	52.5
	ChatGen-Evo (2B)	0.283	0.231	0.493	40.7	72.5	25.0	5.1	59.2

Table 4. Efficiency comparison. Training efficiency is measured in GPU hours, while inference is expressed as the seconds required to process each data. The training was conducted on 8 A100 GPUs, and inference was performed on a single A100 80GB GPU.

Method	Params	Training	Inference
ChatGen-Base(2B)	2.21B	76h	1.1s
ChatGen-Base(8B)	8.08B	240h	2.3s
ChatGen-Evo(2B)	2.24B	100h	1.9s

ChatGen-Evo. It is tasked with selecting the image that better matches the image quality and relevance to the given input. We sample 2,000 image pairs for supervised and 1,000 for few-shot. More details are provided in the Appendix.

As shown in Figure 4, the human evaluation results are consistent with the quantitative metrics, highlighting that ChatGen-Evo outperforms ChatGen-Base in both image quality and relevance. Additionally, in few-shot settings, ChatGen-Evo demonstrates a higher win rate, showcasing the effectiveness of our approach in data-scarce scenarios.

4.2.3 Efficiency Comparison

Similar to existing multi-stage reasoning methods [25, 40, 46], ChatGen-Evo does not offer an efficiency advantage over direct prediction approaches like ChatGen-Base. However, the significant performance gains more than compensate for this drawback. As shown in Table 4, ChatGen-Evo achieves the performance level of ChatGen-Base at 8B parameters using only 2B parameters. Therefore, when comparing efficiency under equal performance, ChatGen-Evo maintains a relative advantage over ChatGen-Base.

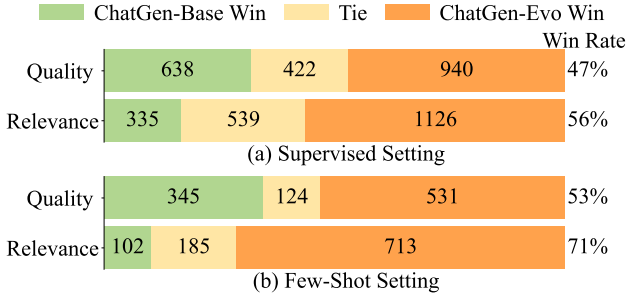

Figure 4. User study results of ChatGen-Base and ChatGen-Evo.

Table 5. Ablation experiments on the supervised setting.

Stage	Methods	Step Score ↑	FID Score ↓	Image Reward ↑	Unified Metric ↑
Prompt Writing	Baseline	0.026	19.3	-13.1	50.8
	Chat-Base	0.184	21.3	3.1	59.7
	Chat-Evo	0.247	17.6	10.5	66.8
Model Selection	Chat-Base	0.206	20.5	9.1	66.0
	Chat-Evo	0.553	18.2	16.8	69.7
Argument Setting	Chat-Base	0.384	17.9	10.4	69.1
	Chat-Evo	0.871	17.1	17.9	70.3

4.3. Analysis

4.3.1 Capability Analysis.

We conduct ablation experiments on ChatGenBench to evaluate the contribution of individual steps to final performance by providing ground truth for the other steps.

Prompt Writing: We first compare the prompt writing capabilities of different methods, along with their final performance. In these experiments, each method's predicted prompt was passed along with the correct model and argument. As shown in Table 5, all methods exhibit noticeable gaps compared to human-validated prompts, emphasizing the complexity of prompt rewriting. Comparisons with the results in Table 3 also highlight the further improvements achieved by selecting the correct model and arguments, particularly for the "Baseline" methods. Moreover, variations in prompts significantly influence the final results, highlighting its critical role in Automatic T2I.

Model Selection: Table 5 also shows the impact of model selection where human-validated prompt and argument are provided. With well-crafted prompts, ChatGen-Evo demonstrates a substantial performance boost from 32.8% to 55.3%, indicating the strong influence of prompt quality on model selection accuracy. This supports our perspec-

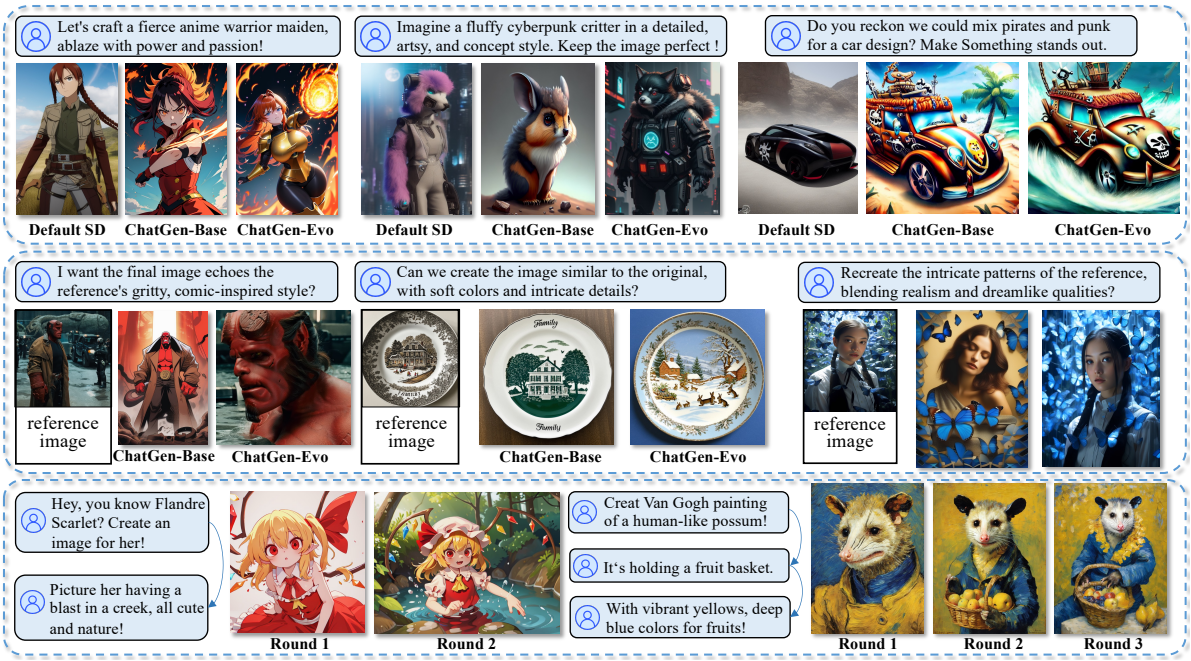

Figure 5. Examples of images generated by different methods. Three rows represent single, multi-modal and historical inputs, respectively.

tive that Automatic T2I fundamentally involves multi-step reasoning. In contrast, ChatGen-Base, which directly predicts all results, fails to adapt and thus produces unchanged outputs. Furthermore, these results also emphasize the substantial impact of model selection on overall performance.

Argument Configuration: When provided with high-quality prompts and appropriate model selection, ChatGen-Evo exhibits notable performance improvements. It improves configuration accuracy from 53.7% to 87.1% and Unified Score from 65.9 to 70.3. Overall, while argument configuration has a relatively smaller impact compared to previous stages, it remains a crucial component.

The above findings suggest that outcomes in earlier steps significantly influence predictions in subsequent ones. Therefore, exploring more reasoning methods for advancing automatic T2I represents a promising research direction.

4.3.2 Input Type Analysis.

Table 6 presents ChatGen-Evo performance across different input types. Multimodal inputs lead to better performance, as images may offer clearer prompt and model identification cues compared to text alone. Additionally, handling historical data remains a significant challenge, with the lowest performance across all metrics. These results point to valuable directions in enhancing history-based prompt generation.

4.4. Visualizations

Figure 5 presents examples of images generated by different methods. In the first row, ChatGen-Evo understands user requirements and identifies suitable models to generate style-matching images. The second row demonstrates results based on multi-modal user inputs, where ChatGen-Evo shows a superior understanding of the reference image and preserves more details to generate refined outputs. The third row illustrates ChatGen-Evo's capability in handling historical data, ensuring that each step inherits the previous style while making appropriate modifications.

Table 6. The evaluation results of different input types.

Input Type	Step-wise Evaluation			Final
	Prompt BertScore ↑	Selection Acc ↑	Config Acc ↑	Unified Score ↑
Single	0.252	0.331	0.539	68.1
M-Modal	0.277	0.381	0.554	69.3
History	0.165	0.259	0.505	60.1

5. Conclusion

Our research aims to automate tedious steps in T2I generation, allowing users to simply describe their needs in a freestyle chatting way. We introduce **ChatGenBench** for benchmarking Automatic T2I task. It includes high-quality paired data with diverse freestyle user inputs, enabling comprehensive evaluation across all steps. Furthermore, we argue that Automatic T2I should be regarded as a multi-step reasoning task. Consequently, we propose **ChatGen-Evo**, a multi-stage evolution strategy that progressively equips models with essential automation skills. Extensive evaluations not only demonstrate the superiority of ChatGen-Evo but also provide valuable insights for advancing Automatic T2I. We believe this represents a significant step toward the future of automated generative models.

Acknowledgments

This work was supported by the National Nature Science Foundation of China (No. 62192781, No. 62272374), the Natural Science Foundation of Shaanxi Province (2024JC-JCQN-62), the National Nature Science Foundation of China (No. 62202367, No. 62250009, No. 62137002), the Key Research and Development Project in Shaanxi Province No. 2022GXLH-01-03, Project of China Knowledge Center for Engineering Science and Technology, and Project of Chinese academy of engineering "The Online and Offline Mixed Educational Service System for 'The Belt and Road' Training in MOOC China", and the K. C. Wong Education Foundation.

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, 2

[2] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 2

[3] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 2

[4] Tingfeng Cao, Chengyu Wang, Bingyan Liu, Ziheng Wu, Jinhui Zhu, and Jun Huang. Beautifulprompt: Towards automatic prompt engineering for text-to-image synthesis. *arXiv preprint arXiv:2311.06752*, 2023. 2, 4

[5] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024. 6

[6] Civitai. Civitai. https://civitai.com/, 2022. 1, 3

[7] Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, et al. Dreamllm: Synergistic multimodal comprehension and creation. *arXiv preprint arXiv:2309.11499*, 2023. 2

[8] Hugging Face. Hugging face. https://huggingface.co/, 2016. 1

[9] Yutong Feng, Biao Gong, Di Chen, Yujun Shen, Yu Liu, and Jingren Zhou. Ranni: Taming text-to-image diffusion for accurate instruction following. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4744–4753, 2024. 2

[10] Shibo Hao, Tianyang Liu, Zhen Wang, and Zhiting Hu. Toolkengpt: Augmenting frozen language models with massive tools via tool embeddings. *Advances in neural information processing systems*, 36, 2024. 5

[11] 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. 6

[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[14] Chengyou Jia, Minnan Luo, Zhuohang Dang, Guang Dai, Xiaojun Chang, Mengmeng Wang, and Jingdong Wang. Ssmg: Spatial-semantic map guided diffusion model for free-form layout-to-image generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2480–2488, 2024. 2

[15] Chengyou Jia, Minnan Luo, Zhuohang Dang, Qiushi Sun, Fangzhi Xu, Junlin Hu, Tianbao Xie, and Zhiyong Wu. Agentstore: Scalable integration of heterogeneous agents as specialized generalist computer assistant. *arXiv preprint arXiv:2410.18603*, 2024. 5

[16] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023. 2

[17] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941, 2023. 2

[18] Dongxu Li, Junnan Li, and Steven Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[19] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22511–22521, 2023. 2

[20] Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. Llm-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *arXiv preprint arXiv:2305.13655*, 2023. 2

[21] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023. 4

[22] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*, 2023. 5

[23] 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, 2

[24] OpenAI. Chatgpt, 2022. Large language model. 2

[25] OpenAI. Learning to reason with llms. https://openai.com/index/learning-to-reason-with-llms/, 2024. Accessed: 2024-11-06. 2, 4, 7

[26] 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. 3

[27] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 2

[28] Jie Qin, Jie Wu, Weifeng Chen, Yuxi Ren, Huixia Li, Hefeng Wu, Xuefeng Xiao, Rui Wang, and Shilei Wen. Diffusiongpt: Llm-driven text-to-image generation system. *arXiv preprint arXiv:2401.10061*, 2024. 3, 4

[29] 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

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

[31] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 1, 2

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2

[33] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023. 2

[34] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[35] SpaCy. spacy: Industrial-strength natural language processing in python. https://spacy.io/, 2015. 4

[36] Omost Team. Omost github page, 2024. 2

[37] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2

[38] Jiapeng Wang, Chengyu Wang, Tingfeng Cao, Jun Huang, and Lianwen Jin. Diffchat: Learning to chat with text-to-image synthesis models for interactive image creation. *arXiv preprint arXiv:2403.04997*, 2024. 2, 4

[39] Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. Genartist: Multimodal llm as an agent for unified image generation and editing. *arXiv preprint arXiv:2407.05600*, 2024. 3

[40] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 4, 7

[41] 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. 6

[42] 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. 6

[43] Ling Yang, Zhaochen Yu, Chenlin Meng, Minkai Xu, Stefano Ermon, and CUI Bin. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal llms. In *Forty-first International Conference on Machine Learning*, 2024. 2

[44] 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. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[45] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022. 2

[46] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024. 7

[47] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 2

[48] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*. 3, 6

[49] Lirui Zhao, Yue Yang, Kaipeng Zhang, Wenqi Shao, Yuxin Zhang, Yu Qiao, Ping Luo, and Rongrong Ji. Diffagent: Fast and accurate text-to-image api selection with large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6390–6399, 2024. 2, 3, 4, 6