PromptCoT: Align Prompt Distribution via Adapted Chain of Thought (Supplementary material)

Anonymous Author(s)
Affiliation
Address
email

A Overview

This document serves as supplementary material to the main paper. We present additional implementation details in Section B, including the construction of datasets, fine-tuning settings, and an introduction to evaluation metrics. Section C contains additional experimental results, while Section D discusses the ablation study on CoT dataset and adapters. Furthermore, we include extra visualization examples in Section E. We also address the limitations and societal impact of our work in Section F, and provide a checklist in Section G.

B Additional Implementation Details

Data collection. We first explore how the length of the text descriptions impacts the generation performance of the model. Figure 1 displays the distribution of text length in the LAION dataset [9], revealing that the majority of text descriptions fall within the range of 10 to 150 characters. To facilitate distinct analysis, the dataset is divided into three separate groups, each consisting of 20,000 data samples. The first group, named *short-cap*, encompasses captions with a length of less than 40 characters. The second group, referred to as *mid-cap*, comprises captions exceeding 90 characters but falling short of 110 characters. Finally, the third group, denoted as *long-cap*, includes captions surpassing 150 characters. The intentional avoidance of consecutive length ranges ensures clear differentiation between the groups, allowing for ease of distinction. Utilizing a pre-trained latent diffusion model, three sets of images are generated based on the text descriptions from the respective groups. The calculated mean aesthetic scores [7] for each group are as follows: 6.01 for *short-cap*, 6.03 for *mid-cap*, and 5.99 for *long-cap*. Furthermore, the Fréchet Inception Distance (FID) [2] is computed, resulting in values of 13.1 for *short-cap*, 9.4 for *mid-cap*, and 10.8 for *long-cap*. Notably, no significant impact of text length on the quality of the generated images is observed. Consequently, a uniform sampling strategy is employed for all sub-datasets utilized throughout the paper.

Training settings. All experiments are based on pre-trained LLaMA-7B [11], an open-sourced Large Language Model with seven billion parameters. The fine-tuning process of each aligner follows [10, 12] using 8$\times$A100-80GB GPUs, which takes three hours until converge. More specifically, we set 2e-5 for the learning rate, 0.0 for weight_decay, 0.03 for warmup_ratio, and cosine decay for the learning rate schedule. For all one-step aligners, including text continuation, text imitation, and direct aligner with training dataset from CoT, the max sequence length is set to 512 while the batch size is 2 and gradient accumulation steps are 8. For CoT aligners, the max sequence length is set to 1500 while the batch size is 1 and the gradient accumulation steps are 2.

Adapter setting. In PromptCoT, we add adapter layers following [1]. For all aligners, we set the number of adapter layers to 30 with each length of 10, initial learning rate to 9e-3, weight_decay to 0.02 and 5 epochs within 2 warming up epochs. For all one-step aligners, including text continuation,

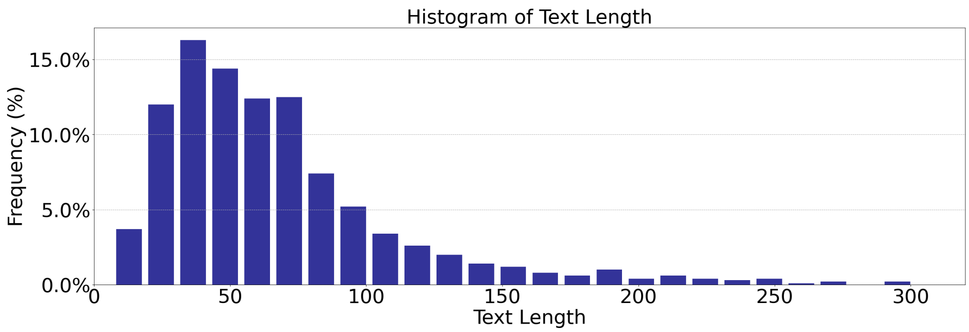

Figure 1: The distribution of text lengths in the LAION dataset.

text imitation, and direct aligner with the training dataset from CoT, the max sequence length is set to 512 while batch size is 8. For PromptCoT aligners, the max sequence length is set to 1500 while batch size is 1. The use of adapter significantly reduces memory cost since it takes $n \times 26GB$ for n finetuned aligners but only $26GB + n \times 4.8MB$ for n aligners with adapters.

Evaluation Metrics. We evaluate the generation performance with Fréchet Inception Distance (FID) [2], Inception Score (IS) [8], CLIP score [6], Aesthetic Score [7] and PickScore [3]. The definitions of FID, IS, and CLIP score are strictly following previous works[2, 8, 6, 7, 3]. We here give more detailed explanations of Aesthetic Score and PickScore in this paragraph.
Aesthetic Score is calculated with a pre-trained aesthetics predictor provided by LAION [9]. It also has been used for data filtering of recent popular latent diffusion models [7]. It is designed based on CLIP ViT/14 with an extra linear layer at the top of the model. The model is optimized to predict the ratings collected from people's answers to questions such as "How much do you like this image on a scale from 1 to 10?". In this paper, we use the aesthetic score to show that after being refined by our prompt aligner, generative models can create images that human regards as amusing.
PickScore [3] is a scoring function trained over Pick-a-Pic by combining a CLIP-style model with a variant of InstructGPT's [5] reward model objective whose goal is to predict human preferences. We use PickScore to construct two kinds of evaluation metrics to represent how humans like the generated image. Each time we input a group of generated images led by prompts refined from our different aligners and the prompt refined from the aligner being evaluated. The average PickScore is the probability that a human is predicted to prefer the image generated by the input prompt among this group of images, while the recall PickScore is the rate that predicted human reaction is preferring the corresponding image.

C Additional Experiments

C.1 PickScore for Adapter

We provide additional PickScore results of aligners with adaptation in Table C. Experiments indicate that all aligners consistently improve this metric compared to the baseline.

Table 1: **Text-to-image generation performance of aligners with adaptation.**

Base Model	**Aligner**	**PickScore(%) (Average/Recall)**
Adapter	baseline	27.3/37.3
	t-continue	41.1/67.9
	t2t-blip	42.5/66.7
	t2t-inter	33.8/48.7
	cot_d	41.9/66.2

C.2 PickScore for Stable Diffusion v2

Table 2: **Text-to-image generation performance of fully fine-tuned aligners.**

Generator	Aligner	PickScore(%) (Average/Recall)
SD v2.1 ddim step=250 scale=12.0	baseline	29.3/41.9
	t-continue	44.7/70.7
	t2t-blip	56.4/83.2
	t2t-inter	37.3/56.4
	cot_d	41.0/64.4

Table 2 presents additional PickScore [3] results for the generation performance of various aligners on the COCO [4] validation set. The experiments are conducted using Stable Diffusion v2.1. Our results show that all aligners significantly outperform the baseline on this metric.

D Ablation Study

D.1 Training PromptCoT Exclusively with CoT Dataset

We conducted the ablation study to compare the performance of the full-pipeline PromptCoT aligner, *cot*, with several variants on a subset of the COCO [4] validation dataset consisting of 1,000 images. The variants included *cot_d*, which is an aligner trained exclusively on the results of the final step (step 5) to accelerate inference. The variants also include *cot_only*, which is trained without datasets of Alpaca [10], text continuation, and text imitation, solely on the CoT dataset to accelerate training. Our experiments (Table 3) indicate that although these more efficient variants have a subtle impact on marginal aspects, they still deliver impressive final performance.

Table 3: **Text-to-image generation performance on different CoT aligners.** All metrics are evaluated on a subset of the COCO [4] validation dataset consisting of 1,000 images. Images are generated by Stable Diffusion with corresponding prompts under the same conditions.

Aligner	Aesthetic Score	CLIP Score	PickScore (%) (Average/Recall)
baseline	5.62	0.231	28.4/40.7
cot_d	5.79	0.291	47.0/**65.1**
cot_only	**5.80**	**0.293**	43.2/59.5
cot	**5.80**	**0.293**	**47.2**/64.4

D.2 PromptCoT with Adapter

Table 4: **Text-to-image generation performance with adaptation.** PromptCoT with adaptation achieves comparable results compared to the fully fine-tuned counterpart.

Base Model	Aligner	Aesthetic Score	FID	CLIP Score
Adapter	baseline	5.60	58.02	0.266
	cot_d	**5.85**	51.06	0.251
	PromptCoT	5.80	**46.54**	**0.291**

We further conduct a complementary evaluation of full-pipeline PromptCoT with the adaption approach on COCO validation dataset with 25,000 images in Table 4. Experiments indicate that adaptation achieves comparable performance on Aesthetic Score and improvement on FID and CLIP Score, compared to the fully fine-tuned counterpart.

D.3 Comparison between PromptCoT and Human-refined Prompts

To compare the capability of refining prompts between PromptCoT and human beings, we first collect a set of text prompts from the captions of COCO dataset. We then invited a group of 30 research volunteers to refine the collected prompts to improve the image generation quality. The volunteers are all specialized in deep learning algorithms and are thus expected to perform well on this task. The findings are succinctly presented in Table 5. Upon careful examination, it is evident that humans possess the ability to modify prompts to achieve better content alignment between the text descriptions and the generated images, resulting in an improved CLIP score. However, it should be noted that there is a slight decrease in aesthetic scores when employing this approach. Conversely, PromptCoT demonstrates its capability to generate prompts that enhance not only the aesthetic score but also the CLIP score and PickScore, surpassing human performance by a significantly larger margin.

Table 5: **Comparison to human-refined promtps. We evaluate the generation quality on Aesthetic Score [7], CLIP Score [6] and PickScore [3].**

Aligner	Aesthetic Score	CLIP Score	PickScore(%) (Average/Recall)
Baseline	5.68	0.23	33.2/39.1
Human	5.62	0.27	48.1/58.2
PromptCoT	**5.77**	**0.30**	**57.5/73.6**

E Additional Visualization

E.1 Impacts of Prompts in Training Data on Generation Performance

Our empirical findings indicate a positive correlation between the quality of prompts associated with high-quality images in the training dataset and the generation of superior images when applied to pre-trained latent diffusion models. This relationship is visually represented in Figure 2. Figure 2 portrays an instance of a text-image pair characterized by low visual quality, prominently displayed in the top-left corner and highlighted in orange. Consequently, the resulting generated images derived from such prompts exhibit a corresponding decline in visual quality. Conversely, the last two rows of Figure 2 present a contrasting scenario where text prompts sourced from high-visual-quality training samples yield images of commendable visual quality.

E.2 Impacts of PromptCoT Compared to Online Users

In this section, we utilize prompts collected from an online database [13], where users share their self-generated prompt-image pairs. We also verify the effectiveness of PromptCoT on those real-world prompts. The results are shown in Figure 4. The left column shows the images generated with the original prompt used by the public and the right column shows the images generated with the refined prompt by PromptCoT. The original prompt and the refined prompt are also listed under the corresponding image pairs. It is essential to highlight that the quality of the generated images cannot be attributed solely to the prompt's length. Even when users provide detailed descriptions, the generated images may still fall short of expectations. For example, in the first row in Figure 4, the online user attempts to depict a construction worker in a construction field by providing unorganized key concepts. However, the resulting generation exhibits flaws in the worker's clothing, eyes, and background, indicating a lack of coherence and quality. In the second-row pairs, the user-generated image lacks the “full body” concept, leading to a partial representation of the prompt. In the bottom-row pairs, the user's prompt for generating the well-known character "Rocket Raccoon" exhibits unrealistic body proportions. In each of these instances, the utilization of PromptCoT yields a noteworthy enhancement in the quality of generated outputs. This improvement is achieved through the process of prompt re-writing, which ensures a more effective alignment with the training text data. As a result, the generated images exhibit a heightened level of fidelity and aesthetics, thereby attaining a closer resemblance to the intended expectations.

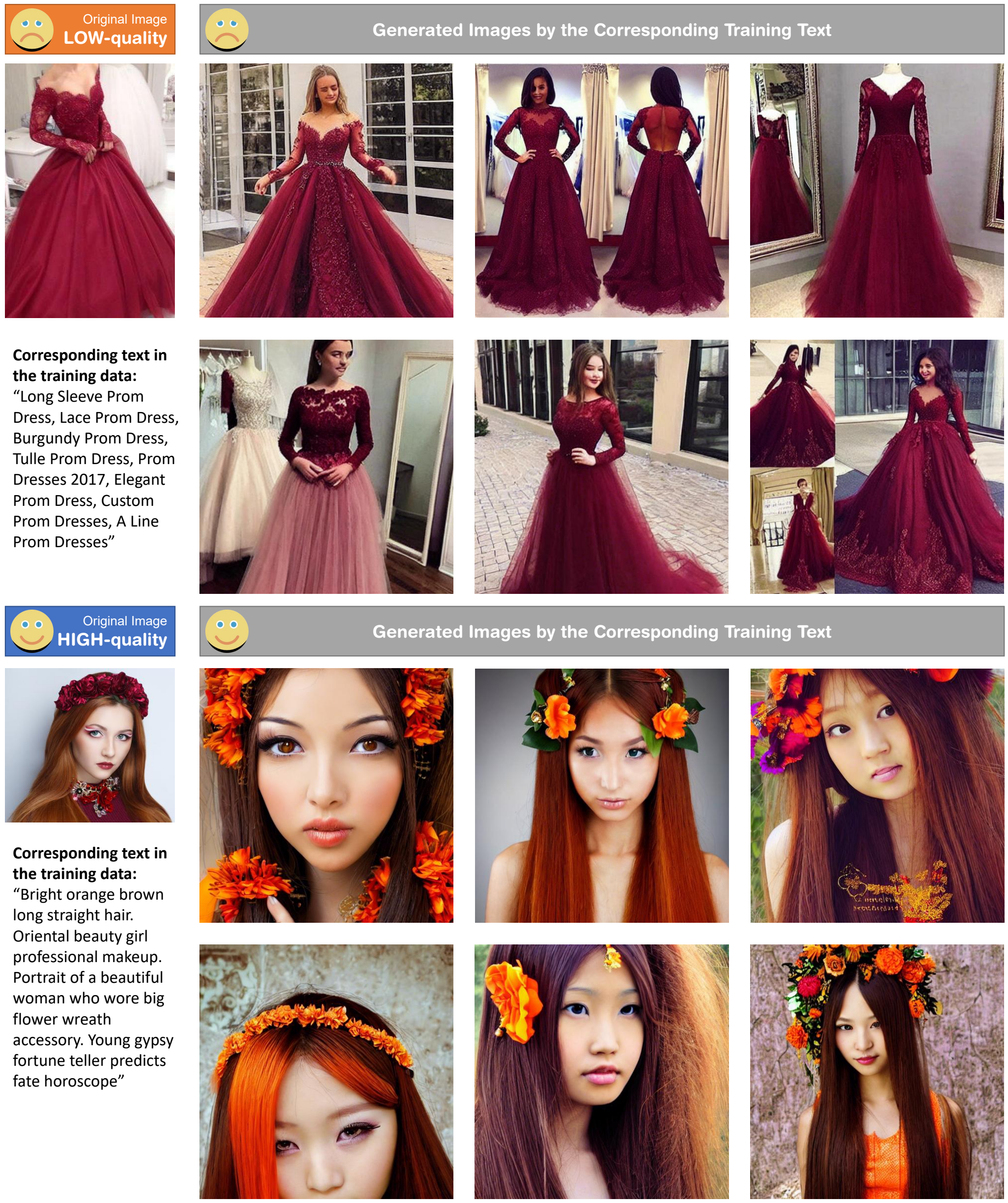

Figure 2: “Low-quality prompt” refers to the text in the training set whose corresponding image **(left)** has low quality. **(Up)** Images generated by a low-quality prompt. “High-quality prompt” refers to the text in the training set, and whose corresponding image has high quality. **(Bottom)** Images generated by a high-quality prompt.

E.3 Visualization of Different Aligners

In Figure 5, we provide a detailed visual comparison of images generated using the original prompt and those refined with different aligners (tcontinue, t2t_blip, t2t_inter, cot_davinci, cot_d, and PromptCoT). We have highlighted inconsistencies between the prompt and the images within the figures, accompanied by annotations below each image. It is noteworthy that not only do the images generated using PromptCoT exhibit superior quality, but they also display a better alignment with the textual contents. For instance, in the top-row images generated from the prompt "A surfer on a whiteboard riding a small wave," PromptCoT stands out by effectively capturing all the desired elements, while others may struggle to interpret the prompt accurately with all key concepts.

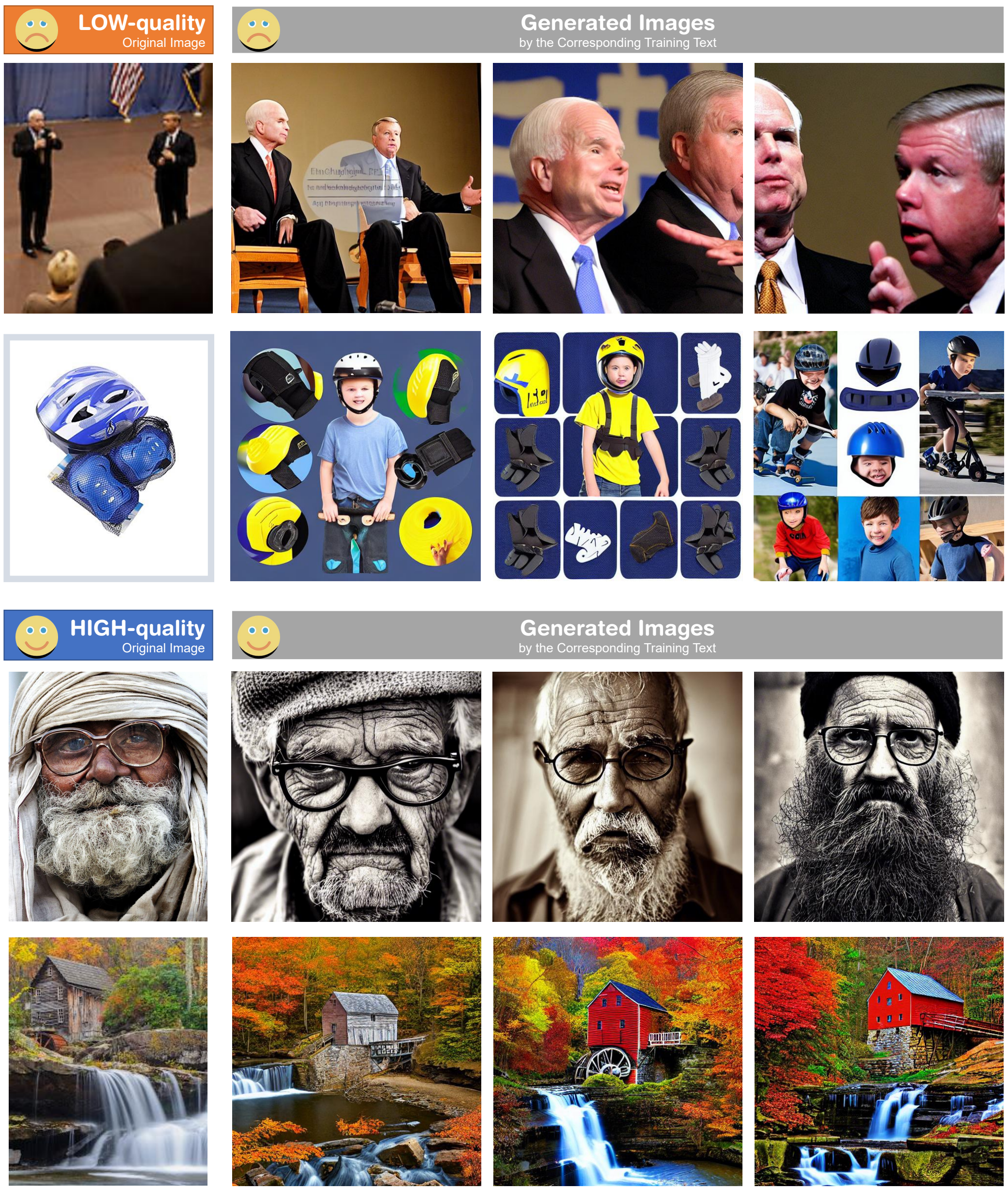

Figure 3: More examples of images generated by low/high-quality prompts.

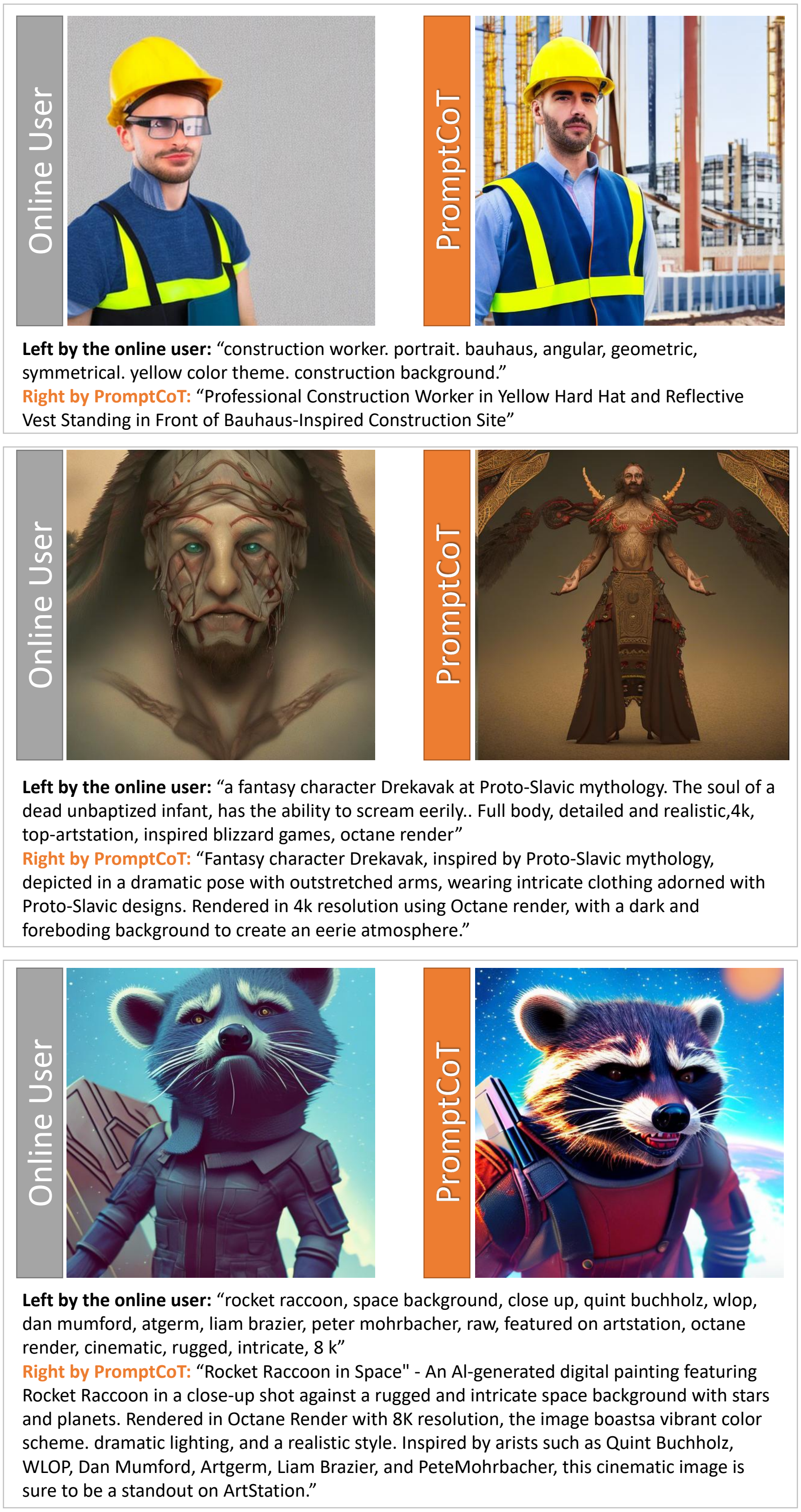

Figure 4: Comparison between the online users and PromptCoT. Images are placed in pairs of (left) the online user and (right) PromptCoT.

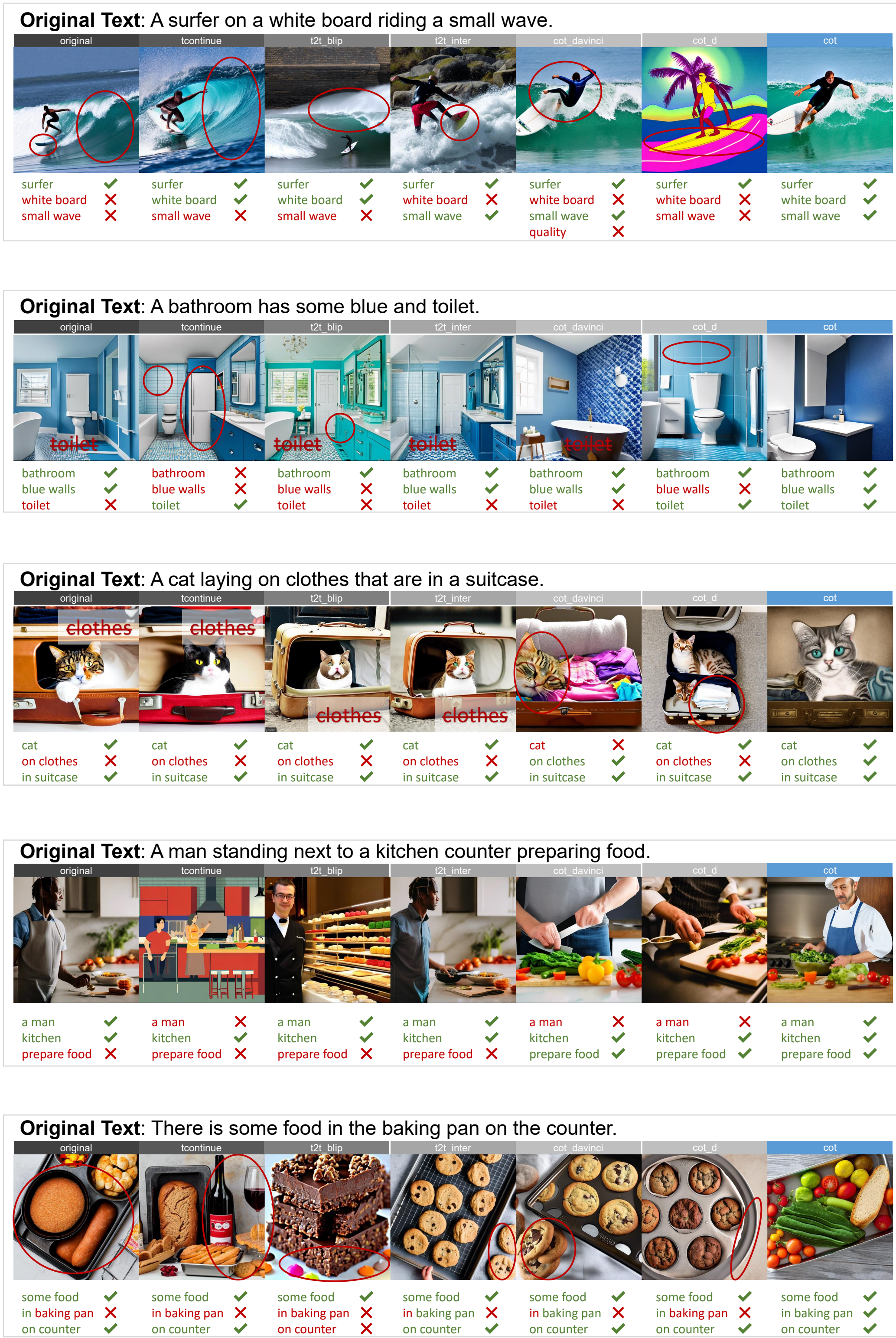

Figure 5: From left to right, images are generated via original prompts and prompts refined by tcontinue, t2t_blip, t2t_inter, cot_davinci, cot_d, and PromptCoT, respectively.

F Limitations and Societal Impact

Limitations While PromptCoT is able to enhance the generation performance of generative models by a significantly larger margin, the extent of this enhancement is reliant on the underlying capabilities of the pre-trained generative models. Additionally, if the prompts provided to the generative models are already of high quality, the further improvements brought by PromptCoT would also be limited.

Societal Impact We believe that PromptCoT is a versatile approach that can help users to improve the quality of the generation performance by a large margin on various generative applications, reducing the re-generation process and thus reducing the emission of greenhouse gases. Moreover, with lightweight adaptation, PromptCoT can be applied to multiple tasks within negligible memory overhead, providing a highly efficient once-for-all approach for industrial deployment. However, in this study, we only evaluated the effectiveness of PromptCoT in enhancing visual quality-related performance and did not address longstanding concerns related to privacy, security, and copyright issues in the field. In future research, we will explore the effectiveness of PromptCoT in addressing these concerns and ensuring the safety of generated content, while maintaining high-quality generation.

G Checklist

1. For all authors:
(a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]

(b) Did you describe the limitations of your work? [Yes]

(c) Did you discuss any potential negative societal impacts of your work? [Yes]

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results:

(a) Did you state the full set of assumptions of all theoretical results? [N/A]

(b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments:

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We will release our codebase after the double-blind review.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] The time and resources are same as the open-source assets we used.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets:

(a) If your work uses existing assets, did you cite the creators? [Yes]

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects:

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

References

[1] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, Hongsheng Li, and Yu Qiao. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023.

[2] 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.

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

[4] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.

[5] 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. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[6] 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. 2021.

[7] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[8] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

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

[10] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[11] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[12] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

[13] Zijie J. Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. DiffusionDB: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv:2210.14896 [cs]*, 2022.