Enhancing Intent Understanding for Ambiguous Prompt: A Human-Machine Co-Adaption Strategy

Anonymous ACL submission

Abstract

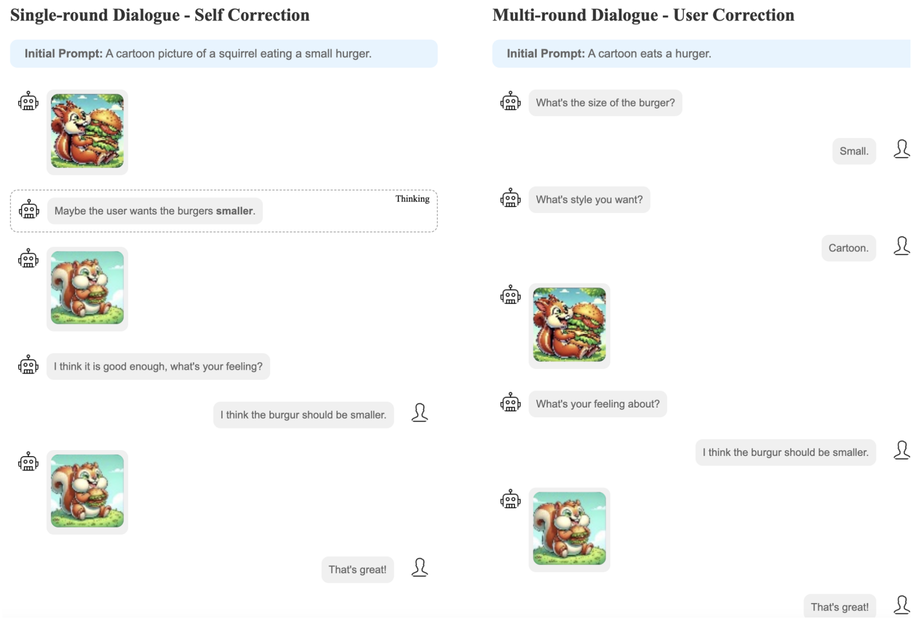
Modern text-to-image generation models are capable of producing realistic and high-quality images. However, user prompts often contain ambiguities, making it difficult for these systems to interpret users' actual intentions. Consequently, users often need to modify their prompts several times to ensure the generated images meet their expectations. Although some previous works aim to refine prompts for generating images that align with user requirements, comprehending the true needs of users, particularly non-expert individuals, remains a challenge for the model. In this research, we aim to enhance the visual parameter-tuning process, making the model user-friendly for individuals without specialized knowledge and it can better understand user needs. We propose a human-machine co-adaption strategy by maximizing the mutual information between the user's prompts and the pictures under modification as the optimizing target in order to make the system better adapt to user needs. We find that an improved model can reduce the necessity for multiple rounds of adjustments. We also collect multi-round dialogue datasets with prompts and images pairs and user intent. Various experiments demonstrate the effectiveness of the proposed method in our proposed dataset.

1 Introduction

Generative image models guided by text prompts have significantly advanced in quality and versatility over the past few years. Models like DALL·E 2 (Ramesh et al., 2022), IMAGEN (Saharia et al., 2022), Stable Diffusion (Rombach et al., 2022), and Muse (Chang et al., 2023) can produce novel and realistic images based on textual descriptions (Gozalo-Brizuela and Garrido-Merchan, 2023). Despite significant progress, there's still room for improvement, especially in generating higher-resolution images that better reflect the semantics of input text and in creating more user-friendly interfaces (Frolov et al., 2021). Many models struggle to understand nuanced human instructions, often resulting in a mismatch between user expectations and generated outputs. Additionally, the impact of variable adjustments on the final image is not always clear, posing challenges for non-expert users who haven't systematically studied prompt engineering. This complexity hinders those without technical backgrounds from fully utilizing advanced AI models. To address these challenges, we introduce an innovative approach to enhance the user experience for non-professional users. Unlike traditional models that require a deep understanding of underlying mechanisms and control elements, our approach enables users to adjust and optimize image generation with minimal technical knowledge. Inspired by human-in-the-loop co-adaptation (Reddy et al., 2022), our model evolves with user feedback to better meet user expectations. Figure 3 illustrates the operational flow as interacted by users. Our main contributions are:

- **Adaptive Prompt Engineering and Personalized Image Generation:** We propose visual co-adaptation (VCA), an adaptive framework that fine-tunes user prompts using a pretrained language model enhanced through reinforcement learning, aligning image outputs more closely with user preferences and creating images that truly reflect individual styles and intentions.

- **Human-in-the-Loop Feedback Integration:** Our work considers incorporating human feedback within the training loops of diffusion models. By assessing its impact, we demonstrate how human-in-the-loop methods can surpass traditional reinforcement learning in enhancing model performance and output quality.



Single-round Dialogue - Self Correction

Initial Prompt: A cartoon picture of a squirrel eating a small hurger.

Thinking

Maybe the user wants the burgers **smaller**.

I think it is good enough, what's your feeling?

I think the burgur should be smaller.

That's great!

Multi-round Dialogue - User Correction

Initial Prompt: A cartoon eats a hurger.

What's the size of the burger?

Small.

What's style you want?

Cartoon.

What's your feeling about?

I think the burgur should be smaller.

That's great!

Figure 1: Users have the choice between single-round dialogue, where they provide detailed inputs for the model to generate and self-adjust an image on the left, or multi-round dialogue on the right, where the model engages in iterative refinement based on user feedback, asking questions to clarify any unclear requirements. This allows for either model-driven optimization through self-reflection or user-driven customization to meet specific needs. Our proposed visual co-adaption system can successfully handle both scenarios.

- **Comparative Analysis and Tool Development for Non-Experts:** Through comparative analysis, we explore the superiority of mutual information maximization over conventional reinforcement learning in tuning model outputs to user preferences. Additionally, we introduce an interactive tool that grants non-experts easy access to advanced generative models, enabling the creation of personalized, high-quality images, thus broadening the applicability of text-to-image technologies in creative domains.

2 Related Work

2.1 Memory Mechanism for LLM-based Agents

In LLM-based agents, the memory module is considered one of the critical components for storing, processing, and retrieving information relevant to the agent's tasks. Memory plays a crucial role in determining how the agent accumulates knowledge, processes historical experiences, and supports its actions. To enhance the self-evolution capabilities of LLM-based agents, researchers are focused on designing and optimizing memory modules. Past research has explored various designs and implementations of memory modules. For example, some researchers combine information from trials and cross-trials to construct memory modules, thereby enhancing the agent's reasoning abilities. Other researchers store memory information in natural language form to improve the module's interpretability and user-friendliness. Additionally, some studies focus on designing memory read-write operations, enabling agents to interact effectively with their environment and complete tasks. Although past research has made progress in the design and implementation of memory modules, further improvement in the self-adjustment capabilities and memory management efficiency of LLM-based agents is still needed to address complex problems in real-world applications. Therefore, our approach introduces a memory optimization mechanism, allowing agents to better cope with complex and dynamic task environments.

2.2 Human Preference-Driven Optimization for Text-to-Image Generation Models

Zhong et al. (Zhong et al., 2024) significantly advance the adaptability of large language models (LLMs) to human preferences through their innovative approach. Their method utilizes SVD-based low-rank adaptation for nuanced, preference-sensitive model adjustments, eliminating the need for exhaustive model retraining. Xu et al. (Xu et al.,

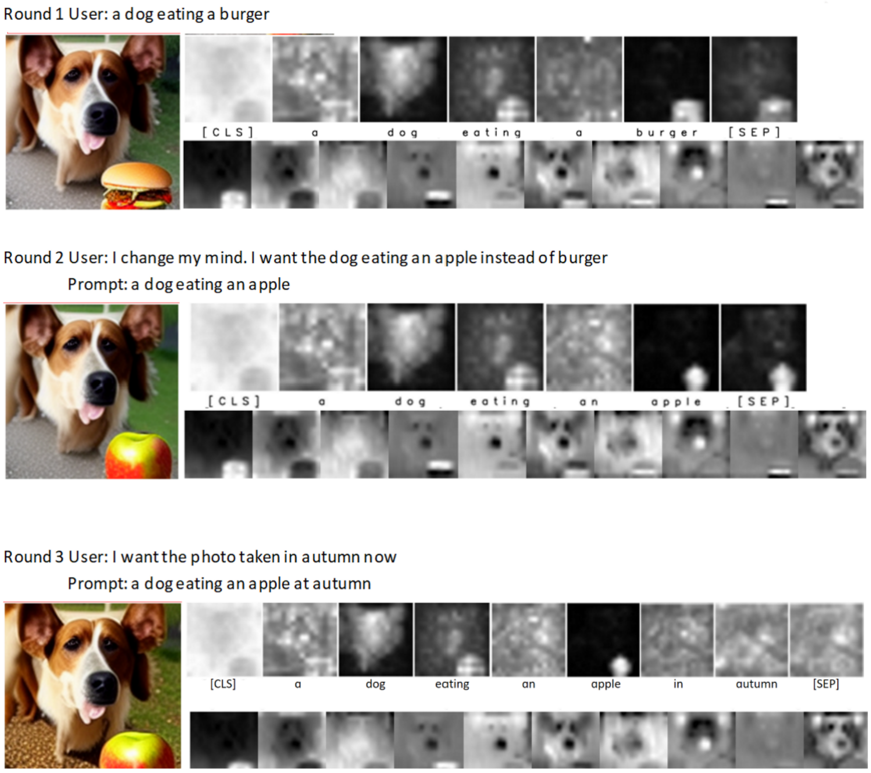

Round 1 User: a dog eating a burger
[CLS]
a
dog
eating
a
burger
[SEP]
Round 2 User: I change my mind. I want the dog eating an apple instead of burger
Prompt: a dog eating an apple
[CLS]
a
dog
eating
an
apple
[SEP]
Round 3 User: I want the photo taken in autumn now
Prompt: a dog eating an apple at autumn
[CLS]
a
dog
eating
an
apple
in
autumn
[SEP]

Figure 2: The diagram shows our model's architecture with cross attention in the first row and self attention in the second. It incorporates an improved cross attention mechanism that maintains shape consistency and aligns well with prompt tokens, enabling effective multi-round modifications based on user feedback. The model captures intricate cross attention details, optimizing parameters for progressively better single-generation performance, demonstrating few-shot learning adaptation with minimal dialogue iterations.

2024) adopt a distinctive strategy by harnessing extensive expert insights to develop their ImageReward system, setting a new benchmark for creating images that resonate deeply with human desires. Together, these advancements represent a pivotal shift towards more intuitive, user-centric LLM technologies, heralding a future where AI seamlessly aligns with the intricate mosaic of individual human expectations.

2.3 Exploration of Self-Correction Strategies

Advances in large language models (LLMs) self-correction such as, Pan et al (Pan et al., 2023), Shinn et al. (Shinn et al., 2023), Madaan et al (Madaan et al., 2024), improving language understanding and production. Huang et al (Huang et al., 2022) showcased self-debugging and zero-shot learning for reasoning evaluation, underscoring the potential and limits of self-correction. These contributions collectively highlight the progress and future challenges in enhancing LLMs' self-corrective capabilities (Hertz et al., 2022; Rosenman et al., 2023; Mehrabi et al., 2022; Xu et al., 2024). Meanwhile, we can find that multi-modal self-correction is less investigated. It is also very important to teach the vision model to think it step by step. We explore the integration of self-correction strategies into image generation to produce images that more closely align with user intentions.

2.4 Ambiguity Resolution in Text-to-Image Generation

Natural dialogue often contains ambiguity due to grammar, polysemy, and vagueness. Humans manage this ambiguity with clarifying questions and contextual cues, but machines find it challenging. To address this, text-to-image generation employs various strategies. For example, masked transformers (Chang et al., 2023) and visual annotations (Endo, 2023) help clarify prompts, while model evaluation benchmarks (Lee et al., 2024) and auto-regressive models (Yu et al., 2022) improve image alignment. Frameworks for abstract (Liao et al., 2023) and inclusive imagery (Zhang et al., 2023), as well as layout guidance (Qu et al., 2023) and feedback mechanisms (Liang et al., 2023), fur-

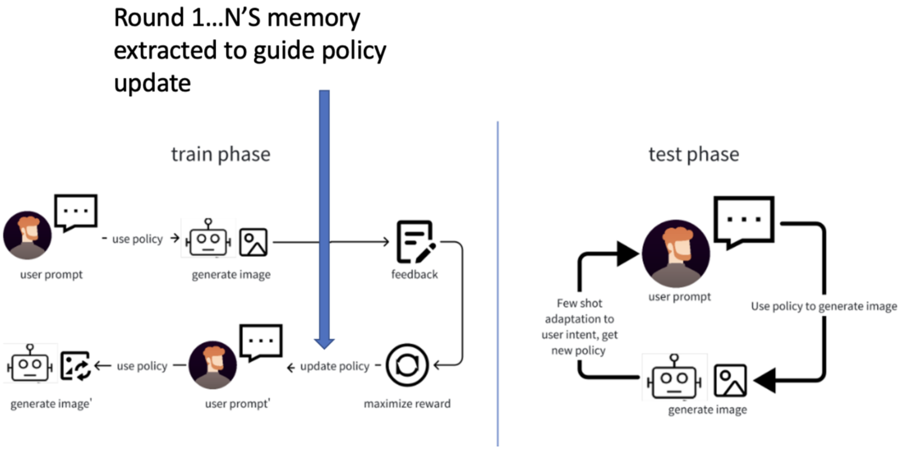


Figure 3: This figure illustrates our reinforcement learning framework. In training, the policy (three editing operations with trainable parameters, more details in section 3.1.1 and A.3) updates based on human feedback (environment), where the state is the prompt and the action is the generated image. In testing, few-shot adaptation refines the policy (π_{new}) to generate images, allowing efficient model adaptation with minimal dialogue interactions.

ther enhance quality. The TIED framework and TAB dataset (Mehrabi et al., 2023) use user interaction to refine prompt clarity. Our model integrates these techniques across multiple dialogue rounds to elicit users' true intentions, effectively reducing prompt ambiguity and generating results that align with user expectations, thus enhancing image generation quality.

3 Method

3.1 Policy Model: Controlling Cross-Attention in a Reinforcement Learning Framework

In our framework, the Imagen text-guided synthesis model (Saharia et al., 2022) constructs the basic composition and geometric layout of images at a 64×64 resolution. The model uses a U-shaped network during each diffusion step t to predict the noise component ϵ based on the text embedding $\psi(P)$ and the noise-added image z_t. Crucial to shaping the image's final appearance $I = z_0$, the attention maps $M = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)$ influence its spatial and geometric properties. Here, Q and K are the query and key matrices formed from image and text features, respectively. We define the diffusion step function $\text{DM}(z_t, P, t, s)$ that computes a single step of the diffusion process, outputting the noisy image z_{t-1} and the attention map M_t, if utilized. Overriding the attention map with an additional map M_c while maintaining the values V from the prompt is indicated as $\text{DM}(z_t, P, t, s)\{M \leftarrow M_c\}$. The modified prompt P^* generates a new attention map M_t^*, and the general edit function $\text{Edit}(M_t, M_t^*, t)$ manages the attention maps at any step t for both the original and modified images.

3.1.1 Editing Operations

In our framework, we employ three strategic editing operations—Word Swap, Adding a New Phrase, and Attention Re-weighting. Each operation is optimized through reinforcement learning (RL) as the policy model to maximize a reward function. This reward function is based on the interaction results between the action output in a specific context state and the environment (human feedback), using gradient ascent. This approach learns parameters that are highly aligned with human preferences. For more details about the RL training framework, refer to Appendix A.2.

In the **Word Swap** method, users replace tokens in the prompt (e.g., "a big red bicycle" to "a big red car"), and we control attention map injection steps to manage compositional freedom:

$$Edit(M_t, M_t^*, t) := \begin{cases} M_t^* & \text{if } t < \tau \\ M_t & \text{otherwise} \end{cases} \quad 230$$

The attention map M_t^* is updated as follows:

$$M_t^* = M_t^* + \eta \nabla_{M_t^*} \mathcal{R}(M_t^*) \quad 232$$

In the **Adding a New Phrase** method, new tokens are added to the prompt (e.g., "a castle next to a river" to "children drawing of a castle next to a river"), targeting shared tokens with an alignment function A:

$$(\text{Edit}(\text{M}_t, M_t^*, t))_{i,j} := \begin{cases} (M_t)_{i,A(j)} & \text{if } A(j) \neq \text{None} \\ (M_t)_{i,j} & \text{otherwise} \end{cases}$$

The alignment function A_t is updated as follows:

$$A_t = A_t + \eta \nabla_{A_t} \mathcal{R}(A_t)$$

In the **Attention Re-weighting** method, token influence is adjusted to enhance or diminish features (e.g., scaling the attention map of "fluffy red ball" for token j^* with a parameter $c \in [-2, 2]$):

$$(Edit(M_t, M_t^*, t))_{i,j} := \begin{cases} c \cdot (M_t)_{i,j} & \text{if } j = j^* \\ (M_t)_{i,j} & \text{otherwise} \end{cases}$$

This parameter c provides intuitive control over the induced effect. The scaling parameter c_t is updated as follows:

$$c_t = c_t + \eta \nabla_{c_t} \mathcal{R}(c_t) \tag{1}$$

Each operation refines text-image interactions through cross-attention layers, aligning outputs with human preferences. The RL framework optimizes these strategies by updating M_t, A_t, and c_t through gradient ascent. For detailed optimization processes of the three editing operations, see Appendix A.3.

3.2 Human-Machine Co-Adaptation with Mutual Information Maximization

In this section, we explain how our model can adapt to human intent. Let X denote the user inputs and Y the images generated by the model. The adaptation mechanism seeks to maximize the mutual information $I(X; Y)$, which quantifies the amount of information shared between X and Y. The mutual information is given by:

$$I(X; Y) = \int_{x \in X} \int_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \, dy \, dx, \tag{2}$$

where $p(x, y)$ is the joint probability of x and y, and $p(x)$ and $p(y)$ are the marginal probability of x and y, respectively.

Adaptive Feedback Loop

The adaptive feedback loop updates the model parameters θ to better align with human intent, utilizing the gradient of mutual information that is now conditioned on user feedback f. This feedback directly represents human preferences and intents, guiding the model towards desired outcomes:

$$\theta_{\text{new}} = \theta_{\text{old}} + \eta \nabla_\theta I(X; Y \mid f), \tag{3}$$

where η is the learning rate and f encapsulates the feedback signals from users. This adaptive approach measures effectiveness through an increase in conditional mutual information, reflecting improved alignment with user expectations, and higher user satisfaction scores in image generation tasks.

Algorithm 1 Prompt-to-Prompt Image Editing with Human-Machine Co-Adaptation (Training)

```
Input: Original prompt $P_0$, Edited prompt $P_1$, Initial image $I_0$
Output: Edited image $I_1$
Initialize interface $\pi$ with parameters $\theta$
Generate initial attention maps $A_0$ for $I_0$ using $\pi(P_0)$
Set $I_t \leftarrow I_0$
Initialize user feedback loop
for $t = 1$ to Convergence do
    Collect user feedback on image $I_t$ and prompt $P_t$
    Adapt $\pi$ (Using editing operation in Section 3.1.1) to maximize mutual information $I(A; I|P)$ incorporating feedback
    Apply $P_1$ to generate new attention maps $A_1$
    Generate $I_1$ by applying $A_1$ in diffusion step
    Evaluate $I(A; I|P)$ between $(P_0, P_1)$ and $(I_0, I_1)$
    Update $\theta$ to align more closely with user preferences
end for
Conduct final evaluation of $I_1$ with user
return$I_1$ =0
```

Algorithm 2 Evaluation of Adaptation to New User Preferences

```
Input: Trained interface $\pi$ with parameters $\theta$, New user initial prompt $P_{\text{new}}$
Output: Adapted image $I_{\text{adapted}}$ aligns with new user preferences
Initialize new user interaction session
for $i = 1$ to few-shot rounds do
    Present $I_{\text{current}}$ generated from $P_{\text{new}}$ using $\pi$
    Collect new user feedback on $I_{\text{current}}$
    Update $P_{\text{new}}$ based on user feedback
    Adapt pre-trained $\theta$ minimally to reflect new user preferences
    Generate new $I_{\text{current}}$ using updated $\pi(P_{\text{new}})$
    if user feedback is positive then
        Break the loop and finalize $I_{\text{adapted}}$
    end if
end for
Evaluate user satisfaction with $I_{\text{adapted}}$
return$I_{\text{adapted}}$ =0
```

3.3 TD Error Historical Experience Replay with Gradient Descent and Joint Gradient Ascent Training for Reward Function

Our reinforcement learning framework uses Human Feedback (E) to optimize a Text-to-Image model with Proximal Policy Optimization (PPO). The state (s_t) includes the generated image and text, while the action (a_t) is the image generation. The reward (r_t) is calculated by the CLIP model. Temporal Difference Learning computes the TD error ($\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$) to guide updates (measuring the difference between predicted and actual rewards). Prioritized Experience Replay samples experiences ((s_t, a_t, r_t, s_{t+1})) based on

TD error magnitude ($p_t \propto |\delta_t| + \epsilon$), with learning rates adjusted by $\alpha_t = \frac{1}{(n \cdot p_t)^\beta}$. New experiences have their TD error set to the maximum value to ensure priority. PPO maximizes the objective: $L^{\text{PPO}}(\theta) = \mathbb{E}_t\left[\min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}\hat{A}_t, \text{Clip}\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}, 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right]$, balancing new and old policy probability ratios. This combines reward maximization ($\max_\theta \sum_t \gamma^t r_t$) and TD error minimization ($\min_\theta \sum_t \delta_t^2$). Joint training optimizes both reward and TD error, deriving policy gradients for parameters (θ_{rew}, θ_{ref}, and θ_{rep}). The reward maximization objective ($J(\theta_{\text{rew}}, \theta_{\text{ref}}, \theta_{\text{rep}}) = \mathbb{E}_\pi\left[\sum_t \gamma^t r_t\right]$) is optimized by ascending the gradient ($\nabla_\theta J = \mathbb{E}_\pi\left[r_t \nabla_\theta \log \pi(a_t|s_t)\right]$). The TD Error Learning process involves action selection ($a_t = \pi(s_t, \theta)$), state transition ($(s_{t+1}, r_t) \sim P(s_t, a_t)$), TD error calculation ($\delta_t = r_t + \gamma V(s_{t+1}, \theta) - V(s_t, \theta)$), and parameter update ($\theta \leftarrow \theta - \alpha \nabla_\theta(\delta_t^2)$). Iterative updates minimize TD error. Integrating these objectives refines the policy for optimal performance, ensuring the generation of high-quality, text-aligned images (combining reward maximization and TD error minimization leads to better policy).

4 Experiments

4.1 Dataset

We developed QA software that annotates prompts on our platform, generating JSON files with detailed multi-turn dialogue information. An example of user interface annotations is shown in Appendix A.4. Our training set includes 1673 JSON files, annotated with prompts, QA sequences, image paths, unique identifiers, and ratings for image alignment and fidelity. This dataset instructs our model on user expectations and artistic intentions, analyzing subjects, emotions, settings, styles, perspectives, and extra elements. Feedback refines prompts, enabling the model to grasp complex artistic directions. We use 95% of the data for training and 5% for validation, supporting efficient few-shot learning to enhance performance and user satisfaction.

4.2 Comparison Study

4.2.1 Trends Across Baselines Over Iterative Rounds

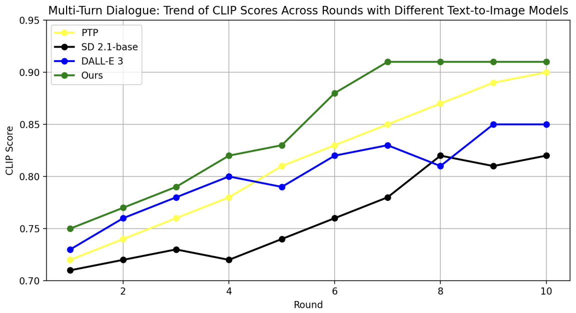


Figure 4: This graph shows CLIP score trends over 10 rounds for various text-to-image models (PTP (Hertz et al., 2022), SD 2.1-base, DALL-E 3, and ours)

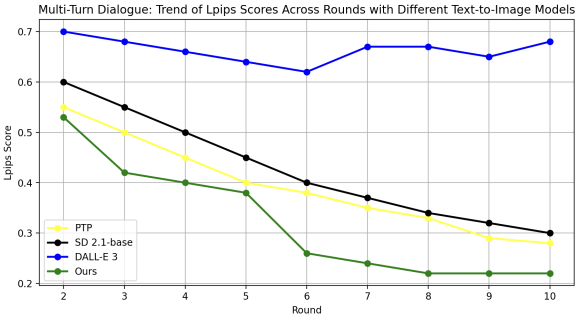


Figure 5: Illustrated in the graph are the trends of LPIPS scores for several text-to-image models (PTP, SD 2.1-base, DALL-E 3, and ours) over 10 rounds.

Figure 4 showcases our model's superior performance on a validation prompt describing "A serene ancient fantasy sanctuary constructed of stone, with white birds flying in the distance." and achieves high CLIP scores early, our model reaches 0.78 by round 3 and peaks at 0.91 by round 7, surpassing competitors. It also excels in Lpips, as is shown in Figure 5 recording a score of 0.42 by round 3 and stabilizing at 0.22 by round 8. This rapid stabilization highlights our model's adaptability and efficiency, maintaining high consistency and user satisfaction across fewer dialogue rounds. Each round incrementally builds on the last, refining details without altering the prompt's core structure.

4.2.2 Prompt Refinement

Table 1 compares **self-reflection prompt refinement** and **multi-round dialogue prompt refinement**. Self-reflection is faster (3.4s vs. 12s), but multi-round dialogue better captures user preferences, leading to higher satisfaction (4.7 vs. 3.0). It also shows improved Purpose Adaptability (4.8 vs. 3.3), Clarity (4.7 vs. 4.2), and Detail Level (4.2 vs. 4.1). For algorithm details, see Appendix A.7.

4.3 Ablation Study: Reinforcement Learning for Parameter Tuning

Table 2 highlights the impact of Reinforcement Learning (RL) tuning on dialogue system perfor-

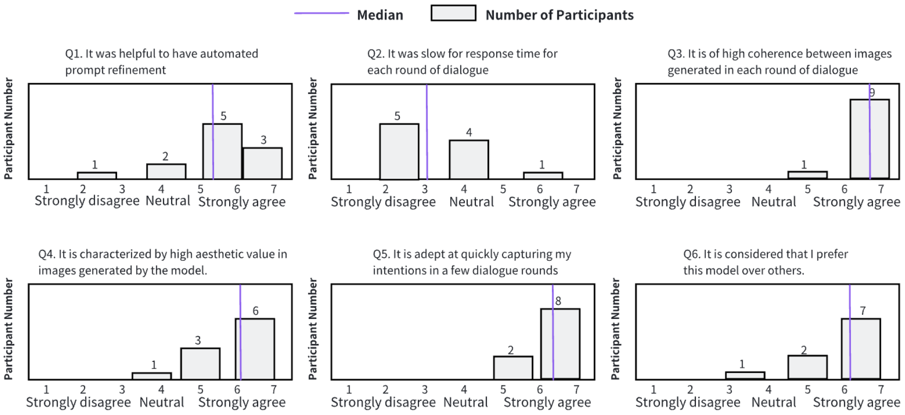


Figure 6: The chart shows user feedback on a model, highlighting mixed responses with positive feedback on image coherence and capturing intentions, but concerns over response time.

Table 1: Comparative Analysis of Prompt Refinement from 100 users, averaged and rounded to one decimal. Metrics are scored on a 0-5 scale. Response Time indicates average duration for self-reflection and multi-dialogue processes.

Metric & Category	Refine Type	
	Self-reflection	Multi-dialogue
Prompt Quality		
Clarity	4.2/5	**4.7**/5
Detail Level	4.1/5	**4.2**/5
Purpose Adaptability	3.3/5	**4.8**/5
Image Reception		
User Satisfaction	3.0/5	**4.7**/5
CLIP Value	0.8/1	**0.9**/1
Response Time	**3.4**s	12s

Table 2: Ablation result on the effects of RL using data averaged from randomly selected 10 users, with final interaction CLIP and Aesthetic Scores.

Metrics	With RL	Without RL
Rounds	**4.3**	6.9
CLIP Score	**0.92**/1.0	0.83/1.0
User Satisfaction	**4.73**/5	4.14/5
Aesthetic Score	**4.89**/5	4.88/5

Table 3: Ablation results for edited cross attention (CA), averaging data from randomly selected 10 users, with CLIP and Aesthetic Scores from the final interaction.

Metrics	Edited CA	Normal CA
Rounds	**3.7**	6.1
CLIP Score	**0.88**/1.0	0.81/1.0
User Satisfaction	**4.82**/5	3.94/5
Aesthetic Score	**4.71**/5	4.48/5

mance. RL systems require fewer dialogue rounds (4.3 vs. 6.9), showing greater efficiency. The CLIP score improves from 0.83 to 0.92, indicating better alignment of images with prompts. User satisfaction increases from 4.14 to 4.73 out of 5, reflecting a better user experience. Both systems perform similarly in aesthetic quality (4.89 vs. 4.88), but RL tuning enhances functionality and user satisfaction. Users noted lower consistency in image quality from non-RL-tuned models, emphasizing RL's effectiveness in dynamically adapting to user feedback. For detailed parameter updates with RL tuning, see Appendix A.5.

4.4 Ablation Study: Comparing Edited Cross Attention with Normal Cross Attention.

Table 3 highlights the superior performance of edited cross attention (CA) over normal CA in dialogue systems, emphasizing their distinct adaptability. Normal CA computes static attention weights, while edited CA dynamically adjusts these weights in response to dialogue context and user feedback. This adaptability reduces dialogue rounds to an average of 3.7 compared to 6.1 for normal CA, enhancing system performance. For instance, edited CA achieves a higher CLIP score of 0.88 versus 0.81 and increases user satisfaction from 3.94 to 4.82 out of 5. The aesthetic quality of images also improves with edited CA, scoring 4.71 compared to 4.48 for normal CA. These results underscore the effectiveness of integrating reinforcement learning with edited CA to refine tuning and improve output consistency and relevance in denoising tasks. For an in-depth exploration of edited cross attention mechanisms, refer to Appendix A.6.

4.5 Visualization Results

Dialogue Rounds Across Different Models

Figure 2 compares dialogue rounds across different

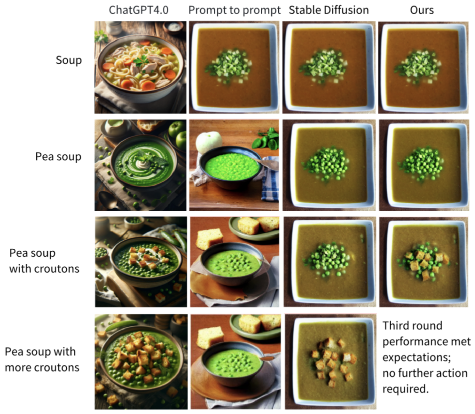

ChatGPT4.0
Prompt to prompt
Stable Diffusion
Ours
Soup
Pea soup
Pea soup with croutons
Pea soup with more croutons
Third round performance met expectations; no further action required.

Figure 7: The comparison demonstrates our model's few-shot learning capability, effectively adapting to user preferences with minimal dialogue.

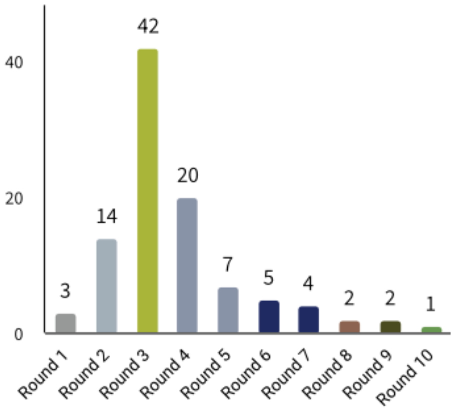


Figure 8: The chart shows the rapid decline in user interaction rounds needed for satisfaction, peaking by Round 5, demonstrating the model's efficient few-shot learning.

models: ChatGPT, Stable Diffusion v2.1, Prompt-to-Prompt (Hertz et al., 2022), and our model. Initially, images from Stable Diffusion, Prompt-to-Prompt, and our model are similar due to the lack of feedback. By the second round, "pea soup" preferences cause significant changes in ChatGPT-4 and Stable Diffusion, affecting consistency. In the third round, with croutons added, our model excels by fine-tuning parameters via reinforcement learning, maintaining balance, while Prompt-to-Prompt struggles, and ChatGPT-4 shows inconsistencies. By the fourth round, our model achieves satisfactory results and opts out, while the others continue ineffective adjustments. This highlights our model's superior ability to understand and respond to user feedback, achieving optimal results by the third round and demonstrating effective multi-round dialogue learning. Despite ChatGPT-4's realistic visuals, it struggles with consistency and adapting to human preferences. Our model, preferred by 89% of users, effectively adapts with minimal dialogue.

User Satisfaction Distribution for Our Model Over Multiple Rounds

Figure 8 illustrates our model's efficiency in adapting to user feedback. Initially, the satisfaction rate increases rapidly, with 59 users satisfied by Round 3, demonstrating the model's quick alignment with user preferences. By Round 5, satisfaction peaks at 99 out of 100 users, underscoring the model's effectiveness in achieving high user satisfaction swiftly.

Users' Overall Evaluation of Our Model

Figure 6 presents user evaluations across various model aspects. The majority found the automated prompt refinement to be helpful, indicating approval. In contrast to typical concerns about speed in models with complex computations, most users disagreed with the notion that the model's response time per dialogue round was slow, suggesting that the integration of reinforcement learning for fine-tuning did not significantly impact perceived efficiency. The model was highly praised for its coherence across images generated in each dialogue round and received commendations for aesthetic quality. It was also recognized for adeptly capturing user intentions within just a few rounds of dialogue. Overall, the participants showed a strong preference for this model over others, reflecting its effectiveness and user satisfaction.

5 Conclusion and Future Work

In this study, we introduced a new image generation method using a human-in-the-loop approach that enhances user interaction and responsiveness to ambiguous prompts. Our findings highlight the model's ability to closely match user expectations through adaptive prompt engineering and mutual information optimization. Looking ahead, we plan to release our training dataset, improving transparency and enabling broader testing. Additionally, we aim to refine the model's interpretive skills, expand its applications across different domains, and conduct comprehensive benchmarks to gauge the alignment between user intentions and generated images. These initiatives will advance personalized and intuitive image generation technologies, making advanced modeling tools more accessible without requiring deep technical expertise.

6 Limitation

The study's limitations mainly involve the model's reliance on user feedback and its generalization capabilities. The model may struggle with highly ambiguous or contextually complex prompts, especially those needing subtle cultural nuances or specialized knowledge. Its performance relies heavily on iterative user feedback, which may not always be practical or available. This dependency could limit the model's applicability in scenarios requiring rapid, autonomous decision-making, restricting its utility in diverse or less interactive environments where adaptability and minimal human intervention are crucial.

References

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

Yuki Endo. 2023. Masked-attention diffusion guidance for spatially controlling text-to-image generation. *The Visual Computer*, pages 1–13.

Stanislav Frolov, Tobias Hinz, Federico Raue, Jörn Hees, and Andreas Dengel. 2021. Adversarial text-to-image synthesis: A review. *Neural Networks*, 144:187–209.

Roberto Gozalo-Brizuela and Eduardo C Garrido-Merchan. 2023. Chatgpt is not all you need. a state of the art review of large generative ai models. *arXiv preprint arXiv:2301.04655*.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. 2022. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*.

Tony Lee, Michihiro Yasunaga, Chenlin Meng, Yifan Mai, Joon Sung Park, Agrim Gupta, Yunzhi Zhang, Deepak Narayanan, Hannah Teufel, Marco Bellagente, et al. 2024. Holistic evaluation of text-to-image models. *Advances in Neural Information Processing Systems*, 36.

Youwei Liang, Junfeng He, Gang Li, Peizhao Li, Arseniy Klimovskiy, Nicholas Carolan, Jiao Sun, Jordi Pont-Tuset, Sarah Young, Feng Yang, et al. 2023. Rich human feedback for text-to-image generation. *arXiv preprint arXiv:2312.10240*.

Jiayi Liao, Xu Chen, Qiang Fu, Lun Du, Xiangnan He, Xiang Wang, Shi Han, and Dongmei Zhang. 2023. Text-to-image generation for abstract concepts. *arXiv preprint arXiv:2309.14623*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Ninareh Mehrabi, Palash Goyal, Apurv Verma, Jwala Dhamala, Varun Kumar, Qian Hu, Kai-Wei Chang, Richard Zemel, Aram Galstyan, and Rahul Gupta. 2022. Is the elephant flying? resolving ambiguities in text-to-image generative models. *arXiv preprint arXiv:2211.12503*.

Ninareh Mehrabi, Palash Goyal, Apurv Verma, Jwala Dhamala, Varun Kumar, Qian Hu, Kai-Wei Chang, Richard Zemel, Aram Galstyan, and Rahul Gupta. 2023. Resolving ambiguities in text-to-image generative models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14367–14388.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *arXiv preprint arXiv:2308.03188*.

Leigang Qu, Shengqiong Wu, Hao Fei, Liqiang Nie, and Tat-Seng Chua. 2023. Layoutllm-t2i: Eliciting layout guidance from llm for text-to-image generation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 643–654.

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

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

Siddharth Reddy, Sergey Levine, and Anca Dragan. 2022. First contact: Unsupervised human-machine co-adaptation via mutual information maximization. *Advances in Neural Information Processing Systems*, 35:31542–31556.

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

Shachar Rosenman, Vasudev Lal, and Phillip Howard. 2023. Neuroprompts: An adaptive framework to optimize prompts for text-to-image generation. *arXiv preprint arXiv:2311.12229*.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494.

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

Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning.(2023). *arXiv preprint cs.AI/2303.11366*.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. 2024. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. 2022. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5.

Cheng Zhang, Xuanbai Chen, Siqi Chai, Chen Henry Wu, Dmitry Lagun, Thabo Beeler, and Fernando De la Torre. 2023. Iti-gen: Inclusive text-to-image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3969–3980.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595.

Yifan Zhong, Chengdong Ma, Xiaoyuan Zhang, Ziran Yang, Qingfu Zhang, Siyuan Qi, and Yaodong Yang. 2024. Panacea: Pareto alignment via preference adaptation for llms. *arXiv preprint arXiv:2402.02030*.

A Appendix

A.1 Reinforcement Learning configuration

To train our policy model, we employ Proximal Policy Optimization (PPO) (Schulman et al., 2017), initializing the value and policy networks from a supervised fine-tuned model. We use diverse beam search (Vijayakumar et al., 2016) with a beam size of 8 and a diversity penalty of 1.0 to ensure exploration quality and diversity. The maximum generation length is randomly set between 15 to 75 at each step, and one completion is randomly selected to update the policy. Each prompt generates one image, computing the clip score as the reward function to reduce variance. Training involves 12,000 episodes, four PPO epochs per batch, a batch size of 256, and a learning rate of 5e-5, with value and KL reward coefficients set at 2.2 and 0.3, respectively. Based on human fragmented language feedback, ChatGPT provides new prompts with minimal structural changes but reflects human intent very well.

A.2 Reinforcement Learning Framework

The reinforcement learning framework for our human-machine co-adaptation system in image editing involves the following elements:

State (S)

The state in our framework represents the current situation of the system, which includes:

- The current image I_t being edited.
- The current prompt P_t describing desired modifications or features in the image.
- Optionally, it can also include historical user interactions and feedback to provide context to the state, enabling the model to better understand and predict user preferences.

Action (A)

Actions in this context refer to the modifications applied to the image based on the input prompt and model's interpretation:

- Adjustments or transformations applied to the image I_t to generate a new image I_{t+1}.
- These actions are driven by the interpretation of the user's prompt, potentially influenced by machine learning algorithms that predict optimal changes.

Reward (R)

The reward function is crucial as it guides the training of the RL model by quantifying the success of actions taken based on the state:

- It could be defined using objective metrics such as the similarity between the generated image and user's expected outcome, measured by tools like CLIP score.
- Feedback from users after viewing the modified image can also be used as part of the reward, where positive feedback increases the reward and negative feedback decreases it.
- The reward aims to maximize the alignment between the user's intent and the image output, effectively training the model to interpret and act upon ambiguous prompts accurately.

This reinforcement learning setup enables our system to iteratively learn and adapt from each user interaction, improving its ability to decode ambiguous prompts and align image outputs with user expectations.

A.3 Optimization Details

To optimize image generation, the model dynamically selects among three strategies (adding phrases, word swapping, re-weighting) using the CLIP score as the reward function to update all parameters of the chosen strategy. This feedback-driven approach optimizes parameters within one strategy per iteration, yielding three well-adjusted parameter sets that adapt image generation to human preferences. The strategies correspond to three controllers: Attention-Replace, Attention-Refine, and Attention-Reweight. Our text-to-image model uses controllers to adjust cross-attention during generation, with each controller utilizing cross-attention information between images and prompts in each dialogue round. The controllers correspond to three strategies with trainable parameters, including the dynamic proportion of self-attention during the sampling process, the proportion of attention injection steps, and adaptive updates to cross-attention maps based on dialogue feedback. The optimization process for parameter updates can be mathematically represented as follows:

Reward function:

This is computational framework for the reward function $\mathcal{R}(\theta)$ in a reinforcement learning context, where the CLIP score assesses the similarity between generated images and textual prompts. Specifically:

$$\mathrm{R}(\theta) = \mathrm{CLIPScore}(I_{\text{gen}}, P_{\text{prev}}) + \lambda \cdot \mathrm{CLIPScore}(I_{\text{gen}}, P_{\text{new}}) \tag{4}$$

This formula ensures that the parameters are finely tuned, with λ serving as a balancing factor between aligning the generated image with the previous prompt and the new prompt, fostering both continuity and responsiveness to new requirements. Extensive experimentation has determined that setting $\lambda = 0.2$ is optimal, as it allows the CLIP score to converge more rapidly to its maximum value. When incrementally increasing λ from 0.1 to 1, the performance peaks at 0.2. However, increasing λ beyond 1 leads to a significant decline in performance, falling even below the levels observed at $\lambda = 0.1$. Further, to underscore the iterative update mechanism integral to the reinforcement learning cycle:

$$I_{\text{gen}}^{(k+1)} = \text{Update}(I_{\text{gen}}^{(k)}, \theta^{(k)})$$

Here, $I_{\text{gen}}^{(k)}$ signifies the image generated at iteration k, and $\theta^{(k)}$ indicates the parameters at that iteration. The update function modifies the image based on the current parameters, capturing the dynamic nature of the learning process across successive rounds.

Attention-Replace Strategies: update method directly adjusts the mapping matrix M using gradient ascent and then multiplies it with the cross-attention matrix $M_{\text{cross\_attention}}$ called `mapper` to alter the attention distribution, impacting the generated image's features and quality.

$$M_{\text{new}} = (M + \eta \cdot \Delta M) \cdot M_{\text{cross\_attention}} \tag{5}$$

Attention-Refine Strategies: Update the attention weights by combining the original and new attention maps derived from the modified prompt. In the `Attention-Refine` class, the `mapper` aligns base attention weights with the new prompt structure while `alphas` blend original and modified weights, ensuring the final output accurately reflects user modifications and maintains consistency. The `mapper` tensor aligns tokens between prompts, enabling correct transfer of attention weights; updated as

$$\theta'_m = \theta_m + \eta \nabla_{\theta_m} \mathbb{E}[R]$$

to maximize the expected reward ($\mathbb{E}[R]$) using gradient ascent with learning rate η. The `alphas` weights control the blending of original and modified attention weights, determining each token's influence; updated as

$$\theta'_\alpha = \theta_\alpha + \eta \nabla_{\theta_\alpha} \mathbb{E}[R]$$

to maximize the expected reward ($\mathbb{E}[R]$) using gradient ascent with learning rate η. Sure, here is the updated explanation and mathematical representation:

The attention weights are updated by combining the original and new attention maps derived from the modified prompt. The original attention is processed using the `mapper`, which aligns the attention weights by permuting dimensions based on the mapped indices:

$$\text{attn\_base\_replace}_{ijk} = \text{attn\_base}_{ijk} \cdot \text{mapper}_{kj}$$

$$\implies (\text{attn\_base\_replace})_{permute(2,0,1,3)}$$

Here, mapper_{kj} indicates the mapping from index k in the original prompt to index j in the new prompt. The operation $(\text{attn\_base\_replace})_{permute(2,0,1,3)}$ permutes the dimensions of the resulting tensor to align with the expected structure for further processing. The updated attention weights are calculated as:

$$M^{(t)}_{\text{update}} = \beta_t \cdot M^{(t)}_{\text{orig}} + (1 - \beta_t) \cdot M^{(t)}_{\text{new}} \tag{6}$$

Attention-Reweight Strategies:modifies the distribution of attention by first blending the original and new attention maps, and then scaling the weights according to user preferences. The blending of attention maps is given by:

$$M^{(t)}_{\text{refine}} = \beta_t \cdot M^{(t)}_{\text{orig}} + (1 - \beta_t) \cdot M^{(t)}_{\text{new}}, \quad \beta_t = \beta_{t-1} + \gamma \cdot \nabla_{\beta_t} \mathcal{R}(\theta) \tag{7}$$

with β_t adjusting the blending ratio dynamically based on feedback, and γ is the learning rate for β_t. After blending, the attention distribution is further modified by scaling the weights:

$$M^{(t)}_{\text{reweight}} = \sum_i \gamma_{t,i} \cdot M^{(t,i)}_{\text{refine}}, \quad \gamma_{t,i} = \gamma_{t-1,i} + \kappa \cdot \nabla_{\gamma_{t,i}} \mathcal{R}(\theta) \tag{8}$$

where $\gamma_{t,i}$ are the weight multipliers that adapt the emphasis on specific features, and κ is the learning rate for $\gamma_{t,i}$. Below is the pseudocode:

In addition to these, we also update the proportions related to specific attention mechanisms:

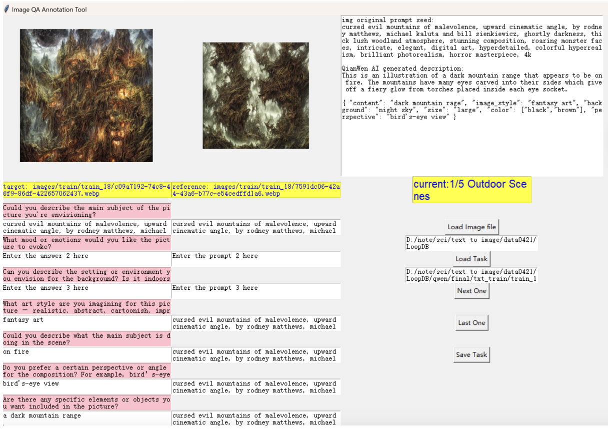
$$\alpha_{t+1} = \alpha_t + \eta \nabla_{\alpha_t} \mathcal{R}(\theta) \tag{9}$$

$$\zeta_{t+1} = \zeta_t + \gamma \nabla_{\zeta_t} \mathcal{R}(\theta) \tag{10}$$

$$\delta_{t+1} = \delta_t + \kappa \nabla_{\delta_t} \mathcal{R}(\theta) \tag{11}$$

Here, α represents the proportion of self-attention features injected at different stages of the sampling process, ζ represents the replacement proportion of the cross-attention map, and δ represents the overall number of sampling steps.

A.4 Q&A Software Annotation Interface



img original prompt seed:
cursed evil mountains of malevolence, upward cinematic angle, by rodney matthews, michael kaluta and bill sienkiewicz, ghostly darkness, thick lush woodland atmosphere, stunning composition, roaring monster faces, intricate, elegant, digital art, hyperdetailed, colorful hyperrealism, brilliant photorealism, horror masterpiece, 4k

QianWen AI generated description:
This is an illustration of a dark mountain range that appears to be on fire. The mountains have many eyes carved into their sides which give off a fiery glow from torches placed inside each eye socket.

{ "content": "dark mountain rage", "image_style": "fantasy art", "background": "night sky", "size": "large", "color": ["black","brown"], "perspective": "bird's-eye view" }

target: images/train/train_18/c09a7192-74c8-46f9-86df-422657062437.webp

reference: images/train/train_18/7591dc06-42a4-43a6-b77c-e54cedffd1a6.webp

current:1/5 Outdoor Scenes

D:/note/sci/text to image/data0421/LoopDB

D:/note/sci/text to image/data0421/LoopDB/qwen/final/txt_train/train_1

Could you describe the main subject of the picture you're envisioning?
cursed evil mountains of malevolence, upward cinematic angle, by rodney matthews, michael | cursed evil mountains of malevolence, upward cinematic angle, by rodney matthews, michael

What mood or emotions would you like the picture to evoke?
Enter the answer 2 here | Enter the prompt 2 here

Can you describe the setting or environment you envision for the background? Is it indoors
Enter the answer 3 here | Enter the prompt 3 here

What art style are you imagining for this picture — realistic, abstract, cartoonish, impr
fantasy art | cursed evil mountains of malevolence, upward cinematic angle, by rodney matthews, michael

Could you describe what the main subject is doing in the scene?
on fire | cursed evil mountains of malevolence, upward cinematic angle, by rodney matthews, michael

Do you prefer a certain perspective or angle for the composition? For example, bird' s-eye
bird's-eye view | cursed evil mountains of malevolence, upward cinematic angle, by rodney matthews, michael

Are there any specific elements or objects you want included in the picture?
a dark mountain range | cursed evil mountains of malevolence, upward cinematic angle, by rodney matthews, michael

Figure 9: Screenshot of the Q&A software annotation interface.

A.5 Ablation of RL tuning

The RL tuning process and static parameter configuration are mathematically represented as:

$$\theta^{\text{RL}} = \theta_0 + \sum_{t=1}^{T} \eta \nabla_\theta \mathcal{R}(\theta_t), \quad \theta^{\text{Fixed}} = \theta_0 \tag{12}$$

Here, θ^{RL} are the parameters iteratively updated with RL, θ_0 is the initial parameter setting, η is the learning rate, and $\nabla_\theta \mathcal{R}(\theta_t)$ is the gradient of the reward function at iteration t. This setup without RL results in more dialogue rounds and less optimal outcomes.

A.6 Ablation of cross attention control

$$\theta^{(t+1)}_{\text{Weighted}} = \theta^{(t)}_{\text{Weighted}} + \eta \nabla_\theta \mathcal{L}(I_t, \text{Feedback}_t, M) \tag{13}$$

$$\theta^{(t+1)}_{\text{Empty}} = \theta^{(t)}_{\text{Empty}} + \eta \nabla_\theta \mathcal{L}(I_t, \text{Feedback}_t, M_{\text{new}}) \tag{14}$$

This setup employs only new attention without blending it with the base cross attention. Each strategy involves a distinct function to modify the cross attention map, directed by its corresponding controller. For standard cross attention, the controller is set to 'empty control' within the code.

A.7 LLM Prompt Refinement

The Multi-dialogue Refine process in ChatGPT-4 iteratively refines prompts until they meet predefined conditions and are ambiguity-free. Initially,

Algorithm 3 Multi-dialogue Prompt Refine Process for ChatGPT-4

```
Input: Initial prompt $p_0$
Output: Refined prompt $p_i$ that meets conditions and is ambiguity-free
Define $C(p)$: Checks if prompt $p$ meets all predefined conditions.
Define $A(p)$: Checks if prompt $p$ is free of ambiguities.
$i \leftarrow 0$
while $\neg C(p_i) \vee \neg A(p_i)$ do
    if $\neg A(p_i)$ then
        $p_{i+1} \leftarrow$ ResolveAmbiguities$(p_i)$ {Clarify prompt, ensuring clarity.}
    else if $\neg C(p_i)$ then
        $p_{i+1} \leftarrow$ ModifyToMeetConditions$(p_i)$ {Adjust prompt to meet conditions.}
    end if
    $i \leftarrow i + 1$
end while
return $p_i$ =0
```

the model assesses if the prompt p_0 meets specific criteria and lacks ambiguities. If issues are identified, the process loops to rectify them. The model evolves with each iteration, described mathematically as:

$$y_{t+1} = M(p_{\text{refine}} \parallel x \parallel y_0 \parallel \text{fb}_0 \parallel \ldots \parallel y_t \parallel \text{fb}_t),$$

where y_t is the output at iteration t, M represents the model, p_{refine} is the refined prompt, x is the input data, and fb_t is the feedback at iteration t. The model refines prompts by engaging in multi-turn dialogue, asking clarifying questions until the prompts are comprehensive and unambiguous. This self-reflection mechanism allows the model to produce initial responses and evaluate them for retrieval, relevance, support, and utility. Necessary modifications are made based on feedback to enhance accuracy and usefulness, represented as:

$$y_{t+1} = M(x \parallel y_t \parallel \text{fb}_t).$$

A.8 Experiments Settings

The experiments are conducted using 4 NVIDIA 4090 GPUs, This setup allows us to utilize complex algorithms such as diverse beam search with a beam size of 8 and a diversity penalty of 1.0, ensuring thorough exploration and diversity in the generated responses. The model parameters are initialized from a fine-tuned baseline, which provides a robust starting point for further optimization. Over three days of training session, which encompass 12,000 episodes, with four PPO epochs per batch and a batch size of 256. The learning rate is set at 5×10^{-5}, and the value and KL reward coefficients are meticulously calibrated to 2.2 and 0.3, respectively, to balance the learning dynamics. For additional details due to page constraints, see Appendix A.1.

A.9 Evaluation Metrics

The experimental framework of this study is meticulously designed to evaluate our text-to-image generation model across three key dimensions.

LPIPS (Zhang et al., 2018): is a deep learning metric that evaluates how image modifications preserve the original structure, with lower scores indicating minimal visual differences and alignment with human perception. It measures the consistency and perceptual coherence of images generated in successive dialogue rounds.

CLIP Score (Radford et al., 2021): Based on the CLIP model, the system evaluates image-text alignment, assigning scores from 0 (no similarity) to 1 (perfect alignment). In dialogues, the LLM subtly adjusts prompts and selects one of three strategies following user feedback. The text-to-image model, using reinforcement learning and CLIPScore, iteratively refines images until reaching a satisfactory score. For detailed information on how the ChatGPT-4 modifies prompts based on human input, refer to the Appendix A.7.

Human Evaluation: In a study with 100 diverse users, we utilize a randomized control trial with stratified sampling based on age, gender, and technical proficiency. Using a blind design, participants are unaware of the models or components being tested to prevent biases. Detailed feedback is collected through electronic surveys post-interaction, utilizing standardized forms with scaled and open-ended questions. A cross-over design ensures that each user experiences all model variations in a randomized order, maximizing exposure. Statistical power analysis confirms that 100 participants provide sufficient power to detect significant results.