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

Anonymous ACL submission

Abstract

Today's image generation systems 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, many users must modify their prompts several times to ensure the generated images meet their expectations. While some methods focus on enhancing prompts to make the generated images fit user needs, the model is still hard to understand users' real needs, especially for non-expert users. In this research, we aim to enhance the visual parameter-tuning process, making the model user-friendly for individuals without specialized knowledge and better understand user needs. We propose a human-machine co-adaption strategy using mutual information between the user's prompts and the pictures under modification as the optimizing target 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 AI holds great promise for boosting economic development by streamlining both creative and non-creative tasks. Models like DALL·E 3 (Betker et al., 2023), Imagen (Saharia et al., 2022), Stable Diffusion (Esser et al., 2024), and Cogview 3 (Zheng et al., 2024) excel at generating lifelike images from text (Gozalo-Brizuela and Garrido-Merchan, 2023). However, challenges remain in producing higher-resolution images, improving semantic alignment, and enhancing user-friendly interfaces (Frolov et al., 2021), as models often struggle with nuanced human instructions. Additionally, the impact of variable adjustments

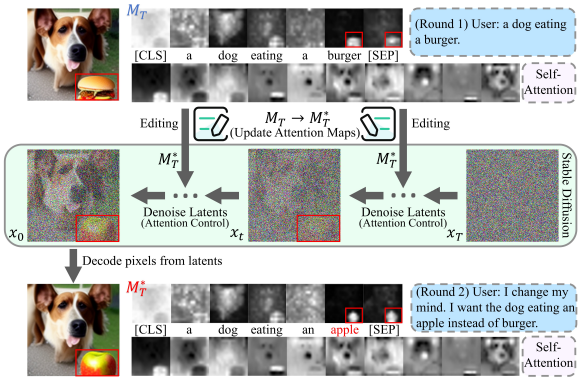

Figure 1: The user initially requests an image of a dog eating a burger (Round 1), then revises it to a dog eating an apple (Round 2). The system updates the attention maps during the denoising process.

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 2 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 pre-trained language model enhanced through reinforcement learning, aligning image outputs more closely with user preferences and creating images that truly reflect individual styles

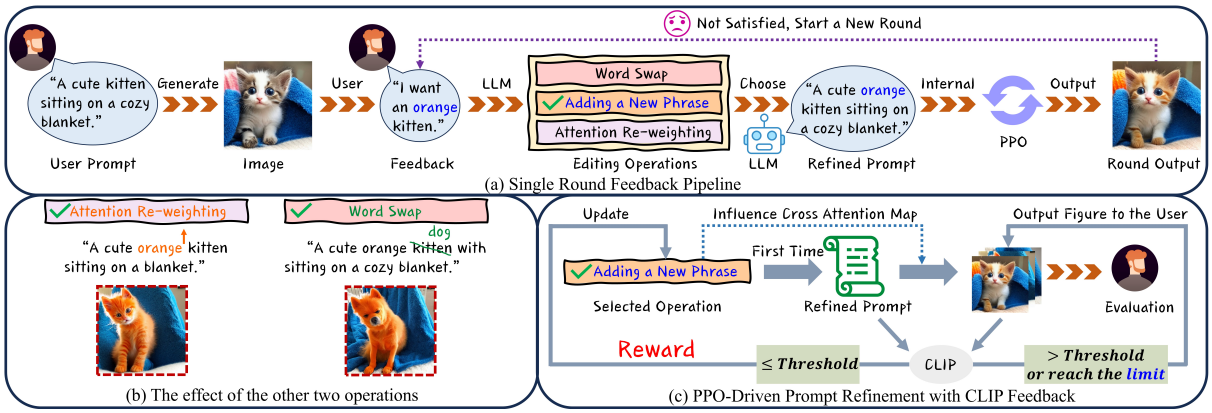

Figure 2: Overview of the overall framework: (a) The LLM refines the user's prompt through Word Swap, Adding a Phrase, or Attention Re-weighting, using PPO until satisfaction; (b) Effects of these operations on the image are demonstrated; (c) PPO's internal loop computes rewards based on CLIP feedback to meet the threshold or limit.

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.

- **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 Text-Driven Image Editing Framework:

Image editing is fundamental in computer graphics, with textual prompts providing an intuitive way for users to edit images. Recent advancements in text-to-image generation focus on aligning models with human preferences, using feedback for image refinement. Studies like Hertz et al. (Hertz et al., 2022a)'s framework leverage diffusion models' cross-attention layers for high-quality, prompt-driven modifications. Methods like ImageReward (Xu et al., 2024) develop reward models based on human preferences, collecting rich feedback (Wu et al., 2023; Liang et al., 2023) and training models for better image-text alignment and adaptability (Lee et al., 2023) to diverse preferences. However, these methods often lack convenient and efficient editing capabilities. For instance, Hertz et al.'s framework requires users to adjust complex parameters such as cross-attention and attention weights, demanding high professional knowledge. Unlike traditional image editing, ImageReward (Xu et al., 2024) generates a new image instead of editing the existing one, failing to preserve previous information and risking invalidating prior modifications.

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 leverages advanced mathematical techniques for nuanced, preference-sensitive model adjustments, eliminating the need for exhaustive model retraining. Xu et al. (Xu et al., 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.

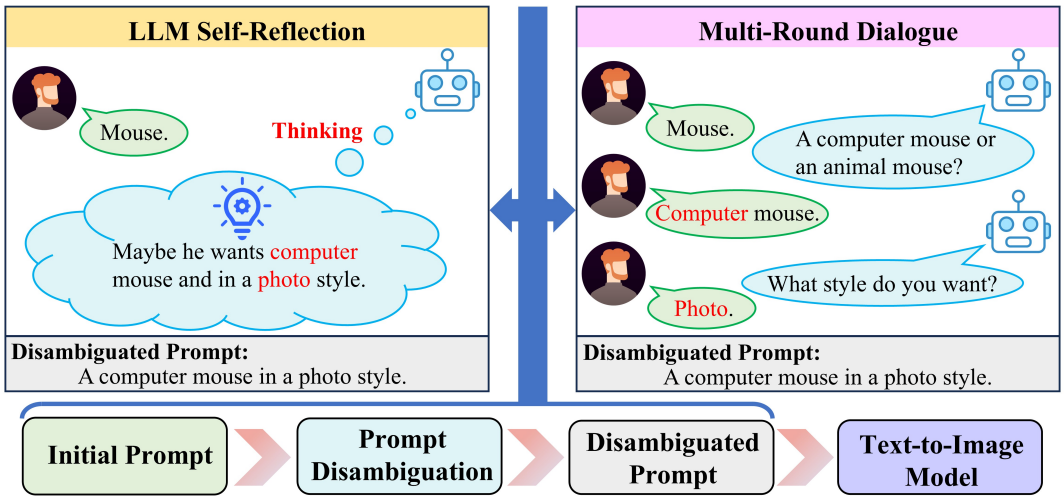

Figure 3: Resolving Ambiguous Prompts: Addressing Insufficient Context and Semantic Uncertainty through Clarifying Questions and Visual Setups.

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., 2022a; 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.

3 Method

3.1 Mutual Information for Co-Adaptation

Our system adapts to user preferences by maximizing mutual information $I(X;Y)$ between user inputs (prompts and feedback) and generated images, refining its output each round to better align with the user's intent. Mutual information $I(X;Y)$ measures how much the generated image Y reveals about the user input X, guiding the system to better predict user preferences by maximizing this information:

$$I(X;Y) = \int_x \int_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \, dy \, dx, \tag{1}$$

where $p(x,y)$ is the joint probability of x and y, while $p(x)$ and $p(y)$ are the marginal distributions. This allows the system to model the dependency between X and Y, improving user-specific alignment through multiple rounds of refinement. In practice, we optimize the CLIP score to measure and enhance the alignment between the generated images and the user input, as a proxy for maximizing mutual information. Mutual information is optimized throughout the training process to ensure the system can adapt dynamically to changing inputs and feedback.

Adaptive Feedback Loop. The feedback loop updates model parameters θ based on user feedback f, refining predictions over time:

$$\theta_{\text{new}} = \theta_{\text{old}} - \eta \nabla I(X;Y \mid f), \tag{2}$$

where η is the learning rate, and user feedback is incorporated to continuously enhance the relevance and accuracy of generated images. For details on reward-based optimization using CLIP scores, refer to Section 3.3.

3.2 Editing Operations

To facilitate co-adaptation and maximize mutual information, the system iteratively refines the prompt and adjusts attention maps M_t, alignment parameters A_t, and scaling factors c_t through gradient ascent within an RL framework, using the CLIP similarity score as a reward signal to align generated images with user intent.

Word Swap. The Word Swap operation lets users replace tokens in the prompt $P_t = \{w_1, w_2, \ldots, w_n\}$ to modify key attributes of the

generated image, such as changing $\mathcal{P}$ ="a serene blue lake" to $\mathcal{P}^*$ ="a vibrant green forest". By replacing w_i with w_i', the prompt updates to $P_{t+1} = \{w_1, \ldots, w_i', \ldots, w_n\}$, and the attention map M_t is conditionally updated to preserve the overall composition:

$$Edit(M_t, M_t^*, t) := \begin{cases} M_t^*, & \text{if } t < \tau, \\ M_t, & \text{otherwise.} \end{cases} \quad (3)$$

Here, τ controls the number of steps for injecting the updated attention map M_t^*, which is refined through gradient ascent to optimize the image based on user feedback without disrupting its structure.

$$M_t^* = M_t^* + \eta \nabla_{M_t^*} \mathcal{R}(M_t^*), \quad (4)$$

where η is the learning rate, and $\mathcal{R}(M_t^*)$ is the reward used to align the attention map with user preferences.

Adding a New Phrase. The Adding a New Phrase operation lets users introduce new elements into the generated image by inserting tokens into the prompt, such as changing $\mathcal{P}$ ="a tranquil garden" to $\mathcal{P}^*$ ="a tranquil garden with blooming flowers". The updated prompt becomes $P_{t+1} = \{w_1, \ldots, w_i, w_{\text{new}}, w_{i+1}, \ldots, w_n\}$, and coherence is maintained by adjusting attention maps with an alignment function $A(j)$ that maps indices from M_{t+1} to the original M_t:

$$(\text{Edit}\,(M_t, M_t^*, t))_{i,j} := \begin{cases} (M_t^*)_{i,j}, & \text{if } A(j) = \text{None}, \\ (M_t)_{i,A(j)}, & \text{otherwise.} \end{cases} \quad (5)$$

The adjustment ensures existing tokens retain their attention distribution while new tokens receive appropriate attention, with the alignment function A_t updated through gradient ascent to maintain consistency with user feedback.

$$A_t = A_t + \eta \nabla_{A_t} \mathcal{R}(A_t), \quad (6)$$

where η is the learning rate and $\mathcal{R}(A_t)$ is the reward function.

Attention Re-weighting. In the Attention Re-weighting method, token influence is adjusted to enhance or diminish features. For instance, scaling the attention map for a specific token (e.g., "sunflower") with a parameter $c \in [-2, 2]$ allows users to control its prominence in the image:

$$(\text{Edit}\,(M_t, M_{t+1}, t))_{i,j} := \begin{cases} c \cdot M_t(i,j), & \text{if } j = j^*, \\ M_t(i,j), & \text{otherwise.} \end{cases} \quad (7)$$

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

$$c_t = c_t + \eta \nabla_{c_t} \mathcal{R}(c_t), \quad (8)$$

Here, η is the learning rate and $\mathcal{R}(c_t)$ is the reward function, ensuring alignment with user preferences through gradient ascent; for detailed optimization processes, refer to the Appendix B.1.

3.3 TD Error Experience Replay with Joint Training

Our model uses PPO to update its policy $\pi_\theta(a_t \mid s_t)$ based on user feedback, where the state s_t includes the image and fixed prompt with rewritten feedback from the previous round, and the action a_t governs image generation or editing to optimize the cumulative reward:

$$J(\theta) = \mathbb{E}_{\pi_\theta} \left[\sum_{t=0}^{T} \gamma^t r_t \right], \quad (9)$$

With $\gamma \in [0, 1]$ as the discount factor and T as the time horizon, $V(s_i, \theta)$ estimates the expected cumulative reward, while $r_i = \text{CLIP}(I_i, P_i)$ measures the immediate reward. The value function $V^\pi(s_t)$ is updated by minimizing the temporal difference (TD) error:

$$\delta_t = r_t + \gamma V(s_{t+1}, \theta) - V(s_t, \theta). \quad (10)$$

Prioritized experience replay uses non-uniform sampling, storing transitions (s_r, a_t, r_t, s_{t+1}) and TD errors in a memory pool $\mathcal{M}$, prioritizing unfamiliar states and recent trajectories when minimizing TD errors based on a forgetting curve and historical data:

$$p_i \propto e^{-\lambda t_i} \cdot |\delta_i| + \epsilon, \quad (11)$$

The sampling probability p_i is proportional to $e^{-\lambda t_i} \cdot |\delta_i| + \epsilon$, with λ controlling the forgetting rate, t_i the experience age, $|\delta_i|$ the TD error, and ϵ ensuring non-zero probability; learning rates are adjusted by scaling each experience by $(n \cdot p_t)^{-\beta}$, where β increases from a small value to 1, and PPO updates the policy model (Three strategies of editing attention) by maximizing the objective function:

$$\mathcal{L}^{\text{PPO}}(\theta) = \mathbb{E}_t \left[\begin{array}{l} \min \Big(\rho_t(\theta) \hat{A}_t, \\ \quad \text{clip}\left(\rho_t(\theta), 1-\epsilon, 1+\epsilon\right) \hat{A}_t \Big) \end{array} \right] \quad (12)$$

The policy, consisting of three cross-attention map editing strategies, is refined by jointly maximizing rewards through gradient ascent with learning rate α_r and minimizing TD errors via gradient descent with learning rate α_δ, where $\rho_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio between new and old policies, and the advantage function $\hat{A}_t = r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t)$ reflects the relative improvement of the current action. At each iteration t in PPO Training, we compute the CLIP similarity score: $s_t = \text{CLIP}(I_t, P_t)$. If $s_t \geq \tau$, we return I_t to the user; otherwise, we continue editing until $t = N_{\text{max}}$, then return I_t.

4 Experiments

4.1 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 evaluation metrics including CLIP, LPIPS, user satisfaction, and aesthetic score, please refer to Appendix A. For additional details due to page constraints, see Appendix B.

4.2 Dataset

We developed a Q&A software that annotates prompts on our platform, automatically generating JSON files containing multi-turn dialogue information. The annotation interface is shown in Appendix C. Our training set consists of 1673 JSON files with prompts, Q&A sequences, image paths, unique IDs, and ratings for image alignment and fidelity (see Figure 4). This dataset informs our model on user expectations, analyzing subjects, emotions, styles, and settings. Feedback refines prompts, helping the model capture complex artistic intentions. We use 95% of the data for training and 5% for validation, supporting few-shot learning for better performance and user satisfaction. More details on evaluation and metrics are in Appendix C.1.

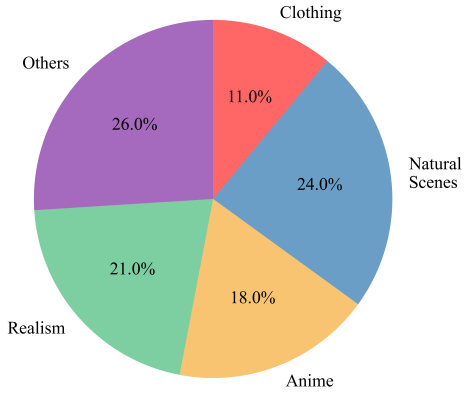

Figure 4: Dataset distribution across five categories: Clothing, Natural scenes, Anime, Realism, and Others.

4.3 Comparison Study

4.3.1 Trends Over Iterative Rounds

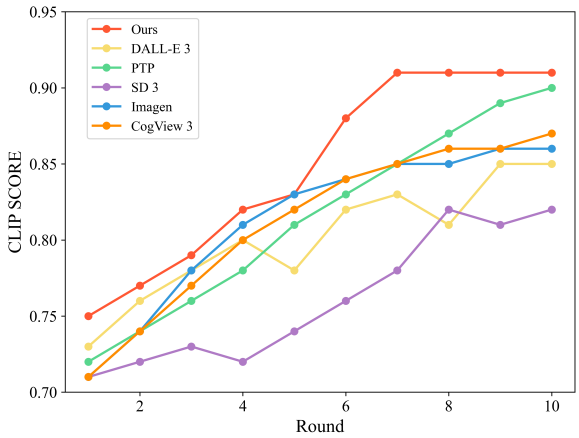

Figure 5: This graph shows CLIP score trends no more than 10 rounds for various text-to-image models.

Figure 5 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 6 recording a score of 0.42 by round 3 and stabilizing at 0.22 by round 8. Each round incrementally builds on the last, refining details without altering the prompt's core structure. For dialogues exceeding 10 rounds, we consider the process as a failure since our focus is on optimizing performance with fewer dialogue rounds. Additionally, we observed that beyond 10 rounds, the relative ranking and trends between methods remain consistent, so this scenario is not discussed.

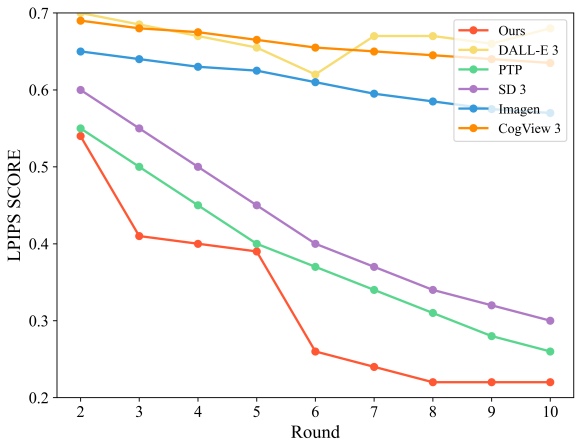

Figure 6: Illustrated in the graph are the trends of LPIPS scores for several text-to-image models no more than 10 rounds.

Table 1: Comparative analysis of prompt refinement from 100 users, with metrics scored on a 0-5 scale and response time reflecting the average duration for self-reflection and multi-dialogue processes.

Metric & Category	**Refinement Type**	
	Self-reflection	**Multi-dialogue**
Prompt Quality		
Clarity	4.2	**4.7**
Detail Level	4.1	**4.2**
Purpose Adaptability	3.3	**4.8**
Image Reception		
User Satisfaction	3.0	**4.7**
CLIP Value	0.8	**0.9**
Response Time	**3.4 s**	12.7 s

4.3.2 Prompt Refinement

Table 1 compares self-reflection and multi-round dialogue refinements. Self-reflection is faster (3.4s vs. 12s), but multi-round dialogue yields higher user satisfaction (4.7 vs. 3.0), better Purpose Adaptability (4.8 vs. 3.3), and marginal gains in Clarity (4.7 vs. 4.2) and Detail (4.2 vs. 4.1). See Appendix F for algorithm details.

4.4 Ablation Study

4.4.1 RL for Parameter Tuning

Table 2 shows the impact of Reinforcement Learning (RL) tuning on dialogue systems. RL-tuned systems reduce dialogue rounds (4.3 vs. 6.9), improve CLIP scores (0.92 vs. 0.83), and increase user satisfaction (4.73 vs. 4.14) with similar aesthetic quality (4.89 vs. 4.88). RL mainly enhances functionality and adapts better to feedback. See Appendix D for RL parameter updates.

Table 2: Comparing the effects of using PPO-based reinforcement learning (RL) versus using only three editing strategies (without RL) on user satisfaction and image quality metrics. Data is averaged from 100 diverse users, with final CLIP and Aesthetic Scores based on 0-5 subjective ratings.

Metrics	With PPO (RL)	Without PPO (Only 3 Strategies)
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: Assessing cross attention (CA) impact by averaging CLIP and Aesthetic Scores from final interactions, based on 0-5 user ratings of image attractiveness, using data from a diverse sample of 100 users.

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

4.4.2 Comparing Edited Cross Attention with Normal Cross Attention.

Table 3 shows the superior performance of edited cross attention (CA) over normal CA. Edited CA adjusts dynamically based on context and feedback, reducing dialogue rounds (3.7 vs. 6.1) and improving CLIP scores (0.88 vs. 0.81), user satisfaction (4.82 vs. 3.94), and image aesthetics (4.71 vs. 4.48). For details on how edited CA integrates with reinforcement learning, see Appendix E.

4.4.3 Impact of Strategy Removal

Table 4 shows the effects of removing key strategies on model performance. Removing strategies like Word Swap and Attention Re-weighting increases dialogue rounds (up to 6.2) and lowers user satisfaction (as low as 4.10/5). Our model, with all strategies intact, achieves the best results with 4.3 dialogue rounds, a 0.92 CLIP score, and a satisfaction score of 4.73/5.

4.5 User Study

4.5.1 User Satisfaction Evaluation

Figure 7 summarizes user evaluations, showing high satisfaction with prompt refinement, image coherence, and quick intent capture. Despite complex computations, response times were not seen as slow. Figure 8 highlights the model's fast adaptation, with 59 users satisfied by Round 3 and 86 by Round 5, demonstrating its efficiency in aligning

Table 4: Impact of Removing Each Strategy on Model Performance

Model Variant	Dialogue Rounds	CLIP Score	User Satisfaction	Aesthetic Score
w/o Word Swap	5.8	0.85	4.20/5	4.70/5
w/o Adding Phrase	6.2	0.80	4.10/5	4.65/5
w/o Attention Re-weighting	5.5	0.88	4.50/5	4.80/5
Ours	**4.3**	**0.92**	**4.73/5**	**4.89/5**

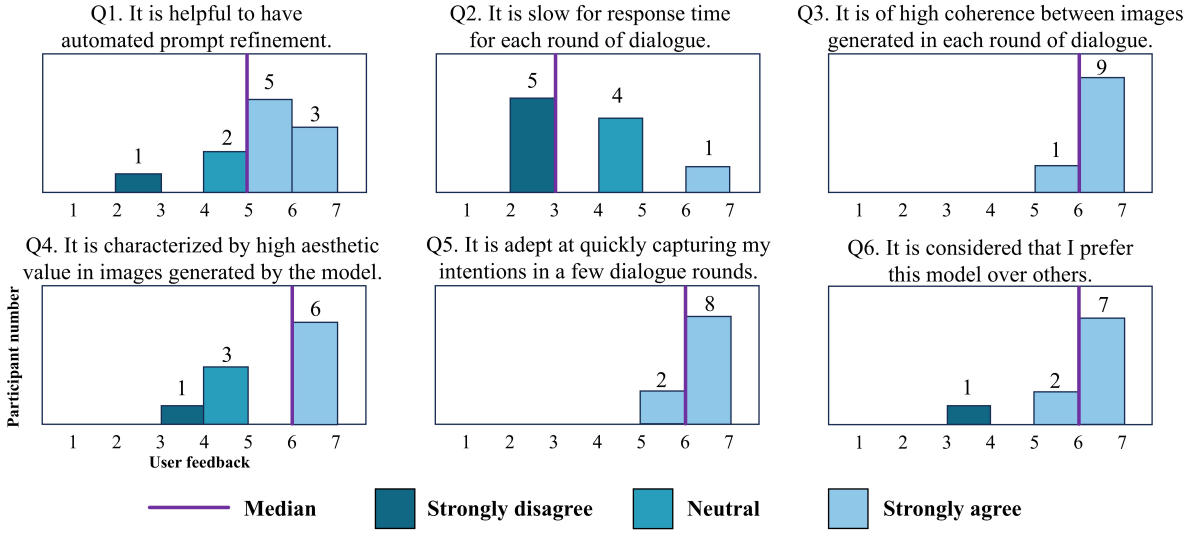

Figure 7: The chart shows user feedback on a model, highlighting mixed responses with positive feedback on image coherence and capturing intentions, but concerns over response time.

with preferences.

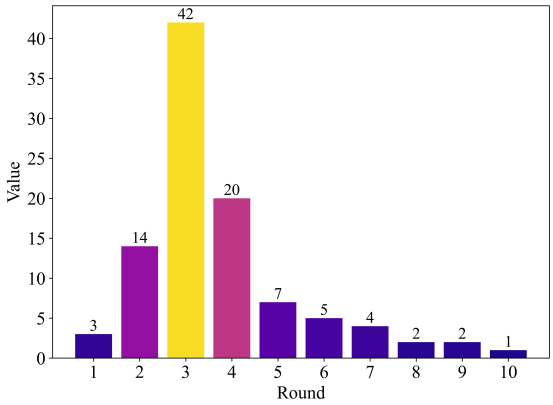

Figure 8: The chart shows the rapid decline in user interaction rounds needed for satisfaction.

4.5.2 Task-Based Model Performance Evaluation

Table 5 compares the SSIM performance of six models across three tasks. 81 random users participated in this testing, responsible for generating images as closely aligned with the target images as possible using prompts. Task 1 involves simple target images, Task 2 includes more complex images, and Task 3 requires generating intricate, highly detailed images. The standard deviation (SD) reflects consistency, with lower SD in Task 1 showing more uniform results and higher SD in Task 3 indicating greater variation due to the increased complexity of the target images.

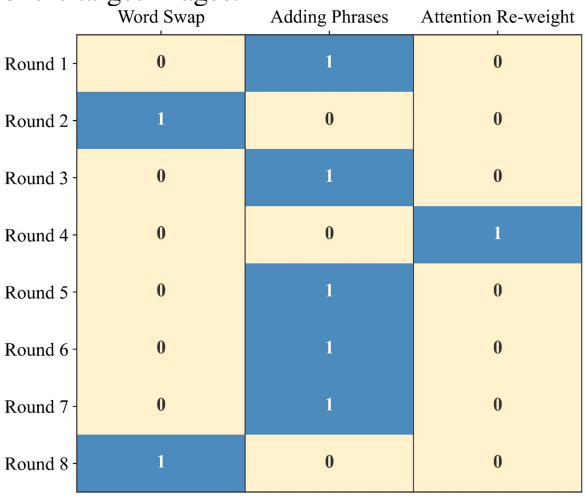

Figure 9: The figure shows the various operations applied during different rounds.

4.6 Visualization Results

4.6.1 Strategy Usage Across Rounds

Figure 9 presents a case study illustrating the prompt refinement choices made across dialogue

Table 5: Model performance in three tasks measured by SSIM. Higher SSIM indicates better performance.

Model	Task 1		Task 2		Task 3		Overall	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
CogView 3 (Zheng et al., 2024)	0.382	0.017	0.330	0.079	0.207	0.093	0.306	0.102
SD 3 (Esser et al., 2024)	0.446	0.043	0.437	0.091	0.334	0.094	0.406	0.077
Imagen (Saharia et al., 2022)	0.415	0.008	0.388	0.017	0.296	0.044	0.366	0.053
P2P (Hertz et al., 2022b)	0.510	0.007	0.289	0.005	0.107	0.011	0.302	0.165
DALL·E 3 (Betker et al., 2023)	0.537	0.007	0.297	0.025	0.367	0.014	0.400	0.102
Ours	0.518	0.018	0.468	0.007	0.396	0.030	0.461	0.054

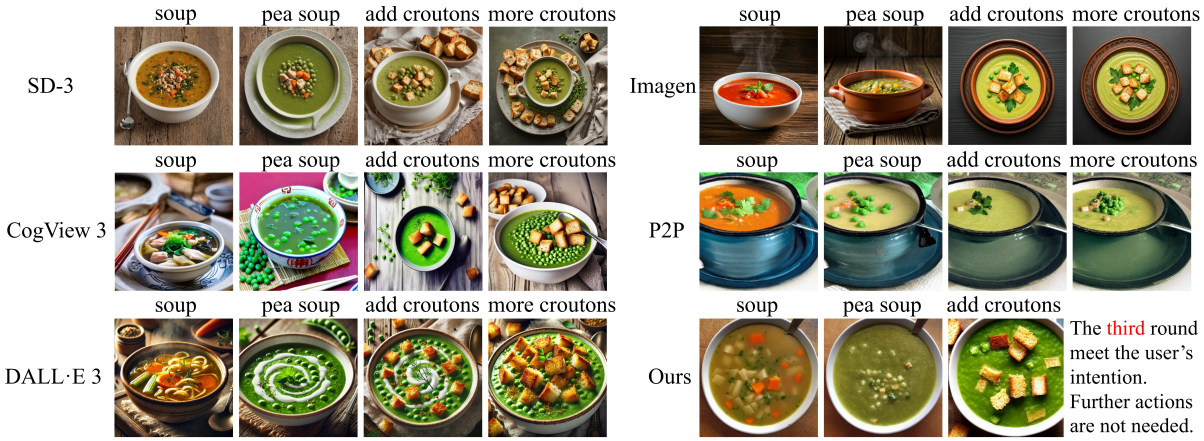

Figure 10: The comparison showcases our model's ability to efficiently align with user intentions, such as refining soup images by adding specific elements like croutons. By the third round, our model fully captures the user's intention, demonstrating minimal need for further actions.

rounds from one user. Word Swap occurs selectively in Rounds 2 and 8, while Attention Reweighting appears in Rounds 4 and 6. The distribution showcases the adaptive nature of the prompt refinement process, where different operations are employed based on the evolving dialogue context. Beyond this case, overall usage rates from users indicate that Adding Phrases is the most frequently used strategy at 64.47%, followed by Attention Reweighting at 17.84%, and Word Swap at 17.69%.

4.6.2 Multi-Round Dialogue Comparison

Figure 10 compares image generation models in adapting to prompts like "soup" and "add croutons." Our model quickly aligns with user intent by round three, while others require more rounds for further refinement.

5 Conclusion and Future Work

In this study, we introduced a new image generation method using a human-in-the-loop approach to enhance interaction and responsiveness to ambiguous prompts. Our model effectively aligns with user expectations through adaptive prompt engineering and mutual information optimization. Future work includes releasing our training dataset for transparency, refining the model's interpretive abilities, expanding to other domains, and conducting comprehensive benchmarks to assess alignment between user intent and generated images. These efforts aim to make personalized image generation more accessible without requiring technical expertise.

6 Limitations

Our proposed strategy for human-machine co-adaptation demonstrates notable improvements in handling ambiguous prompts and refining image generation through user feedback. However, it faces limitations, including its reliance on continuous human feedback, potential difficulty in generalizing to complex domains, challenges in resolving multiple ambiguities simultaneously, increased computational time, and a relatively small dataset for fine-tuning. These factors highlight areas for future improvements, such as enhancing computa-

tional efficiency, expanding to more complex tasks, and reducing dependence on ongoing user input.

References

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. 2023. Improving image generation with better captions. *DALL-E 3*. OpenAI.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. 2024. Scaling rectified flow transformers for high-resolution image synthesis. *Preprint*, arXiv:2403.03206.

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. 2022a. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*.

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

Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. 2023. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*.

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*.

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*.

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*.

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.

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.

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 Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Preprint*, arXiv:2205.11487.

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*.

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

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.

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.

Wendi Zheng, Jiayan Teng, Zhuoyi Yang, Weihan Wang, Jidong Chen, Xiaotao Gu, Yuxiao Dong, Ming Ding, and Jie Tang. 2024. Cogview3: Finer and faster text-to-image generation via relay diffusion. *Preprint*, arXiv:2403.05121.

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 Evaluation Metrics

The experimental framework of this study is meticulously designed to evaluate our text-to-image generation model across four 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 G.

User Satisfaction: This metric is measured through post-interaction surveys where participants rate their satisfaction with the generated images on a Likert scale (e.g., 1-5). It reflects the extent to which the generated images meet the users' expectations and preferences. Higher scores indicate greater user satisfaction with the alignment of the generated images to their intended outcomes.

Aesthetic Score: Aesthetic score assesses the visual appeal of the generated images. Participants evaluate the images based on criteria such as composition, color harmony, and overall attractiveness, rating them on a predefined scale (e.g., 1-5). For this evaluation, we used an electronic feedback form, where users were asked to fill out the form after interacting with the system and generating images. The evaluation form included fields for rating aspects such as color harmony, balance, and visual appeal. Higher scores signify that the images are perceived as more aesthetically pleasing and artistically valuable. The annotators for this study consisted of a diverse group of users, ensuring a broad range of perspectives on image quality.

B 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.

B.1 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.

B.2 Optimization Details

To optimize image generation, the model dynamically selects from three strategies (adding phrases, word swapping, re-weighting) using the CLIP score as the reward function to update the parameters of the chosen strategy. This feedback-driven approach adjusts one strategy per iteration, producing parameter sets that adapt image generation to human preferences. The strategies correspond to three controllers: Attention-Replace, Attention-Refine, and Attention-Reweight, which adjust cross-attention during generation. Each controller uses cross-attention between images and prompts, with trainable parameters for self-attention, attention injection steps, and adaptive updates to cross-attention maps. The optimization for parameter updates is mathematically represented as:

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:

$$\begin{aligned}\mathcal{R}(\theta) = \text{CLIPScore}(I_{\text{gen}}, P_{\text{prev}}) \\ +\lambda \cdot \text{CLIPScore}(I_{\text{gen}}, P_{\text{new}})\end{aligned} \tag{13}$$

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:

$$I_{\text{gen}}^{(k+1)} = \text{Update}(I_{\text{gen}}^{(k)}, \theta^{(k)}) \tag{14}$$

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.

$$M_{\text{new}} = (M + \eta \cdot \Delta M) \cdot M_{\text{cross_attention}} \tag{15}$$

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

$$\theta'_m = \theta_m + \eta \nabla_{\theta_m} \mathbb{E}[R] \tag{16}$$

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

$$\theta'_\alpha = \theta_\alpha + \eta \nabla_{\theta_\alpha} \mathbb{E}[R] \tag{17}$$

to maximize the expected reward ($\mathbb{E}[R]$) using gradient ascent with learning rate η.

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:

$$\begin{aligned} \text{attn_base_replace}_{ijk} &= \text{attn_base}_{ijk} \cdot \text{mapper}_{kj} \\ &\implies (\text{attn_base_replace})_{\text{permute}(2,0,1,3)} \end{aligned} \tag{18}$$

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 then calculated as:

$$M^{(t)}_{\text{update}} = \beta_t \cdot M^{(t)}_{\text{orig}} + (1 - \beta_t) \cdot M^{(t)}_{\text{new}} \tag{19}$$

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:

$$\begin{aligned} M^{(t)}_{\text{refine}} &= \beta_t \cdot M^{(t)}_{\text{orig}} + (1 - \beta_t) \cdot M^{(t)}_{\text{new}}, \\ \beta_t &= \beta_{t-1} + \gamma \cdot \nabla_{\beta_t} \mathcal{R}(\theta) \end{aligned} \tag{20}$$

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:

$$\begin{aligned} M^{(t)}_{\text{reweight}} &= \sum_i \gamma_{t,i} \cdot M^{(t,i)}_{\text{refine}}, \\ \gamma_{t,i} &= \gamma_{t-1,i} + \kappa \cdot \nabla_{\gamma_{t,i}} \mathcal{R}(\theta) \end{aligned} \tag{21}$$

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

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

$$\alpha_{t+1} = \alpha_t + \eta \nabla_{\alpha_t} \mathcal{R}(\theta) \tag{22}$$

$$\zeta_{t+1} = \zeta_t + \gamma \nabla_{\zeta_t} \mathcal{R}(\theta) \tag{23}$$

$$\delta_{t+1} = \delta_t + \kappa \nabla_{\delta_t} \mathcal{R}(\theta) \tag{24}$$

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.

C Q&A Software Annotation

C.1 Annotator Details

We recruited 30 annotators with diverse backgrounds for human evaluations. The group included 10 individuals aged 18-30, 10 individuals aged 31-50, and 10 individuals aged 51-70. In terms of gender, there were 15 male and 15 female participants. Regarding education, 40% of the annotators had a high school diploma or equivalent, 30% held a bachelor's degree, and 30% had postgraduate degrees.

This diversity ensures a comprehensive evaluation of the system's ability to generate images that resonate with different demographics. Each annotator independently labeled and rated the images based on their relevance to the input prompts, visual coherence, and artistic quality. These human evaluations were averaged to ensure consistency and reliability in the final assessment.

C.2 Q&A Software Annotation Interface

D Ablation of RL tuning

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

$$\theta^{\text{RL}} = \theta_0 + \sum_{t=1}^{T} \eta \nabla_\theta \mathcal{R}(\theta_t), \quad \theta^{\text{Fixed}} = \theta_0 \tag{25}$$

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.

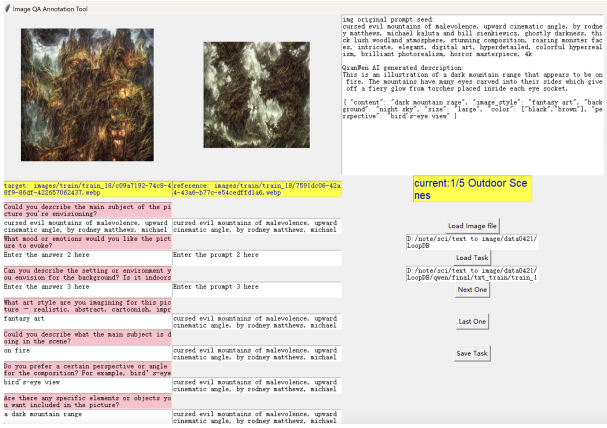

Figure 11: Screenshot of the Q&A software annotation interface.

E Ablation of cross attention control

$$\theta^{(t+1)}_{\text{Weighted}} = \theta^{(t)}_{\text{Weighted}} + \eta \nabla_\theta \mathcal{L}(I_t, \text{Feedback}_t, M) \tag{26}$$

$$\theta^{(t+1)}_{\text{Empty}} = \theta^{(t)}_{\text{Empty}} + \eta \nabla_\theta \mathcal{L}(I_t, \text{Feedback}_t, M_{\text{new}}) \tag{27}$$

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.

F LLM Prompt Refinement

Algorithm 1 Multi-dialogue Prompt Refinement Process for ChatGPT-4

Input: Initial prompt p_0
Output: Refined prompt p_i that meets conditions and is free of ambiguities
Define $C(p)$: Checks if the prompt p meets predefined conditions
Define $A(p)$: Checks if the prompt p is free of ambiguities
Initialize iteration index $i \leftarrow 0$
while $C(p_i) = \texttt{False}$ **or** $A(p_i) = \texttt{False}$ **do**
  if $A(p_i) = \texttt{False}$ **then**
    $p_{i+1} \leftarrow \texttt{ResolveAmbiguities}(p_i)$
  else if $C(p_i) = \texttt{False}$ **then**
    $p_{i+1} \leftarrow \texttt{ToMeetConditions}(p_i)$
  end if
  Increment i: $i \leftarrow i + 1$
end while
return Refined prompt p_i

The Multi-dialogue Refine process in ChatGPT-4 iteratively adjusts prompts to meet predefined, ambiguity-free conditions. Initially, the model checks if the prompt p_0 meets criteria, and if ambiguities exist, the process loops to refine them. The refinement is modeled as:

$$y_{t+1} = M(p_{\text{refine}} \parallel x \parallel y_0 \parallel \text{fb}_0 \parallel \ldots \parallel y_t \parallel \text{fb}_t), \tag{28}$$

where y_t is the output at iteration t, M is the model, and fb_t is feedback at t. Through multi-turn dialogue, the model refines prompts by asking clarifying questions to ensure they are comprehensive. Necessary adjustments are made based on feedback:

$$y_{t+1} = M(x \parallel y_t \parallel \text{fb}_t). \tag{29}$$

G The Processing of Setup Prompts from Human Feedback Using Large Language Models (LLM)

Table 6: ChatGPT-4 Prompt Rewriting and Type Judgment Process

Process Description
Given Prompt:
`{current_prompt}`
User Request:
`{user_input}`
Instructions:
Generate a new prompt by incorporating these changes while keeping the original structure intact. Consider using the following methods: *word swapping*, *adding phrases*, or *attention reweighting*.
Ensure the modifications align closely with the user's request and specify the method used.
Output Format:
`{new_prompt, type}`

H More Visualization Results

Figure 12: The figure illustrates the iterative process of refining a prompt through six rounds. In each round, modifications are made using different strategies: adding phrases, attention re-weighting, and word swap. Each step's impact on the description of the image is shown visually, along with the type of modification used.

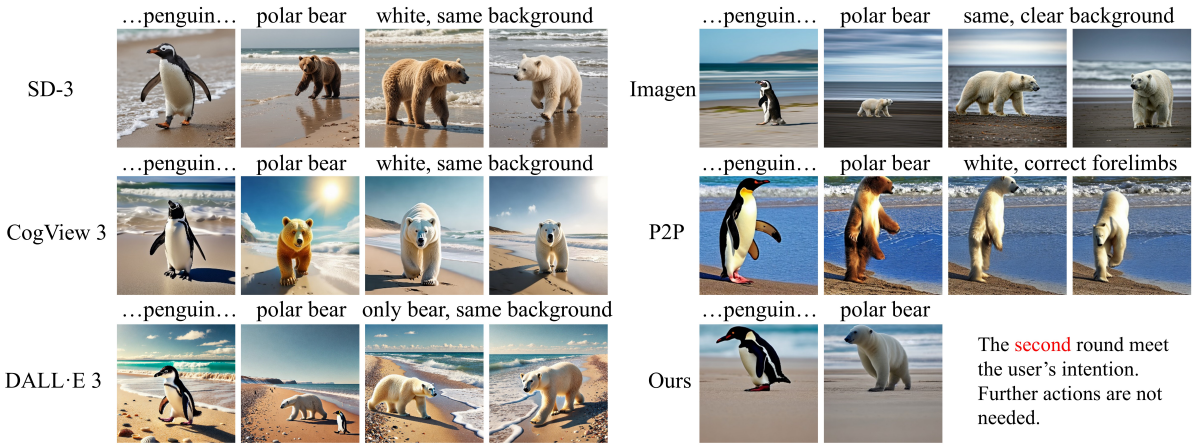

Figure 13: The comparison demonstrates our model's few-shot learning capability, effectively adapting to user preferences with minimal dialogue.