GenPilot: A Multi-Agent System for Test-Time Prompt Optimization in Image Generation

Anonymous ACL submission

Abstract

Text-to-image synthesis has made remarkable progress, yet accurately interpreting complex and lengthy prompts remains challenging, often resulting in semantic inconsistencies and missing details. Existing solutions, such as fine-tuning, are model-specific and require training, while prior automatic prompt optimization (APO) approaches typically lack systematic error analysis and refinement strategies, resulting in limited reliability and effectiveness. Meanwhile, test-time scaling methods operate on fixed prompts and on noise or sample numbers, limiting their interpretability and adaptability. To solve these, we introduce a flexible and efficient test-time prompt optimization strategy that operates directly on the input text. We propose a plug-and-play multi-agent system called GenPilot, integrating error analysis, clustering-based adaptive exploration, fine-grained verification, and a memory module for iterative optimization. Our approach is model-agnostic, interpretable, and well-suited for handling long and complex prompts. Simultaneously, we summarize the common patterns of errors and the refinement strategy, offering more experience and encouraging further exploration. Experiments on DPG-bench and Geneval with improvements of up to 16.9% and 5.7% demonstrate the strong capability of our methods in enhancing the text and image consistency and structural coherence of generated images, revealing the effectiveness of our test-time prompt optimization strategy. The code is available at https://github.com/GenPilot-Any/GenPilot.

1 Introduction

Recently, text-to-image generation models (Ho et al., 2020; Rombach et al., 2022; Ramesh et al., 2022) have witnessed remarkable developments, indicating their excellent performance across a multitude of applications. Nevertheless, translating complex and compositional prompts into semantically aligned, high-fidelity images remains a significant challenge. As prompt complexity increases, existing models struggle to preserve semantic coherence, exposing a persistent semantic gap and resulting in compositionality catastrophe. These limitations are further exacerbated by architectural inconsistencies across models, which hinder the development of a unified and generalizable framework adaptable to diverse T2I paradigms.

To improve multimodal alignment in T2I generation, existing efforts (Mañas et al., 2024; Fu et al., 2024a; Saharia et al., 2022) can be broadly categorized into fine-tuning and prompting. While fine-tuning or retraining model parameters to capture detailed semantics information, it is often computationally intensive and model-specific. In contrast, manual prompting relies heavily on human intuition, lacking scalability across prompts, tasks, and architectures. Recent works, such as OPT2I (Mañas et al., 2024), DPO-Diff (Wang et al., 2024b), and AP-Adapter (Fu et al., 2024a), explore automatic prompt optimization to enhance generation quality. However, most approaches require additional training and are designed for certain models, also often lack systematic error analysis. With the advancement of large language models, test-time scaling has been explored in various scenarios by leveraging additional computational resources and inference-time adjustments to improve the performance. Some studies extend this idea to image generation, for instance, SANA-1.5 (Xie et al., 2025) generates many samples and a verifier selects the best sample.

Although recent progress in automatic prompt optimization (APO) and test-time scaling (TTS) has improved image generation, they still suffer from limitations such as reliance on random exploration or fixed prompts, lack of systematic error identification, or coarse-grained verification, hindering flexibility and interoperability. To address these issues, we propose GenPilot, a plug-

Figure 1: Visualized examples from DALL-E 3 (Betker et al., 2023) with GenPilot processing complicated and lengthy prompts. Compared to the prompt engineering method (PE), generative models with GenPilot successfully achieve accurate results, addressing both the semantic gap, compositionality catastrophe, and even the challenging tasks of exclusion of certain objects.

and-play multi-agent system that brings test-time scaling into the prompt space by formulating the prompt optimization as a search problem, enabling dynamic and interpretable prompt refinement. GenPilot is broadly applicable across diverse models without any model training to improve the prompts for image generation. Some examples are presented in Figure 1.

Our system contains two main stages: the error analysis module and the test-time prompt optimization module. In Error Analysis, GenPilot decomposes the initial prompt, leverages visual question answering (VQA) and captioning to detect and localize semantic inconsistencies. During test-time optimization, GenPilot iteratively refines the prompt based on errors and memory feedback with a multi-modal large language model (MLLM) (Bai et al., 2025) scorer, cluster, and memory.

The main contributions are three-fold:

- We propose **GenPilot**, a plug-and-play multi-agent system that performs test-time prompt optimization as a search problem for interpretable results, improving image consistency without training across diverse T2I models.
- GenPilot introduces systematic error analysis and fine-grained verification, enabling dynamic prompt exploration via clustering and iterative feedback, and memory updates.
- Experiments on both long prompts from DPG-bench (Hu et al., 2024) and short prompts from Geneval (Ghosh et al., 2023) show that GenPilot consistently improves performance across models, demonstrating robustness and generalizability for T2I tasks.

2 Related Work

2.1 Text to Image Generation

Recently, text-to-image models (T2I models) have developed rapidly. Nonetheless, their performance is restricted not only by architectural design but also by the quality of the input prompts. Early methods such as Stable Diffusion models (SD) (Rombach et al., 2022) rely on CLIP-based (Radford et al., 2021) encoder and latent diffusion models. DALL-E 2 (Ramesh et al., 2022) employs unCLIP while DALL-E 3 (Betker et al., 2023) and PixArt-α (Chen et al., 2023) introduce T5 (Raffel et al., 2023) to enhance alignment. More recently, FLUX.1 dev [1] introduces RoPE (Su et al., 2023) to enhance spatial coherence, while FLUX.1 schnell [2] increases inference speed within 1 to 4 steps.

[1] https://huggingface.co/black-forest-labs/FLUX.1-dev

[2] https://huggingface.co/black-forest-labs/FLUX.1-schnell

2.2 Automatic Prompt Optimization for Image Generation

T2I models are still facing challenges in text-to-image consistency (Wu et al., 2023), therefore, Automatic Prompt Optimization (APO) (Pryzant et al., 2023), an automatic technique to optimize the performance of models without training (Ramnath et al., 2025), has been explored. Existing APO studies include backpropagation-free optimization method (Mañas et al., 2024), Proximal Policy Optimization (PPO)-based (Schulman et al., 2017) reinforcement method (Hao et al., 2023; Cao et al., 2023), adapters (Fu et al., 2024b), and some products such as MagicPrompt[3] and PromptPerfect[4]. However, most existing methods lack error analysis, are limited to specific models (e.g., Stable Diffusion (Rombach et al., 2022)), and often rely on coarse-grained evaluators such as CLIPScore (Hessel et al., 2022) or FID (Heusel et al., 2017), which provide limited reliability in assessing image-text alignment (Mañas et al., 2024).

2.3 Test-Time Scaling for Image Generation

In recent years, test-time scaling has been extensively studied in large language models (Zhao et al., 2025) with multiple inference samples and a selection mechanism to find the suitable result (Lightman et al., 2023). The study (Ma et al., 2025) formulates the task as a search problem in noise space and selects the best in N samples. SANA-1.5 (Xie et al., 2025) repeats the number of samples rather than denoising steps to scale up the performance. Also, FK STEERING (Singhal et al., 2025) employs FK-IPS (Moral, 2004) to guide the sample path with the high reward. However, different from those methods operating in the noise space with a fixed input, we formulate the scaling into the input space, which we call "test-time prompt optimization" to generate N samples and cluster them to find the optimal one.

3 Method

3.1 How to Scale at Inference Time for Prompt

For test-time scaling of textual prompts, we formulate it as a search problem aimed at finding the optimal input for diverse image generation models, which is unknown. Unlike the prior work, such as (Ma et al., 2025), which scales the sample noise, our method focuses on the exploration and refinement of the textual inputs. We operate within a predefined discrete text space, and the prompt is scaled through an iterative process. GenPilot generates multiple candidate prompts and scores them, then the candidates are clustered to help identify an optimal one. This optimal candidate then serves as the basis for the subsequent round of optimization. Consequently, performance is expected to scale positively with the progression of this prompt optimization process.

3.2 Overall Framework

As illustrated in Figure 2, GenPilot operates in two coarse-grained stages: **Error Analysis** and **Test-Time Prompt Optimization**.

Beginning at an initial prompt and image, GenPilot decomposes the prompt into "meta-sentences" with an AI agent (Wang et al., 2024a). Based on these units, GenPilot performs parallel error detection via VQA and captioning, named the error integration strategy. The VQA-based branch queries object-level details, while the caption-based branch compares captions with the original prompt. An error-integration agent aggregates the inconsistencies into a comprehensive error list, with another agent mapping each error back to specific prompt segments. In the test-time prompt optimization stage, a refinement agent generates candidate prompts based on the metadata, including the original prompt and image, and error analysis and mapping. Detailed definitions and metadata formats are provided in Appendix A. These candidates are evaluated by an MLLM scorer through VQA and a rating strategy. GenPilot clusters the prompts and selects the optimal cluster for sampling and image generation. The memory module is iteratively updated with visual and textual feedback until convergence or a maximum iteration threshold is reached. The system prompt can be found in Appendix G.

3.3 Error Analysis and Mapping

3.3.1 Prompt Decomposition

Prior work (Wang et al., 2024c) decomposes prompts into object and background details, but often ignores inter-object relationships, causing semantic errors. In contrast, we design a coarser-grained prompt decomposition into pieces with an agent that contains objects, relationships, and background information. For example, given a prompt

[3] https://huggingface.co/Gustavosta/MagicPrompt-Stable-Diffusion

[4] https://promptperfect.jina.ai/

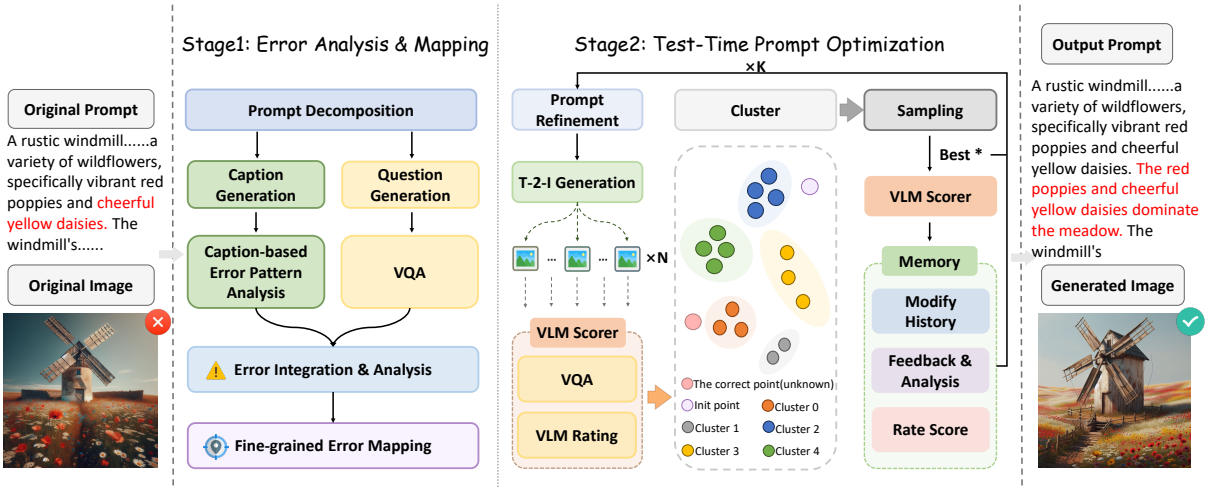


Figure 2: Overview of our proposed multi-agent system for test-time prompt optimization. GenPilot utilizes a multimodal large language model as the agent. In stage 1, we first decompose the prompt, then we introduce the error integration strategy based on image caption and VQA results, and map the error to the original prompt. In stage 2, we introduce the test-time scaling by formulating the problem as a search problem and operating on the input text space. The test-time prompt optimization is iteratively processed with a refinement agent, an MLLM scorer, a clustering algorithm (MacQueen, 1967), and the memory module to sample the optimal currently.

P, the agent segments it as:

$$P = \{s_1, s_2, \ldots, s_n\} \tag{1}$$

where s_n denotes sentence pieces. A more detailed and fine-grained mapping is subsequently performed during the error mapping stage.

3.3.2 Error Analysis and Mapping

Evaluating text-image alignment by VQA with MLLMs is constrained in complex scenes, leading to unreliable scores. Therefore, we design an integrated error analysis strategy that combines VQA-based and caption-based detection.

Question Generation. Inspired by DSG (Cho et al., 2024), we introduce an MLLM agent to generate full coverage questions. Given a decomposed prompt, the question-generator agent identifies the objects and formulates yes/no questions about object existence, attributes, states, spatial relations, and background information for precise analysis.

VQA Analysis. Each generated question is passed to another MLLM that serves as the VQA agent, who provides a label from $\{\texttt{YES}, \texttt{NO}\}$ and brief explanations to the errors, in the form:

$$e_i = (\text{type}_i, \text{explanation}_i) \tag{2}$$

where type_i is the type of inconsistency and explanation_i refers to the detailed errors. The full error set $\mathcal{E}_{vqa}$ is represented as:

$$\mathcal{E}_{vqa} = \{e_1, e_2, \ldots, e_n\} \tag{3}$$

Caption-Based Error Analysis. For caption-based error analysis, an MLLM generates a detailed caption C_i for image I_i, then a comparison agent contrasts C_i with the original prompt P_i to detect semantic discrepancies. The full error set from caption $\mathcal{E}_c$ is represented as:

$$\mathcal{E}_c = \{e_1, e_2, \ldots, e_n\} \tag{4}$$

where e_i denotes the error analyzed from the comparison agent.

Integrated Error Identification. In this stage, an MLLM agent functions as an error-integration agent, tasked with synthesizing information from multiple analytical sources, formulated as:

$$\mathcal{E}_u = A_{error}(I, P, \mathcal{E}_c, \mathcal{E}_{vqa}) \tag{5}$$

where $\mathcal{E}_u$ is the finalized error set, I is the original image, and P is the original prompt. $\mathcal{E}_c$ and $\mathcal{E}_{vqa}$ are the error sets from caption- and VQA-based detection. A_{error} is the error-integration agent.

Error Mapping. Error localization maps an identified error to the pieces of the original prompt that lead to it, bridging the abstract error and concrete prompt to support the refinement module.

3.4 Test-time Prompt Optimization

Prompt Refinement. We first introduce a prompt refinement agent based on metadata to modify the error mapping sentence $m_i \in M$

and generate N diverse candidate modifications $\{m_i^1, m_i^2, \ldots, m_i^N\}$, using multiple references to enhance diversity. Next, each sentence m_i^j is merged into the original prompt P by a branch-merge agent, the process can be formulated as:

$$P_i^j = A_{merge}(P, m_i^j), \quad j = 1, 2, \ldots, N \quad (6)$$

where P_i^j denotes the candidate prompts generated by the branch-merge agent A_{merge}, which are then passed to T2I model to generate images.

MLLM scorer. Subsequently, GenPilot employs an MLLM scorer that acts as a test-time verifier to indirectly evaluate prompt quality via the generated images. Inspired by T2I-CompBench (Huang et al., 2025), we design our evaluation rules from the following three aspects: attribute binding including *color*, *number*, *shape*, *state*, and *texture* of the object, relationship and position, and background information and style including the *background description*, the *style*, and *atmosphere*. A more detailed explanation is provided in Appendix B.

For each candidate prompt and image pair, the VQA agent analyzes potential inconsistencies based on the question list generated, and a rating agent provides more reliable scores. The whole scoring process is defined as:

$$\mathcal{S}(P_i^j) = avg(A_{rate}(I_i^j, P, A_{vqa}(I_i^j, P))) \quad (7)$$

where A_{rate} is the rating agent, A_{vqa} denotes the VQA agent, P_i^j is the candidate prompts and I_i^j refers to the corresponding images, and P is the original prompt.

Clustering. The scored candidate prompts are then processed with K-Means clustering (MacQueen, 1967), including Bayesian updates to progressively identify high-potential prompt candidates. Initially, each cluster j is assigned the prior probability $P_j = 1/K$, and the candidates are clustered into K groups using K-Means. Then posterior probabilities P_j^{post} are computed using the Bayesian update rule, formulated as:

$$P_j^{\text{post}} = \frac{L_j P_j}{\sum_k L_k P_k} \quad (8)$$

where L_j refers to the likelihood. The cluster j^* with the highest posterior probability is identified as the best cluster in this round, shown as:

$$j^* = \arg\max_j P_j^{\text{post}} \quad (9)$$

Following that, a sampled prompt set s^*, which contains m candidate sampled prompts from the cluster j^*. The P_j^{post} serves as the prior distribution for the next round.

Memory. For each prompt in the m sample set, we employ the T2I models to generate the image and evaluate them by MLLM scorer. The average rating and detailed error analysis are stored in the memory module, serving as a historical reference for the subsequent optimization iterations, which can be formulated as:

$$\mathcal{M} \leftarrow \mathcal{M} \cup \{(s^*, I_{s^*}, \mathcal{S}(s^*), A_{sum}(\mathcal{E}_{s^*})\} \quad (10)$$

where s^* denotes the sampled prompt set, I_{s^*} refers to the corresponding images and A_{sum} refers to an agent who summarizes the error analyses for s^*.

4 Experiment

4.1 Implementation Details

In the experiment, we employ Qwen2-VL-72B-Instruct (Bai et al., 2025) as the MLLM agent. Our method operates with 20 candidate prompts, 5 cluster labels, and undergoes 10 modification cycles. Given that users often tend to optimize an image only when initial outputs are unsatisfactory, we construct a challenging subset of 264 prompts selected from the DPG-bench (Hu et al., 2024) dataset, with most prompts falling below a threshold of 0.81, posing significant challenges even for the state-of-the-art models. Though our method is principally designed for complex and lengthy prompts, we also extended our evaluation to short prompts on the GenEval benchmark (Ghosh et al., 2023) to ensure a comprehensive assessment of its capabilities. The results are conducted three times to calculate the average score. All the system prompts are shown in Appendix G.

4.2 Comparison on DPG-bench subset

4.2.1 Quantitative Evaluation

We evaluate GenPilot on a wide range of T2I models and compare it with the Prompt Engineering method (PE) and SD-based methods: MagicPrompt and BeautifulPrompt (Cao et al., 2023). According to Table 1, GenPilot successfully improves the performance in the overall "Average" score on all models tested. For example, the average score improves from 72.04 to 74.08 on DALL-E 3, from 68.16 to 73.32 on FLUX.1 (68.38 by PE), and from 53.16

Model	Average	Global	Entity	Attribute	Relation	Other
DALL-E 3	72.04	**89.47**	82.54	79.97	**90.41**	63.41
DALL-E 3 + PE	72.29	85.37	82.89	**82.98**	88.88	**66.45**
DALL-E 3 + **Ours**	**74.08**	**89.47**	**83.73**	81.96	88.70	60.98
FLUX.1 schnell	68.16	79.12	80.33	81.02	88.24	**65.75**
FLUX.1 schnell + PE	68.38	**81.32**	79.69	77.54	85.99	61.64
FLUX.1 schnell + **Ours**	**73.32**	79.12	**82.42**	**83.20**	**89.86**	61.64
Stable Diffusion v1.4	53.16	85.71	65.23	65.70	78.63	47.37
Stable Diffusion v1.4 + MagicPrompt	53.61	92.85	66.57	64.42	77.86	47.37
Stable Diffusion v1.4 + BeautifulPrompt	55.99	85.71	66.04	66.67	81.68	52.63
Stable Diffusion v1.4 + PE	56.08	85.71	69.27	**70.83**	**88.55**	47.37
Stable Diffusion v1.4 + **Ours**	**62.12**	**100.00**	**71.43**	67.94	79.39	**57.89**
Stable Diffusion v2.1	57.24	93.75	71.92	70.04	82.83	46.15
Stable Diffusion v2.1 + MagicPrompt	58.93	93.75	70.88	**71.81**	78.79	42.31
Stable Diffusion v2.1 + BeautifulPrompt	58.04	90.63	71.58	68.94	82.32	46.15
Stable Diffusion v2.1 + PE	56.49	**96.88**	71.23	69.60	**85.35**	30.77
Stable Diffusion v2.1 + **Ours**	**61.72**	**96.88**	**76.26**	71.16	77.78	**53.85**
Stable Diffusion 3	58.81	79.63	71.15	73.02	**84.01**	51.42
Stable Diffusion 3 + MagicPrompt	59.26	83.33	**72.82**	73.02	81.41	42.86
Stable Diffusion 3 + BeautifulPrompt	60.49	87.04	71.54	**73.51**	80.67	48.58
Stable Diffusion 3 + PE	58.81	81.48	70.26	70.60	82.16	31.43
Stable Diffusion 3 + **Ours**	**62.89**	**88.89**	72.31	68.98	79.55	**51.43**
Sana-1.0 1.6B	73.98	85.71	83.44	81.83	91.63	67.12

Table 1: Quantitative evaluation of T2I generation performance on DPG-bench challenging dataset comparing GenPilot with generative models and other enhancement methods. Our approach consistently achieves superior Average performance and demonstrates notable improvements across various models.

to 62.12 on SDv1.4 (55.99 and 53.61 by BeautifulPrompt and MagicPrompt, respectively). Similar gains are observed on SDv2.1 and SD3, indicating the robustness and generalizability of GenPilot, revealing its ability to enhance weaker models while refining top-tier ones. Compared to TTS method Sana-1.0 1.6B (73.98), GenPilot enables DALL-E 3 (74.08) to surpass it and FLUX.1 schnell (73.32) to perform comparably, highlighting GenPilot's effectiveness through test-time prompt optimization. Although some subcategories show slightly lower scores, GenPilot achieves the highest performance on average, revealing a balance in optimization across different aspects.

4.2.2 Qualitative Evaluation

On the second row on the left in Figure 3, the image generated by SDv1.4 with GenPilot effectively excludes the unwanted items, in contrast to the other three images, which fail this exclusion and contain them to varying extents. These qualitative examples in Figure 3 vividly illustrate the effectiveness and generalization ability of GenPilot in handling challenging prompts, including accurate attribute binding such as counting, complex compositions, spatial reasoning, unrealistic description and the effective processing of negative constraints. More qualitative analysis can be found at Appendix I.

4.3 Comparison on GenEval benchmark

4.3.1 Quantitative Evaluation

As illustrated in Table 2, GenPilot applied to the two base models on GenEval, including FLUX.1 schnell and PixArt-α, compared to the Prompt Engineering(PE). GenPilot improves FLUX.1 schnell from 65.82% to 69.60%, outperforming PE (66.59%) with notable gains in position, color, and number-related tasks. Similarly, PixArt-α with GenPilot achieves 48.54%, surpassing both the base model (46.73%) and its PE-enhanced version (45.98%). These results highlight the capability of GenPilot to improve the image quality and text-to-image consistency across models and prompt types. However, in subcategories such as single- and dual-object scenes, where the base models are already highly proficient, GenPilot shows comparable or slightly lower performance, aligning with its design goal of refining unsatisfactory generations.

4.3.2 Qualitative Evaluation

Figure 4 shows the qualitative results of FLUX.1 schnell and PixArt-α on the GenEval benchmark. As shown in Figure 4, with GenPilot, FLUX.1 schnell and PixArt-α can accurately generate the position-related image and unrealistic prompt, compared to failures in PE and base models. The qual-

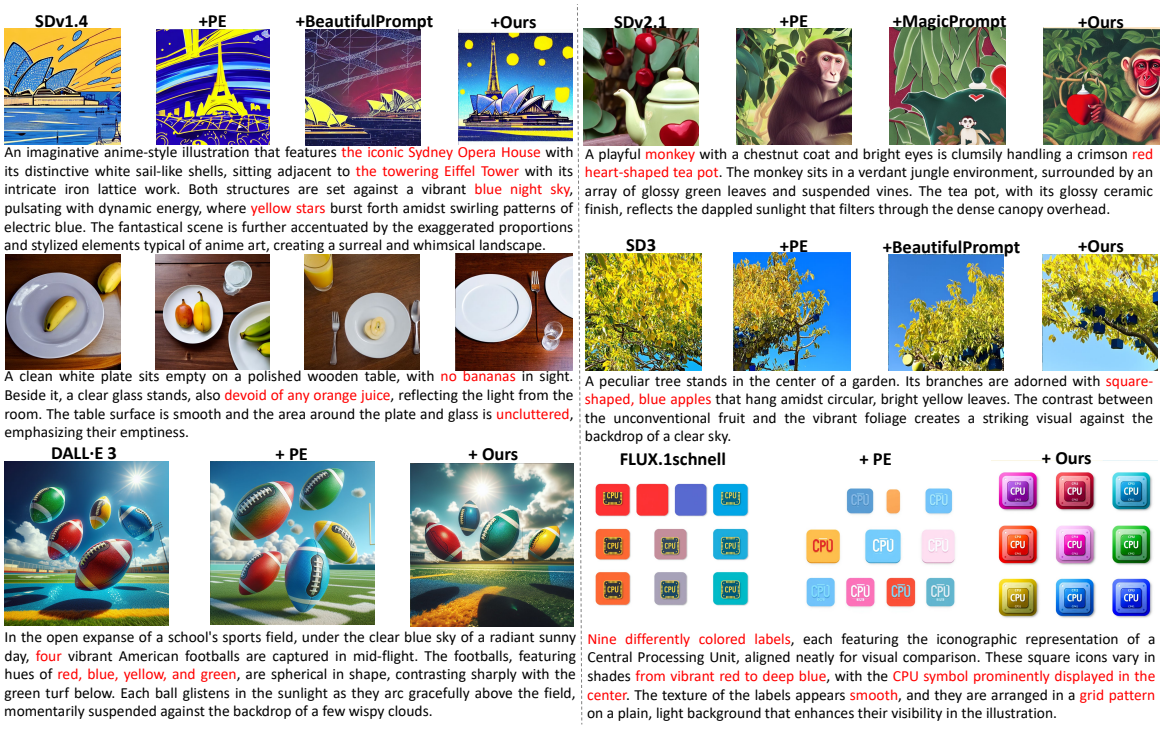


Figure 3: Qualitative comparison with different methods on the DPG-bench challenging dataset on different generative models. The left columns display two generations from SDv1.4 and one from DALL-E 3. The right columns present the results from SDv2.1, SD3, and FLUX.1 schnell. For the SD series, we select the best from BeautifulPrompt and MagicPrompt, along with the PE methods for comparison. GenPilot consistently generates error-free images across all scenarios, demonstrating its superiority in synthesizing high-quality and accurate images.

Model	Overall	Position	Color_Attr	Colors	Sin_Obj	Two_Obj	Counting
FLUX.1 schnell	65.82	29.00	44.50	76.06	**99.69**	**86.62**	59.06
FLUX.1 schnell + PE	66.59	31.75	46.50	80.32	99.06	85.35	56.56
FLUX.1 schnell **+ Ours**	**69.60**	**41.50**	**52.25**	**81.38**	97.19	84.60	**60.62**
PixArt-α	46.73	8.25	7.00	77.66	**98.44**	**50.00**	39.06
PixArt-α + PE	45.98	8.50	8.50	71.54	97.81	45.45	44.06
PixArt-α **+ Ours**	**48.54**	**9.25**	**9.25**	**81.91**	95.31	49.24	**46.25**

Table 2: Quantitative results on GenEval benchmark. All scores are reported as percentages (%). The '%' symbol is omitted for brevity. Sin_Obj refers to a single object, and Two_Obj represents two objects. Color_attr is the color attribute in short. GenPilot demonstrates superior overall generation ability both on FLUX.1 schnell and PixArt-α, with a great improvement on most of the subcategories.

Figure 4: Qualitative examples on GenEval. The left columns show the comparison of FLUX.1 schnell, FLUX.1 schnell and PE for enhancement, and FLUX.1 schnell with GenPilot. The right columns provide the results of PixArt-α, PixArt-α and PE for enhancement, and PixArt-α with GenPilot. GenPilot achieves great success in both position processing and unrealistic prompt generation, highlighting its potential and generalization to improve the quality of images.

Model	Overall	Position	Color_Attr	Colors	Sin_Obj	Two_Obj	Counting
FLUX.1 schnell	65.82	29	44.5	76.06	**99.69**	**86.62**	59.06
+ Ours-M	66.05	35.75	46.75	74.73	98.44	82.83	57.81
+ Ours-C	66.27	35.75	46.75	77.66	97.19	83.08	57.19
+ Ours	**69.60**	**41.50**	**52.25**	**81.38**	97.19	84.60	**60.62**

Table 3: Ablation study results on different variants of our method on GenEval with '%' omitted. "+ Ours-M" refers to FLUX.1 schnell with GenPilot but removing the memory module, and "+ Ours-C" represents the variant without clustering. GenPilot performs the best with comprehensive improvements, illustrating the effectiveness of these modules.

itative results reveal the potential of the generalization ability and effectiveness of GenPilot to improve the text-to-image alignment. More qualitative results are in Appendix J.

4.4 Ablation Study

To comprehensively evaluate the contributions of each core component in GenPilot, we conduct ablation studies on the GenEval benchmark, using FLUX.1 schnell. In this section, we systematically evaluate the impact of the error integration, the clustering, and the memory module. As shown in Table 3, GenPilot achieves the highest score of 69.60%, and the score without memory is 66.05%, and the score without clustering is 66.27%, declining across various subcategories. The results demonstrate the significance of the memory module and clustering algorithm, as the memory provides references and clustering optimizes the search space on text, iteratively scaling up the performance of optimization. Simultaneously, even removing those key components, GenPilot variants still outperform the base model, revealing the effectiveness of the rest modules in GenPilot.

VQA-based	Caption-based	Integration
3.78	3.95	**4.62**

Table 4: Comparison on the accuracy and coverage of error analysis rated by GPT-4o on VQA-based methods, caption-based method, and the integration, highlighting the importance of components in GenPilot.

Meanwhile, we explore the performance of the error integration strategy by GPT-4o (OpenAI et al., 2024) to score the quality of error analysis in VQA-based, caption-based, and integration results from 1 to 5, and 5 is regarded as the best. As illustrated in Table 4, though analysis from both methods provides effective information, the integration strategy highlights the effectiveness of full coverage and accuracy with a 4.62 score. A qualitative comparison example can be found at Appendix H.

During the inference stage, the time cost of optimization increases based on the number of iterations, error, candidate prompt, sentence, the T2I models, and the image batch size. On average, GenPilot costs 8 minutes for one prompt, including the time cost of the image generation process of SDv1.4, when iteration is 3, cluster number is 3, the total number of candidate prompts is 7, and an image batch contains 3 images.

More experiments on visualization of clustering are provided on Appendix C, semantic analysis on embedding is at Appendix D, and analysis on POS distribution shift is shown in Appendix E.

4.5 Patterns on Error Analysis and Refinement

We release 35 patterns and their corresponding refinement strategy summarized by GPT-4o, along with cases for better understanding in Appendix K.

5 Conclusion

In this work, we propose GenPilot, a flexible and effective test-time prompt optimization multi-agent system for enhanced text-to-image generation, aiming to address the semantic gap and the compositionality catastrophe, especially for complicated and lengthy prompts. Unlike previous approaches, GenPilot performs test-time scaling directly on the input prompt space, formulating it as a search problem to find the optimal prompts for T2I models, iteratively refining the prompt with clustering algorithm. The system integrates modular agents for error analysis, prompt editing, multi-modal LLM scoring, and memory-based feedback to support dynamic adjustment. Extensive experiments on GenEval and DPG-bench demonstrate the effectiveness and superiority of GenPilot over other methods, highlighting the potential of test-time prompt optimization for enhancing T2I generation. We further release a set of common error patterns and refinement strategies, providing a practical resource for future research on prompt controllability and optimization.

Limitations

Despite the improved performance of GenPilot in various scenarios, there are still a few challenges to address. Firstly, although our framework avoids model fine-tuning, it introduces additional computation time during inference, which may be non-trivial in latency-sensitive applications. Meanwhile, the performance of GenPilot is influenced by the multimodal large language models used for the agent, which may harm the performance if users utilize a less capable MLLM that lacks sufficient understanding of multimodal information.

Ethics Statement

In this work, we utilize Qwen2-VL 72B Instruct and GPT-4o as tools for agent or evaluation, along with the dpg-bench and GenEval datasets. We fully considered the ethical problems when applying the large language models. The DPG-bench dataset is licensed under Apache 2.0, and the GenEval dataset is available under the MIT license. Our usage strictly follows the licenses and their intended purposes. The data we utilize do not contain any information about unique individual people.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, and 1 others. 2023. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8.

Steven Bird. 2006. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 interactive presentation sessions*, pages 69–72.

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

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. 2023. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *Preprint*, arXiv:2310.00426.

Jaemin Cho, Yushi Hu, Roopal Garg, Peter Anderson, Ranjay Krishna, Jason Baldridge, Mohit Bansal, Jordi Pont-Tuset, and Su Wang. 2024. Davidsonian scene graph: Improving reliability in fine-grained evaluation for text-to-image generation. *Preprint*, arXiv:2310.18235.

Yuchen Fu, Zhiwei Jiang, Yuliang Liu, Cong Wang, Zexuan Deng, Zhaoling Chen, and Qing Gu. 2024a. Ap-adapter: Improving generalization of automatic prompts on unseen text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 37:98320–98346.

Yuchen Fu, Zhiwei Jiang, Yuliang Liu, Cong Wang, Zexuan Deng, Zhaoling Chen, and Qing Gu. 2024b. Ap-adapter: Improving generalization of automatic prompts on unseen text-to-image diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, pages 98320–98346. Curran Associates, Inc.

Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. 2023. Geneval: An object-focused framework for evaluating text-to-image alignment. *Preprint*, arXiv:2310.11513.

Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. 2023. Optimizing prompts for text-to-image generation. *Preprint*, arXiv:2212.09611.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2022. Clipscore: A reference-free evaluation metric for image captioning. *Preprint*, arXiv:2104.08718.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Preprint*, arXiv:2006.11239.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. 2024. Ella: Equip diffusion models with llm for enhanced semantic alignment. *Preprint*, arXiv:2403.05135.

Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. 2025. T2i-compbench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. *Preprint*, arXiv:2307.06350.

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

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. 2025. Inference-time scaling for diffusion

models beyond scaling denoising steps. *Preprint*, arXiv:2501.09732.

James MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, volume 5, pages 281–298. University of California press.

Oscar Mañas, Pietro Astolfi, Melissa Hall, Candace Ross, Jack Urbanek, Adina Williams, Aishwarya Agrawal, Adriana Romero-Soriano, and Michal Drozdzal. 2024. Improving text-to-image consistency via automatic prompt optimization. *Preprint*, arXiv:2403.17804.

Pierre Moral. 2004. *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. Springer.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. *Preprint*, arXiv:2305.03495.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

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.

Kiran Ramnath, Kang Zhou, Sheng Guan, Soumya Smruti Mishra, Xuan Qi, Zhengyuan Shen, Shuai Wang, Sangmin Woo, Sullam Jeoung, Yawei Wang, Haozhu Wang, Han Ding, Yuzhe Lu, Zhichao Xu, Yun Zhou, Balasubramaniam Srinivasan, Qiaojing Yan, Yueyan Chen, Haibo Ding, and 2 others. 2025. A systematic survey of automatic prompt optimization techniques. *Preprint*, arXiv:2502.16923.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. *Preprint*, arXiv:2112.10752.

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. *Preprint*, arXiv:1707.06347.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. 2025. A general framework for inference-time scaling and steering of diffusion models. *Preprint*, arXiv:2501.06848.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding. *Preprint*, arXiv:2104.09864.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. 2024a. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6).

Ruochen Wang, Ting Liu, Cho-Jui Hsieh, and Boqing Gong. 2024b. On discrete prompt optimization for diffusion models. *Preprint*, arXiv:2407.01606.

Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. 2024c. Genartist: Multimodal llm as an agent for unified image generation and editing. *Preprint*, arXiv:2407.05600.

Qiucheng Wu, Yujian Liu, Handong Zhao, Trung Bui, Zhe Lin, Yang Zhang, and Shiyu Chang. 2023. Harnessing the spatial-temporal attention of diffusion models for high-fidelity text-to-image synthesis. *Preprint*, arXiv:2304.03869.

Enze Xie, Junsong Chen, Yuyang Zhao, Jincheng Yu, Ligeng Zhu, Chengyue Wu, Yujun Lin, Zhekai Zhang, Muyang Li, Junyu Chen, Han Cai, Bingchen Liu, Daquan Zhou, and Song Han. 2025. Sana 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. *Preprint*, arXiv:2501.18427.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2025. A survey of large language models. *Preprint*, arXiv:2303.18223.

A A Detailed Explanation of Metadata

To offer more structural data for the agent to better understand, we design a structural data called metadata. Initially, the main components in metadata are error analysis, error mapping, question list, history feedback, the original prompt, and the original image generated from that prompt. We provide the error analysis and mapping, along with the original prompt, image, and history for prompt refinement, and offer the question list for the MLLM scorer. With the structured metadata, the agent is capable of better understanding the context and efficiently retrieving data.

B A Detailed Explanation of Scorer Subcategory

We design the rules from the following three aspects inspired by T2I-CompBench (Huang et al., 2025).

Attribute binding: Attribute binding refers to the ability to correctly associate specific properties with the object as described in the prompt, including color, number, shape, state, and texture of the object.

- The color is used to evaluate whether the correct color is applied to a certain object or not, especially when multiple objects have different color specifications.
- The number specifies the exact count of objects. Models might struggle with precise counts, failing to make the very approximate number of different objects.
- The shape refers to the external form or geometric shape of an object, ranging from simple and concrete forms to complex and abstract structures. For example, in the prompt "A person with a muscular build", muscular build refers to the shape of the human.
- The state is a broad category referring to the condition, mode of being, phase, or dynamic activity of an object or entity at a particular time. It contains physical conditions for instance, "ripe" in "ripe bananas", the action, such as the "running" in the prompt "A dog running in a field", the emotional state, for example, the "surprised" in "A surprised cat", and the functional state such as "open" in "An open door", and the texture describes the surface of the object, including smoothness, roughness, softness and so on.

Relationship and position: In addition to the attribute of the object, prompts often include information about how these objects are interconnected and their positions within the scene. These relationships involve various types of interactions. For example, one object acting upon another, such as "a dog catching a ball", and the objects in occlusion, such as "a tree partially obscuring a view of the house", and simple containment or support, such as "Apples in a basket". Similarly, positional information describes where the object is located, either relative to one another or at the absolute position within the image frame.

Background information and style: Finally, we also defined a further descriptive aspect, the background information and style. The background information encompasses details about the scene that are distinct from the main object, including the style and overall atmosphere in the image.

C Clustering Analysis

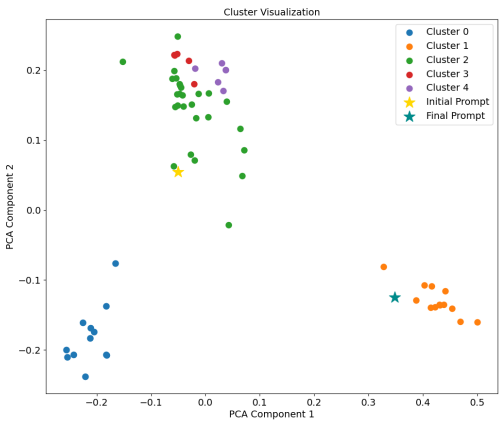


Figure 5: Visualization of clustering result on one case with the number of clusters set to be 5.

Figure 5 vividly shows how our clustering algorithm works. The initial prompt (the yellow star) is close to cluster 2 in green, next to cluster 3 in red and cluster 4 in purple. However, cluster 0 in blue and cluster 1 in orange are far from the initial point. The relevant score of cluster 1 is 5.0 on average, which indicates it as the best prompt this turn, while clusters 2, 3, and 4 with a lower score, such as 4.3 on average. Initially, the candidate prompts generated from the prompt refinement agent might

still predominantly cluster around. As iterations progress, GenPilot explores more directions, including the clusters 0 and 1 illustrated in Figure 5. In this case, cluster 1 represents the optimized area that T2I models prefer to generate high-quality images. By generating multiple samples and scoring them into clusters, GenPilot successfully scales the prompts and optimizes them, revealing the effectiveness and potential of the test-time prompt optimization for improving the image quality.

Another example with an image and a prompt can be found at Appendix F.

D Semantic Analysis on Embedding

Method	GenEval	DPG-bench
Origin	0.2443	0.3705
BeautifulPrompt	0.2573	0.3527
PE	**0.3193**	0.3724
Ours	0.2981	**0.3944**

Table 5: Comparison of the semantic similarity analysis with extremely detailed descriptions by GPT-4o.

We conduct a semantic similarity analysis at the embedding level to evaluate whether prompt optimization leads to richer textual descriptions on the GenEval and DPG benchmark. We generate extremely detailed and specific descriptions as reference prompts using GPT-4o with the particular instruction shown in Appendix G. We then measure the cosine similarity between each method's prompt embeddings and the reference for completeness and semantic content. As shown in Table 5, our method achieves the highest average similarity scores on DPG-bench, given the highest score of performance on DPG-bench in Table 1, indicating that GenPilot introduces meaningful and effective details into the original prompt. And GenPilot is highly competitive on GenEval (0.2981) against other methods like Origin, BeautifulPrompt, and PE. When the prompt is relatively short and simple, rewriting or expanding the abstract prompts significantly improves semantic richness, which positively influences generation. However, on GenEval, we observe that though PE reaches the highest score of similarity, the whole performance of PE is lower than ours when the generative model is FLUX.1 schnell, and even lower than PixArt-α itself. Therefore, higher semantic similarity for more information included does not always lead to better visual results. Simply expanding the prompt, especially for complex and lengthy prompts, may not enhance the image result obviously. In contrast, GenPilot consistently turns the semantic gains into meaningful performance improvements, highlighting its effectiveness and necessity.

E Analysis on POS Distribution Shift

To explore the impact of the linguistic structure of generated prompts, we conduct a part-of-speech (POS) level analysis comparing the original prompts and the optimized ones with NLTK (Bird, 2006). All tools and functions were used with default settings. We focus on adjectives, nouns, verbs, adverbs, pronouns, and so on. A common trend can be found among PE, BeautifulPrompt, and Ours, revealing that adding adjectives may help with more specific information for image generation. Both on DPG-bench and GenEval, our method increases the proportion of adjectives and proper nouns, indicating that prompts generated by GenPilot tend to be more descriptive via adjectives and more specific via proper nouns.

F More Detailed Case Analysis

In this section, we provide a more detailed case during the iterations, as shown in Figure 7. The main error, according to the error mapping sentence in the original picture in the first row, is the number of baozi. In the original prompt, baozi should be 6, while in the image, it only has three. The best sampled prompt in the next round modified the prompt with "exactly" and some other specific descriptions, rated 4.1 in the end. In the second round, the prompt optimization agent tries to emphasize the number by repeating the keyword of six. However, it remains 5 baozi in the image, rated 4.3 by the MLLM scorer. Next round, the prompt optimization agent concludes the failures of the previous round, and makes an attempt to emphasize by adding an adverb. In round three, an image with 5 clearly visible baozi is generated, which is a minor improvement compared to the earlier round. For round 4, prompt optimization tries to change the adverb, which turn out to be successful, rated 5 in the end. After that, the correct modification, the image, and candidate prompt will be stored, as a stop signal for this error.

G System Prompt Template

In this section, we provide the system prompt used in GenPilot to guide the agent. Figure 8 represents

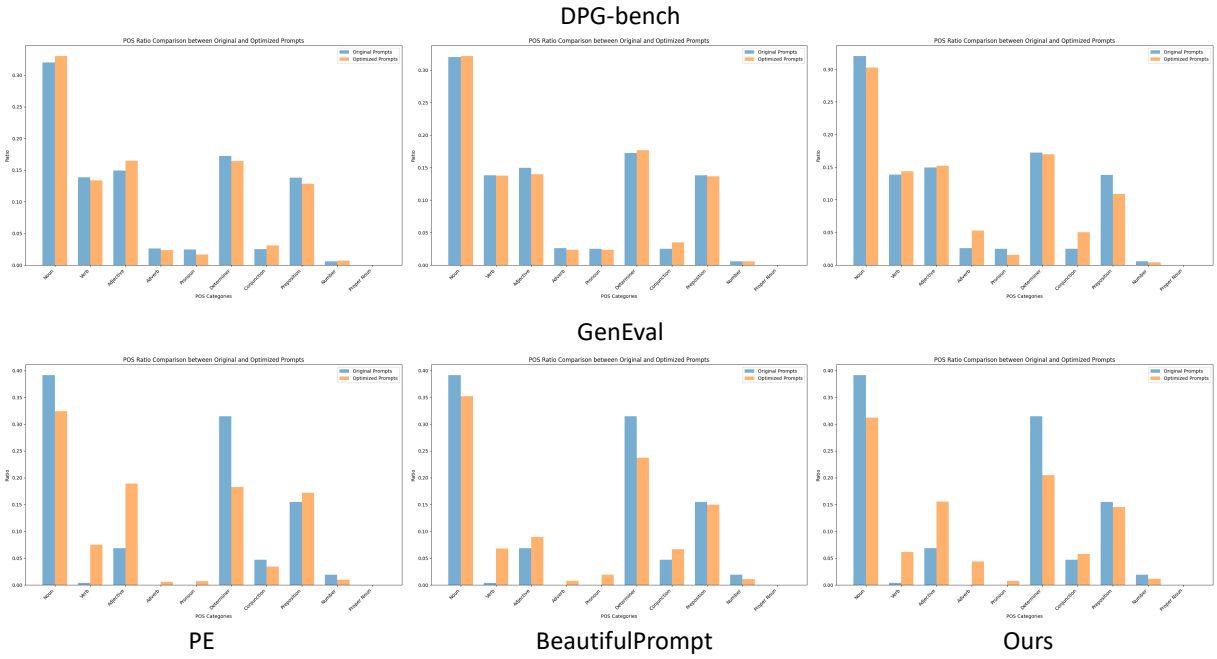


Figure 6: Analysis on POS Distribution Shift of PE, BeautifulPrompt, and Ours compared to original prompt on DPG-bench challenging subset and GenEval benchmark.

the system prompt we use for prompt engineering (PE) in the experiment. PE takes the original image and prompt as the input and generates the optimized prompts.

In Figure 9, we design the instruction for the memory module to store the summary of the detailed errors that occurred, offering a comprehensive reference for the next iteration.

Figure 10 is the system prompt we use to generate the detailed descriptive caption of the image.

In Figure 11, part a represents the instructions for the error integration agent to verify and summarize the errors. The agent will produce a complete list of errors, including patterns and details. Part B in this figure plays the role of the branch merge agent to combine the modified sentence into the complete prompt.

The instructions for GPT-4o to summarize the error pattern and the refinement pattern are shown in Figure 12, part B. The system prompt in A in the Figure 12 is used to rate the accuracy and coverage of VQA-based, caption-based, and integration results.

Following the sequence of A, B, and C, Figure 13 shows the prompt used for the VQA in MLLM scorer, the VQA in error detection, and the error mapping.

In Figure 14, we provide the instructions for MLLM rating (A) and question list generation (B).

Figure 15 shows the system prompt for prompt refinement agent (A) and caption-based error detection (B).

H Detailed Example on Comparison for Error Analysis Methods

As shown in Figure 16, GenPilot takes advantage of both methods and verifies each result to generate a complete error analysis.

I More Results on DPG-bench

In this section, more results conducted on the DPG-bench are illustrated. As shown in Figure 19, we compare the FLUX.1 schnell with PE-optimized and GenPilot-optimized images. For the first row, the main objects in the original prompt are the Pyramids, the Sphinx, an astronaut, and Earth. Our method clearly renders the iconic Great Pyramids, the Sphinx, the astronaut from behind, and a vividly contrasting Earth, while PE provides an astronaut from the front, and the FLUX image mistakenly combines the Pyramids and the Sphinx together. In the second row, our approach successfully generates “two square-shaped pink erasers” next to a toilet, compared to the square erasers on the toilet in the FLUX image and the tube-shaped erasers in the PE image. Moreover, PE image misses the blue bath mat and the handle in the background. Finally, in the challenging prompt of an aged room

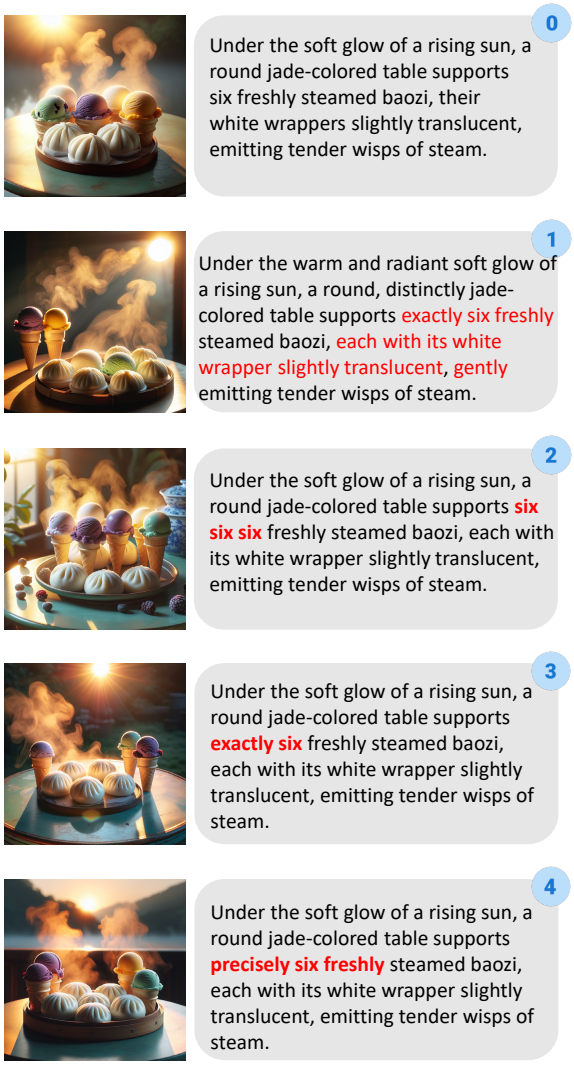


Figure 7: A detailed sample of iterations and the results. 0 represents the initial start point. GenPilot optimizes the sentence with the error "the number of baozi" and achieves the accurate synthesis on the fourth round.

with multiple projectors and keyboards in the third row, GenPilot accurately generates 4 spherical, silver projectors, in contrast to the 5 and 2 in FLUX image and PE image. These qualitative comparisons in Figure 19 demonstrate the superior ability of GenPilot to interpret complex prompts for enhanced image generation. Our approach accurately renders distinct objects with their specified attributes, correct spatial relationships, and the precise number, revealing the effectiveness and potential of GenPilot to improve image quality in text-to-image synthesis.

J More Results on GenEval

In this section, we provide more qualitative experimental results on the GenEval benchmark, as

Based on the image and the original prompt, please optimize the original prompt so that the text-to-image generation model could generate better image. NOTE that you should only give the optimized prompt without any other words.

Figure 8: The system prompt for prompt engineering (PE) with the initial prompt and image as the input.

You are tasked with analyzing and summarizing errors related to an AI-generated image. I will provide a list of text, your goal is to:
Analysis the errors from both pieces of text to produce a complete list of all errors in short.

Please point out the important object or relationship that leads to the error.
Ensure no key detail or information from either text is overlooked while summarizing the errors.
Note that both texts are generated from different AI models, so you must have to judge from comprehensive perspective. Split each error with '\n'.

Input: {prompt}

Figure 9: The system prompt to summarize and explain the reasons for the MLLM agent rating score.

shown in Figure 20 and Figure 21. Though Flux.1 schnell and PixArt-α have achieved relatively great performance, sometimes they may fail, such as in the unrealistic ones and position-related prompts. In Figure 20, when the prompt describes an uncommon scene, "a photo of a train above a potted plant", Flux.1 schnell generates an image of a train behind a plant, which is consistent with real-world principles. With GenPilot, Flux.1 schnell can accurately generate an unreal scene with a train floating above a plant.

PixArt-α in Figure 21 is not skilled in drawing shapes and details, especially for the combination of multiple objects. In contrast, with GenPilot, PixArt-α is capable of generating specific details, for example, the image in the second row of a baseball glove.

The qualitative results highlight the effectiveness and capability of seamlessly applying to various models.

You are an assistant that helps with image description.
Given the image, provide the following information:
- A detailed description of the image

Figure 10: The system prompt for generating the corresponding caption of the image.

K Detailed Pattern on Error and Optimization Analysis

In this section, we list the patterns of errors and the refinement strategy summarized by GPT-4o based on the original prompt and optimized prompt. The system prompt for GPT-4o can be found at Appendix G. We release 35 patterns and their corresponding refinement strategy, along with cases for better understanding.

Quantity Errors: Quantity Errors refer to the number of objects in the generated image that does not match the description in the prompt. To address this issue, the optimized prompt employs a strategy of repeating quantity keywords and incorporates the adverbs "exactly" and "precisely" to enhance precision. For example, the original prompt did not guarantee the correct depiction of exactly eight chairs. The optimized prompt emphasizes the exact number of "eight chairs" and uses "exactly" to reinforce the precision of the quantity, thereby ensuring that the generated image accurately reflects the specified number of objects.

Spatial Positioning Errors: Spatial Positioning Errors arise when objects in the generated image are placed incorrectly relative to one another. The optimized prompt addresses this by introducing a more systematic approach to spatial description. It explicitly defines objects' coordinates, angles, and distances to other objects within a three-dimensional framework. For example, the original prompt caused errors in the depiction of the boy's position relative to the woman, resulting in inconsistencies with the intended positioning. The optimized prompt clarifies spatial positions with terms like "precisely" and "directly behind" to reduce ambiguity and ensures that spatial relationships are conveyed unambiguously, thereby minimizing spatial positioning errors and eliminating inconsistencies in the generated image.

Texture Errors: Texture Errors happen when the surface textures of objects in the generated image do not match real-world expectations or appear missing. The optimized prompt tackles this issue by introducing more detailed texture descriptions and emphasizing them. For example, the original prompt failed to highlight the frosty texture on the boards, which is inadequately visible. The optimized prompt provides more detailed descriptions of the texture and repeatedly emphasizes the frosty texture on both ice and boards to correct texture visibility errors and make the generated image more realistic.

Color Errors: Color Errors mean the colors of objects in the generated image deviate from the specified requirements. The optimized prompt introduces a more systematic approach to color description by incorporating precise color terminology and describing colors across multiple dimensions such as hue, brightness, and saturation. For example, the original prompt's lack of specificity in defining the pear's color resulted in variations and potential color mismatches in the output. To address this, the optimized prompt employs exact color references like "Pantone 376C" to specify the pear's color, thereby reducing ambiguity and enhancing color accuracy in the generated image.

Shape Errors: Shape Errors occur when the shapes of objects in the generated image do not meet the requirements or are illogical. The optimized prompt tackles this issue by repeatedly emphasizing the unique shape of the object and adding detailed descriptions. For example, the glasses on the horse were not clearly differentiated in terms of color and shape from the original prompt. The optimized prompt provides a clearer distinction for the types of glasses by detailing their specific colors and frame shapes through expanded descriptions, thereby enhancing the accuracy and logic of the object's shape in the generated image.

Proportion Errors: Proportion Errors refer to the scale and size of objects in the generated image are imbalanced or illogical. The optimized prompt addresses this by providing detailed descriptions of object proportions and introducing specific measurement references. For example, the original prompt failed to effectively depict the size relationship between the oversized blue rubber ball and the net and hoop. The optimized prompt emphasizes the impossibility of the ball passing through the hoop by enhancing the description of the ball's oversized nature, thereby ensuring a more realis-

A

Your task is to analyze and summarize errors related to an AI-generated image.
I will provide two pieces of text and the original prompt:
First text: analysis of potential errors from the visual question answering conversation. If None, it means no error been analyzed here.
Second text: analysis of potential errors between the breakdown prompt and the caption of image. If None, it means no error been analyzed here.
Original prompt: Full origin prompt. The original prompt is the ground truth.

Your goal is to:
Comparing to the original prompt, analysis the errors from both pieces of text to produce a complete list of all errors.

Follow these rules:
Rule1: Please point out the most important object or relationship that leads to the error in short. Just say the key point.
Rule3: You must have to judge whether the text is an error or not based on the full original prompt.
Rule4: If some thing is not mentioned in the whole orinal prompt, then it is an error and you should point out.
Rule5: If the text say that some thing is not mentioned in breakdown prompt, you should analysis based on the original prompt. If it matches with original prompt, then just ignore it.
Rule6: List all errors carefully and split each error with '\n'.
Rule7: If no error, just say None
Note that do not say "the caption mentions something but the original prompt something else". In that case, just say something is wrong, it should be...

Here is an example:
Input:
Text 1:
The monkey is on the blue bike not a green bike. The monkey should be on a green bike.
The caption mentions a bottle in the image but it does not appears in the breakdown prompt.([here for setences like this, you have to judge whether it is an error based on the full original prompt, in case of bottle mentioned in some other places in the propmt])

Text 2:
Error:
- The prompt describes a green bike with a monkey on it; however, the caption introduces a blue bike, which should be green.
YES.The bottle is not the primary focus of the image; the focus appears to be on the monkey.

Original full prompt:
A monkey is sitting on a green bike and a bottle on the road.

Answer:
Error1: The color of the bike is wrong. The monkey should be on a green bike, not a blue one.

Input: {prompt}

B

You are tasked with integrating modified sentences into an original description while keeping changes minimal and maintaining grammatical correctness. Here is the situation:

I will provide:
1. The original description (prompt) that was used to generate the images.
2. A list of modified sentences that address specific errors found in the generated images.

Your task is to:
Replace the corresponding sentences in the original description with the provided modified sentences.
Ensure all other sentences in the original description remain unchanged.
Keep the overall description coherent, concise, and grammatically correct with the smallest necessary adjustments.

Note that keep changes minimal and do not delete other sentence or phase or other information in the original prompt. Just replace and merge without information missing.
Note that you just say the whole prompt after merge and you do not need to output any other words or prompts.
Note that if no error or none changes just say the original prompt.
Note that the prompt after replacement should be better for generative models to follow the prompt when generating images.
Do not missing other information in the original prompt!

Here is an example
Input:
Prompt:
An icy landscape. A vast expanse of snow-covered mountain peaks stretches endlessly. Beneath them is a dense forest and a colossal frozen lake. Three people are boating in three boats separately in the lake. Not far from the lake, a volcano threatens eruption, its rumblings felt even from afar. Above, a ferocious red dragon dominates the sky and commands the heavens, fueled by the volcano's relentless energy flow.
Modified sentence:
Beneath them is a dense dense dense forest and a colossal frozen lake.
Answer:
An icy landscape. A vast expanse of snow-covered mountain peaks stretches endlessly. Beneath them is a dense dense dense forest and a colossal frozen lake. Three people are boating in three boats separately in the lake. Not far from the lake, a volcano threatens eruption, its rumblings felt even from afar. Above, a ferocious red dragon dominates the sky and commands the heavens, fueled by the volcano's relentless energy flow.

Just say the whole complete prompt after merge without any other words.

Input: {prompt}

Figure 11: The system prompt for integration error analysis (A), which combines and verifies the error analysis from VQA-based methods and caption-based analysis, and the instructions for the branch-merge agent for merging the modifications into the original prompt (B).

A

Please act as a professional image analysis assistant to help me score the errors in text-to-image generation.

Background information:
I am conducting research on text-to-image generation and have obtained a text and the image generated from this text-to-image.

Task requirements:
You need to quantitatively score the obtained errors on a scale of 1-5 points (can not be a decimal).

The scoring criteria are as follows:
- **5**: The error identification is complete and accurate, capturing all major differences between the image and the text. The description of errors is clear and precise.
- **1**: The error identification is completely incorrect or missing, failing to capture any significant differences. The description of errors is unclear or irrelevant. You should focus on whether there are any errors that have not been identified or mentioned. Are there major differences between the image and the text that were overlooked? Did the error identification miss any important semantic errors?

Please output the result in the following JSON format:
```json
{
   \"scores\" : [score],
   \"reasons\" : \"[reason]\"
}
```
Replace [score] with the numerical score and [reason] with a brief explanation of the score.

Please ensure that the output is ONLY the JSON format as specified above.

B

Could you please act as a professional image analysis assistant to help me analyze the prompts before and after optimization in text-to-image generation and the corresponding generated images?
The following are the original prompt and the optimized prompt:
The first four pictures are the ones before optimization, and the last four are the ones after optimization.
prompt before optimization: a photo of a bench
Optimized prompt: a photo of a wooden bench with metal armrests and supports, set against a simple and neutral background with no additional objects or elements.
Please analyze according to the following steps and directly output the final condensed error mode and modification mode:
Compare the original prompt with the optimized one to identify the differences between them in terms of described content, word choice, structure, etc. Focus on the aspects in which the optimized prompt has been improved, such as whether specific details have been added, whether more precise vocabulary has been used, and whether the hierarchy and logic of the description have been adjusted, etc.
2. By combining the original and generated images, analyze the impact of these differences on the image generation effect, and summarize the error patterns caused by the original prompt, such as:
[Quantity Ambiguity] : Ambiguity in quantity expression (such as "several" instead of "eight") leads to quantity deviation in the generation
[Single color] : Only the basic color (" green ") is used to describe without distinguishing the differences in saturation/lightness
[Implicit relationship] : The spatial relationship is not clearly defined (the positional association between "field" and "cabbage" is not clear)
[Lack of texture] : Completely ignoring the description of the surface texture and volume of the object
[Brief description of the environment] : Only the scene elements are mentioned without creating a complete atmosphere
[Proportion distortion] : Short text causes an imbalance in the proportion between the main subject and background elements
3. Similarly, by combining the original and generated images, analyze in which aspects the optimized prompt has been modified and how these modifications have addressed the errors in the original prompt. Summarize the optimized modification patterns and focus on the optimization methods and techniques, such as:
The optimized prompt clearly indicates the specific quantity of the object by repeating the keyword "three apples".
The optimized prompt elaborately describes the spatial positions of the objects by elaborating that "apples are arranged from left to right on the table in sequence".
The optimized prompt clarifies the color of the object by emphasizing "The apple is red".
Disassembly description: The optimized prompt adds texture details of the object by disassembling the description "The surface of the apple is smooth and shiny".
4. Summarize and classify the above-mentioned error patterns and modification patterns respectively. There is no need for a one-to-one correspondence between the two. The categories should be as simple as possible to avoid being overly templated. Make the categories general and targeted, and be able to clearly reflect the problems of the original prompt and the improved strategies after optimization.
Specifically, as follows:
Error mode:
[Category 1] : [Briefly describe the specific error caused by the original prompt]
[Category 2] : [Briefly describe the specific error caused by the original prompt]
...
Modification mode:
[Category 1] : [Briefly describe the modification strategy of prompts before and after optimization]
[Category 2] : [Briefly describe the modification strategy of prompts before and after optimization]
...

Figure 12: The system prompt designed for evaluating the accuracy and coverage of error analysis (A), and the instructions to summarize the systematic patterns of errors and optimization strategies (B).

A

You job is to answer the question based on the AI-generated image and check whether this object aligns with the breakdown of the original prompt.
Your standards need to be very strict. For example, if the question is "Is there apple here?", please answer "No" when the apples in the image are hard to be recognized as apples or hard to be found.
I will give you the image, the prompt and a question.
Note that you first must answer YES or NO. If NO, please analyze the error in the image in one sentence.

Here is an example:
Input:
prompt:
A dense forest surrounds a large frozen lake below the mountains.
Question:
Is the forest dense around the lake, as described?
and an AI-generated image.

Answer:
NO. The forest should be around the lake, but in image this forest is not around the lake.

Input: {prompt}

B

Your job is to answer the question based on the AI-generated image and check whether this object aligns with the breakdown of the original prompt.
I will give you the image, the breakdown prompt and a question.
You should 1. answer the question based on image 2. check whether the objects in the answer are consistent with the breakdown prompt.
Please check the image carefully.
Your standards need to be very strict. For example, if the question is "Is there apple here?", please answer "No" when the apples in the image are hard to be recognized as apples or hard to be found.
If they match, just say YES. If not, point out the detailed errors.

Here is an example:

Input:
Breakdown prompt:
A dense forest surrounds a large frozen lake below the mountains.
Question:
Is the forest dense around the lake, as described?
an AI-generated image.

Answer:
Error: the forest is not around the lake.

Input: {prompt}

C

Your job is to analyze errors in AI-generated images and mapping these errors back to specific sentence of the original prompt.
I will provide: 1. The original prompt, which is a detailed description broken into sentences or phrases. 2. A list of errors identified in the generated images.

Your task is to:
Analyze the errors and determine which specific sentence or phrase in the original prompt each error is related to.
Clearly map each error back to its corresponding sentence or phrase.
Ensure that the mapping is accurate and that no detail from the errors or the prompt is overlooked.
Note that you must accurately paraphrase the corresponding sentence in the original prompt, without making any changes to the original sentence. Copying the sentence as it is should be the answer, and you do not need to output any other words or prompts.

If there is no error, just say NO ERROR. If there are errors, just say the sentence without any other words.

Here is an example:
Input:
Original Prompt:
An icy landscape. A vast expanse of snow-covered mountain peaks stretches endlessly. Beneath them is a dense forest and a colossal frozen lake. Three people are boating in three boats separately in the lake. Not far from the lake, a volcano threatens eruption, its rumblings felt even from afar. Above, a ferocious red dragon dominates the sky and commands the heavens, fueled by the volcano's relentless energy flow.
Identified Errors:
The forest around the lake is not dense, as the area is largely open with some trees visible.

Answer:
Beneath them is a dense forest and a colossal frozen lake.

just say the sentence without any other words.

Input: {prompt}

Figure 13: The system prompt for the VQA module in MLLM scorer (A), the VQA-based error detection (B), and the error mapping (C).

A

Your goal is to identify whether the improvements made in the new prompt lead to more accurate and effective image generation, addressing any errors in the previous image generation.
I will give you the original prompt (round 1), the error in round 1, a modified prompt (round 2), the image generated by modified prompt (round 2) and part of errors in round 2.

You should rate the modified prompt based on the aspects roles.
please rate the quality of the output by scoring it from 1 to 5 individually on alignment with each aspect.
- 1: strongly disagree
- 2: disagree
- 3: neutral
- 4: agree
- 5: strongly agree

aspects = {
"Attribute-Binding":"Evaluate whether the modified prompt improves the accuracy of attribute-object associations in the generated image. A perfect score of 5 indicates that the modified prompt successfully ensures all attributes (e.g., color, shape, texture, number) are correctly bound to their corresponding objects as described, while a 1 suggests the modifications introduced or failed to correct significant errors in attribute binding.",
"Object-Relationship": "Evaluate whether the modified prompt enhances the correctness of the relationships between objects in the generated image. This includes both spatial relationships (e.g., on the left of, near) and non-spatial relationships (e.g., holding, sitting on). A perfect score of 5 indicates all described relationships are accurately depicted after the modifications, while a 1 suggests the changes did not improve or worsened the depiction of these relationships.",
"Background-Consistency": "Assess whether the modified prompt improves the consistency and alignment of the background information or atmosphere in the generated image with the intended context. A perfect score of 5 indicates the background seamlessly matches the described setting or atmosphere after the prompt modification, while a 1 suggests significant mismatches or new errors were introduced."
}

Note that your answer should follow this, returun json format information:
```json
{
  "scores" : {
    "Attribute-Binding": [your Attribute-Binding score here],
    "Object-Relationship": [your Object-Relationship score here],
    "Background-Consistency": [your Background-Consistency score here],
  },
  "reasons" : {
    "Attribute-Binding": [the reasons why you rate this Attribute-Binding score. in short sentence],
    "Object-Relationship": [the reasons why you rate thisObject-Relationship score here. in short sentence],
    "Background-Consistency": [the reasons why you rate this Background-Consistency score here. in short sentence],
  }
}
```

Input: {prompt}

B

You will analyze AI-generated images based on their original prompts, which have been broken down into specific descriptions. Your task is to write one or more object-focused questions aimed at identifying possible errors in the images or clarifying their alignment with the descriptions.
I will provide you with: 1. A breakdown of the original prompt into object-specific descriptions. 2.A generated image.

Based on the provided image and descriptions, you should:
- Compare the image with the descriptions and identify any discrepancies.
- Formulate one or more clear questions focusing on specific objects or relationships in the image to help uncover or address errors.

Please make questions based on the following rules:
Rule1: Make questions about whether it contains every object mentioned in the text. Check every object for existence.
Rule2: The main aspect involves the objects contained in the text, together with their own attributes like their color, shape, texture, number, the positions of the objects, the relationship among these different objects, the background or the scene of the image.
Rule2: You should only check the items mentioned in original prompt for any inconsistencies.

For example:
if the original prompt is "On a calm afternoon, a soft blue linen cloth gently wraps a ripe, deep red apple, standing in stark contrast to the smooth, glossy surface of the fruit"
The main object here is 1. linen cloth 2. apple 3. calm afternoon
The linen cloth has attributes such as soft blue and the number is 1. The apple has attributes ripe, deep red, smooth, glossy surface and the number is 1. The relationship between these two object is that linen cloth gently wraps red apple. And the cloth is contrast to the apple.

So the questions should be:
Is there a linen cloth?
Is there an apple?
Is the linen cloth soft blue?
Is there exactly 1 linen cloth?
Is the apple exactly ripe, deep red, smooth, glossy surface?
Is there exactly 1 apple?
Does the linen cloth gently wrap the apple?
Does linen give you a feeling that contrasts with apples?

Note that split the questions with the '\n'. Please note that only provide the questions without any additional text.

Input: {prompt}

Figure 14: The rules and strategy for rating the generated images (A) and the structural output in JSON format. B represents how we generate the question centered on the object.

A

You are tasked with improving the accuracy of prompts for AI image generation. The image should correspond to the text. Here is the situation:

I will provide you with:
1. A list of errors identified in the generated images
2. the corresponding sentence(s) in original user's prompt where these errors are rooted.
3. some revision histories and their rating(ranging from 1 to 5).

The rating follow these roles
- Attribute Binding: The prompt should ensure accurate associations between the described attributes (e.g., color, shape, texture, number) and their corresponding objects in the generated image.
- Object Relationship: The prompt should guide the AI model to correctly represent the relationships between objects in the image (both spatial and non-spatial).
- Background Consistency: The prompt should result in a background that aligns with the context and atmosphere described, maintaining consistency with the overall scene.

Your task is to:
Please learn from these histories and rating to better modify the prompt. If rate score is lower than 5, which means this modification still can not solve the error.
Then you should learn from the experience and generate some different prompt.
The goal is to adjust the prompt so that the generation model produces images that align with the intended description.
For difficult tasks you can emphasize the key point for avoiding errors, for example, repeating number may be help or add some constrained words.

Note that the prompt after correction should be better for generative models to follow the prompt when generating images.
Note that you just say the prompt after correction and you do not need to output any other words or prompts.
Note that if no error or none changes or None just say NONE.

Here is an example
Input:
1. Errors:
The forest around the lake is not dense, as the area is largely open with some trees visible.
2. Sentence:
Beneath them is a dense forest and a colossal frozen lake.
3. Revision history:
[a list of histories, their rating and the reasons]
Answer:
Beneath them is a dense dense dense forest and a colossal frozen lake.

Just say the prompt after correction without any other words.

Input: {prompt}

B

Your job is to verify whether the caption of a generated image aligns with the breakdown of user's original prompt.
Follow these roles:
Rule1: The main aspect involves the objects contained in the text, together with their own attributes like their color, shape, texture, number, the positions of the objects, the relationship among these different objects, the background or the scene of the image. For example, for the text "A realistic photo with a monkey sitting above a green motorcycle on the left and another raccoon sitting above a blue motorcycle on the right", the caption should contain a monkey, a raccoon, a green motorcycle, a blue cycle. The position of the monkey is on the left, the raccoon is on the right. The monkey and the raccoon should sit on motorcycles. The photo should be a realistic scene, rather than oil painting style or others.
Note that I will only give you the original prompt, so you should compare the same object or relationship or style or background information in caption and the breakdown prompt.
Rule2: You should only check the items mentioned in breakdown prompt for any inconsistencies. If some other objects in caption and not in breakdown prompt, ignore them and only only check the thing mentioned in breakdown prompt for any inconsistencies.

If they match, just say YES. If something is not clear in the caption, just say UNCLEAR. If something is not mentioned in original prompt, just say NOT MENTIONED. If they do not match, please point out the errors.

Here is an example:
Input:
Part of origin prompt:
monkey is sitting above a green motorcycle on the left
Captions:
A realistic photo with a monkey sitting above a green motorcycle on the left and another raccoon sitting above a blue motorcycle on the right

Answer:
ERROR[this should be the answer flag, which is ERROR or YES or UNCLEAR or NOT MENTIONED]:
The motorcycle is not green, and it is red.
On the motorcycle there should be a monkey rather than a rabbit.

Input: {prompt}

Figure 15: The system prompt for prompt refinement based on the prompt, image, error analysis, error mapping and revision history, including rate score, feedback, and modified history (A). And B refers to the system prompt used for caption-based error detection.

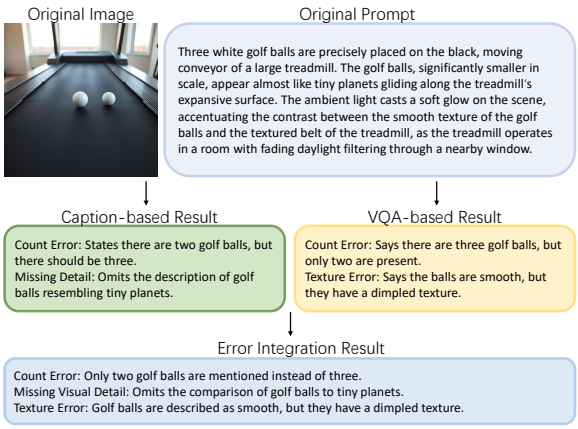


Figure 16: An example that compares the error analysis from the VQA-based method, the caption-based method, and GenPilot. According to the original prompt, the inconsistencies are the number, the texture, and the details of golf balls. VQA-based method misses the details errors and the caption-based method ignores the texture errors. Based on these two analyses, GenPilot is able to perform accurate error analysis.

tic representation of proportions in the generated image.

Action or Pose Errors: Action or Pose Errors occur when the movements or postures of figures or animals in the generated image do not align with the description or logical expectations. The optimized prompt addresses this by incorporating detailed action descriptions and emphasizing dynamic balance. For example, the original prompt resulted in ambiguities related to spatial relationships between body parts, especially in arm and leg positioning, which affected the sense of balance. The optimized prompt utilizes specific descriptions to define spatial relationships and emphasizes precise alignment and balance, thereby enhancing the overall dynamic and harmonic posture in the generated image.

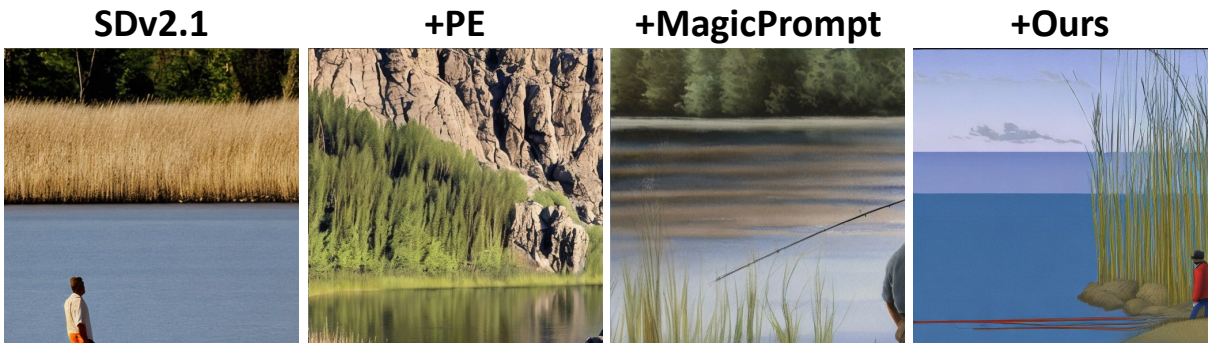
Scene Element Omissions: Scene Element Omissions occur when key components of a scene are missing or underrepresented in the generated image. The optimized prompt solves this by explicitly listing all critical elements required in the scene and reiterating their quantity and spatial relationships. For example, the original prompt mentioned tools and metal racks but failed to highlight their prominence, resulting in a minimalist scene that deviated from the intended complexity. The optimized prompt explicitly lists elements like "tools" and "metal racks" through repetition, ensuring they are visually emphasized and properly positioned, thereby enriching the scene and aligning it with the detailed description provided

Extraneous Scene Elements: Extraneous Scene Elements arise when the generated image includes objects or components not specified in the prompt. The optimized prompt addresses this issue by explicitly excluding unnecessary elements and emphasizing their absence. For example, the original prompt failed to specify the absence of other furniture or objects in the room, leading to the inclusion of unintended elements. The optimized prompt distinctly stated the absence of other elements like benches or lighting in the room, thereby preventing superfluous additions and ensuring the scene remains faithful to the intended description.

Indistinct Background Errors: Indistinct Background Errors mean the background details in the generated image are unclear or underdeveloped. The optimized prompt solves this by explicitly enumerating background elements and emphasizing their characteristics, positions, and spatial relationships relative to the foreground. For example, the original prompt's vague description of the evening sky and surrounding foliage resulted in inconsistent or underdeveloped background details. The optimized prompt added precise descriptions of elements like "large, reflective aviator sunglasses" and the cat's "small, furry face" ensuring these features are clearly generated while also detailing the background to enhance overall image coherence.

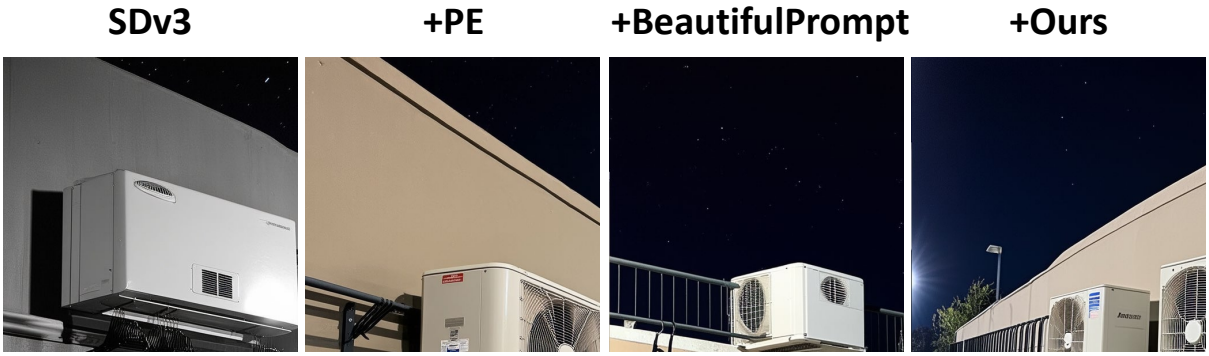
Lighting Errors: Lighting Errors arise when the generated image features lighting that does not align with the intended direction or intensity as described in the prompt. The optimized prompt addresses this by explicitly defining the light source, direction, intensity, color, and its interplay with objects in the scene. For example, the original prompt failed to effectively capture the interaction between light and mist, which is critical for creating a misty atmosphere. The optimized prompt greatly enhances the accuracy of the lighting elements by specifically defining the light's origin, direction, intensity, color, and interaction with mist.

Shadow Errors: Shadow Errors happen when the position and shape of shadows in the generated image don't match the light source and objects. The optimized prompt tackles this issue by clearly specifying the light source, direction, intensity, and the material and shape of objects. For instance, the original prompt's lighting didn't consistently emphasize the bristles or cast long shadows, leading to inaccurate shadow patterns. The optimized prompt highlights the monitor's soft glow as the



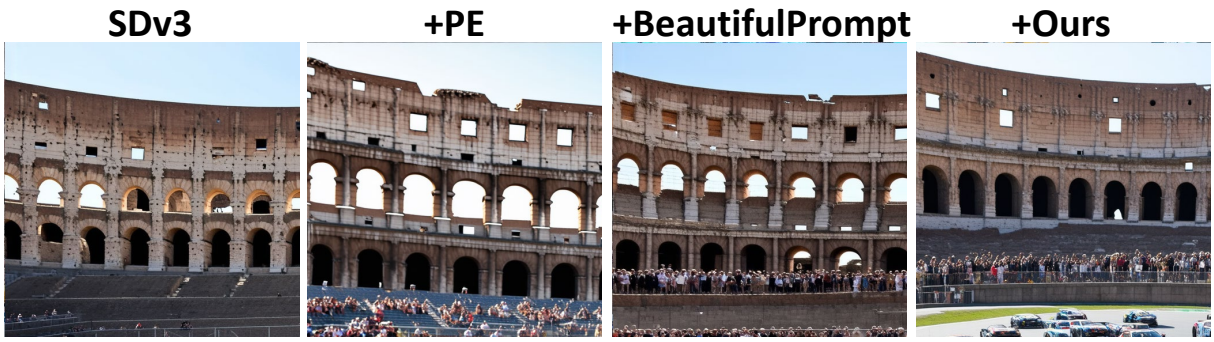
An individual stands at the water's edge, a fishing rod in hand, poised and focused on the task at hand. The bank is lined with reeds and rocks, providing a natural habitat for the fish. In the distance, the gentle flow of the water creates a serene backdrop for this tranquil fishing scene.

In the foreground, two birds with vibrant feathers are perched upon rugged grey rocks that jut out near a tranquil pond with lush green plants at the water's edge. In the midground, a rustic wooden fence creates a boundary line, subtly dividing the natural scene from the world beyond. The background extends into a vast expanse of soft blue sky dotted with tufts of white clouds, stretching far into the horizon.



On a high exterior wall, two large white air conditioning units sit securely bracketed, their vents showing signs of weathering from constant exposure to the elements. Beside them, a rail mounted to the wall supports five sleek black hangers, their long forms casting faint shadows under the faint glow of the nearby street lamp. Above, the dark night sky stretches endlessly, with stars twinkling subtly far in the distance.

Figure 17: The additional examples on DPG-bench challenging dataset with SDv2.1 in the first and second row, and with SD3 on the last row. For comparison, we choose the higher baseline method from BeautifulPrompt and MagicPrompt. The results highlight the superiority of GenPilot in accurately rendering complicated scenes compared to generative models and other enhancement methods.



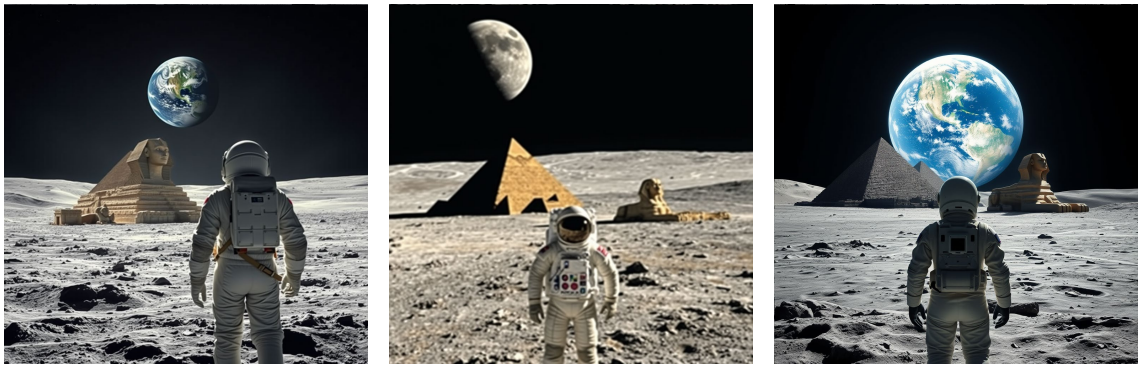
A dynamic scene unfolds at the historic Colosseum, where a fleet of sleek, multicolored racing cars roar past an excited crowd. The vehicles, adorned with vibrant decals and sponsor logos, navigate a temporary circuit that has been meticulously laid out within the ancient arena's interior. Spectators are perched on stone seats that have withstood the test of time, their attention fixed on the blur of machines vying for the lead under the bright afternoon sun.

A rustic wooden table with a natural grain finish, bathed in soft light. On its surface, a cluster of ripe oranges is arranged next to two glass jars filled with a vibrant orange marmalade. The jars catch the light, highlighting the rich color and texture of the contents within.

Three vibrant green lettuce leaves gently float on the surface of crystal-clear water in a shallow white porcelain basin. The sunlight catches the delicate veins of the leaves, highlighting their fresh, crisp texture. Nearby, tiny air bubbles cling to the edges of the leaves and the smooth inner surface of the basin.

Figure 18: More qualitative results on DPG-bench challenging dataset with SD3 in the first row, DALL-E 3 on the second row, and FLUX.1 schnell on the last row. The results clearly demonstrate the significant advantages of our method over the FLUX.1 schnell and the PE method. Specifically, our approach accurately renders key details from the prompt, such as “three”.

FLUX.1schnell **+ PE** **+ Ours**



A surreal lunar landscape unfolds with the iconic Great Pyramids and the Sphinx, all replicated in meticulous detail on the moon's dusty, grey surface. In the foreground, the silhouette of an astronaut, clad in a pearly white spacesuit, is captured from behind, gazing upon the ancient wonders. Above this otherworldly scene, the Earth hangs majestically in the dark expanse of space, its blue and white visage a stark contrast to the barren moonscape.

Two square-shaped pink erasers rest on the tiled floor next to a pristine white porcelain toilet. The erasers feature slight smudges from use and are positioned closely to each other. In the background, the metal toilet flush handle gleams under the bright bathroom light, and a soft blue bath mat lies a short distance away, partially visible in the scene.

An aged and quaint room, lined with crinkled wallpaper, houses a row of four spherical, silver projectors resting on a weathered shelving unit at the rear. These projectors cast bright, focused beams of light toward the room's center, where an expansive antique oak desk sits solemnly. On the desk's polished surface, three electronic keyboards, each with a different design and layout, are neatly arranged, waiting to be played.

Figure 19: Qualitative results for complex and long prompts on DPG-bench challenging dataset compared to PE and FLUX.1 schenell. GenPilot exhibits superior faithfulness to the detailed textual description, for example "from behind" and "metal toilet flush handle" can be accurately generated with the test-time prompt optimization.

Figure 20: The qualitative results on GenEval with FLUX.1 schnell, FLUX.1 schnell with PE, and FLUX.1 schnell with GenPilot. Our system accurately synthesizes the unrealistic image, demonstrating the significant superiority of our method in understanding and generating images.

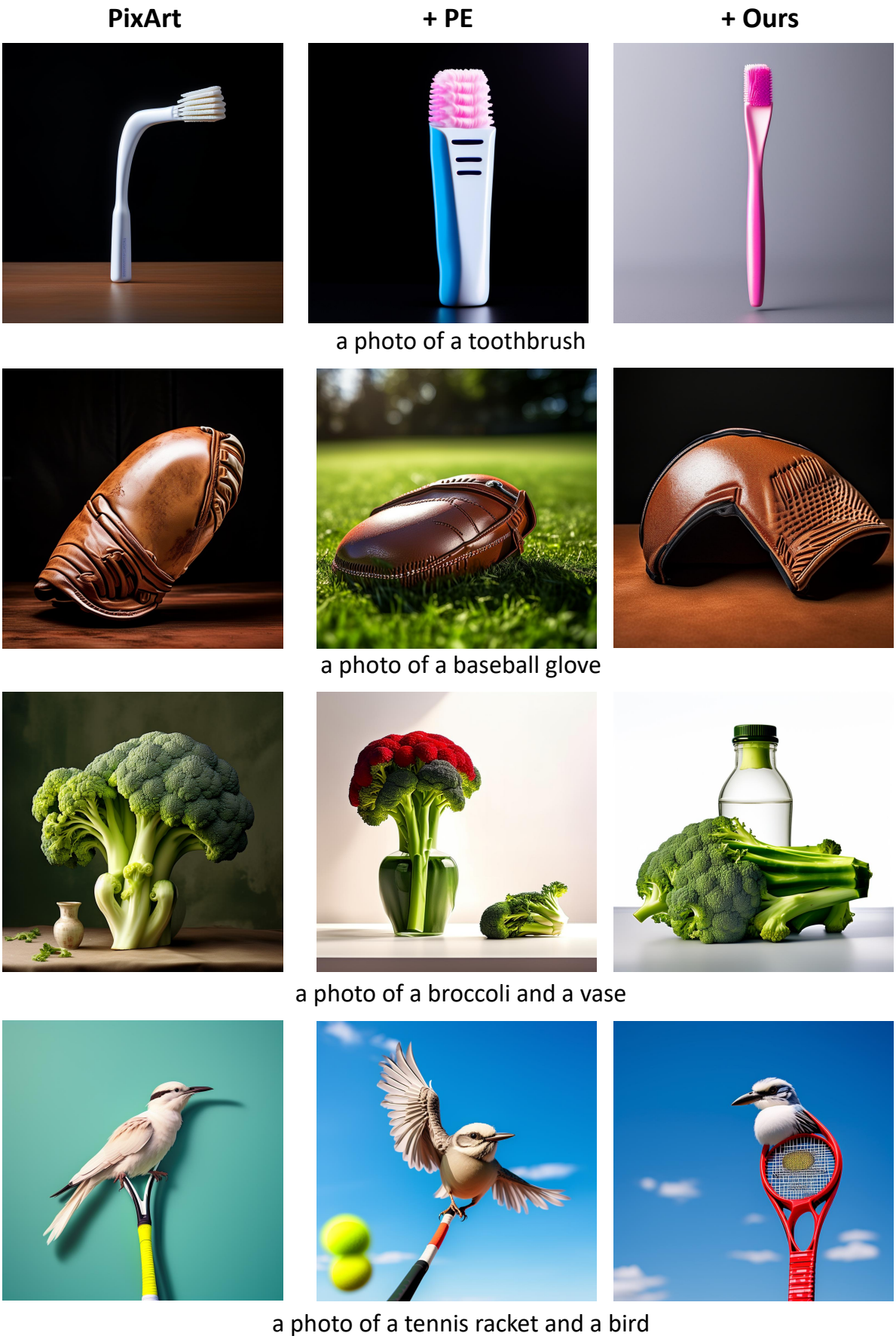


Figure 21: Additional examples on GenEval with PixArt-α and enhancement methods PE and GenPilot. The results reveal the advantages of dealing with the details, such as the baseball glove of GenPilot.

main light source for highlighting the bristles and casting shadows, enhancing shadow depiction accuracy, and ensuring light sources, objects, and their shadows are consistent in the generated image.

Reflection Errors: Reflection Errors occur when the reflection on object surfaces does not comply with physical laws. The optimized prompt addresses this by strengthening the description of the reflection process and detailing the light source's origin, direction, intensity, and the object's material and surface properties. For instance, the original prompt failed to effectively capture the reflection of lighting on the desk's surface due to a lack of emphasis on reflective qualities. The optimized prompt enhances the description of lighting reflection by using phrases like "clearly reflecting the soft glow" thereby improving clarity on reflective surfaces and ensuring the generated image adheres to physical reflection principles.

Object Blurriness: Object Blurriness happens when object outlines and details are unclear in the generated image. The optimized prompt addresses this by emphasizing clear contours and layered details, introducing terms like "sharpness" and "high resolution" while providing multi-level descriptions of local details. For example, the original prompt resulted in a blurred depiction of the anime character's facial features. The optimized prompt emphasizes "ultra-high-definition rendering" and specifies details like "distinct eyelash strands" and "subtle skin pores visible under studio lighting" to ensure clarity in both macro and micro details of the object.

Style Errors: Style Errors arise when the overall style or specific elements in the generated image deviate from the intended aesthetic. The optimized prompt addresses this by introducing stylized keywords and specifying style characteristics such as line thickness, color saturation, and lighting treatment, all while emphasizing stylistic uniformity. For example, the original prompt led to inconsistencies in the steampunk style of the clockwork mechanism. The optimized prompt specifies features like "hyper-detailed brass gears with visible rivets" and "soft Edison bulb illumination" to enforce stylistic coherence across all components, ensuring a unified visual style in the generated image.

Material Errors: Material Errors happen when the generated image inaccurately represents the material properties of objects. The optimized prompt addresses this by explicitly specifying the physical attributes of materials, such as roughness, glossiness, and transparency. For example, the original prompt failed to render the metallic texture of the samurai armor. The optimized prompt uses precise material descriptors like "matte blackened steel with brushed titanium accents" to refine material fidelity, ensuring the generated image accurately reflects the intended texture and finish.

Composition Errors: Composition Errors arise when the layout of the scene in the generated image does not meet the requirements or defies common sense. The optimized prompt resolves this issue by combining composition keywords with quantified object placement and proportion, and by clarifying the hierarchical relationship between the main subject and the background. For example, the original prompt resulted in an unbalanced composition with the main subject positioned at the edge of the frame. The optimized prompt specifies "precise frame composition with the subject centered at the golden ratio point" to achieve harmonic visual balance, ensuring the layout aligns with the intended design principles.

Interaction Errors: Interaction Errors occur when the relationships between objects in the generated image are incorrectly portrayed. The optimized prompt addresses this by using emphasis and contrast to enhance the description of interaction details, ensuring vivid and accurate depictions of how objects interact. For example, the faint trail of damp grass left on the ball as it moves was entirely missing in the original prompt. The optimized prompt includes a clearer depiction of the interaction between the damp grass and the rolling baseball, ensuring the faint trail is distinctly noticeable and the interaction between the two elements is portrayed realistically.

Ambiguous Object States: Ambiguous Object States occur when the condition or status of objects in the generated image is unclear. The optimized prompt addresses this by explicitly defining the specific state of objects, such as motion, power status, or deformation, and incorporating dynamic descriptions. For example, the original prompt led to ambiguity in whether the lamp was on or off. The optimized prompt specifies "the lamp is in an on state with warm light" to clarify its operational status, ensuring the generated image accurately reflects the intended state of the object.

Object Fusion Errors: Object Fusion Errors happen when multiple objects in the generated image are incorrectly merged together. The optimized

prompt addresses this by emphasizing the independence and boundaries of objects, employing clear separation descriptions. For example, the original prompt caused the cat and the dog to merge into a single indistinct shape. The optimized prompt specifies "next to" and enforces "visible fur texture differentiation" to maintain their individual identities, preserving the distinctness of each object in the generated image.

Emphasis Errors: Emphasis Errors occur when elements that should be highlighted in the generated image are not sufficiently emphasized. The optimized prompt addresses this issue by incorporating emphasis keywords such as "highlight" and "emphasize", combined with contrastive descriptions to draw attention to focal points. For example, the original prompt failed to highlight the majestic appearance of the dragon. The optimized prompt emphasizes "the dragon's scales gleaming with iridescent hues" to ensure it stands out as the focal point, thereby enhancing the visual impact and ensuring the intended elements are prominently featured in the generated image.

Atmospheric Mismatch Errors: Atmospheric Mismatch Errors occur when the generated image fails to align with the intended mood or atmosphere described in the prompt. The optimized prompt addresses this by incorporating explicit atmospheric keywords like "mood" and "atmosphere" alongside detailed descriptions of environmental elements such as lighting, color tones, and specific details. For example, the original prompt failed to create the intended mysterious forest atmosphere. The optimized prompt emphasizes "a dark and mysterious atmosphere with fog swirling around ancient tree roots" and specifies "dappled moonlight filtering through dense branches with a cool blue tone" to enhance the intended mood, ensuring the generated image effectively conveys the desired atmosphere.

Cluttered Background Errors: Cluttered Background Errors arise when background elements in the generated image are excessive or disorderly, detracting from the main focus. The optimized prompt addresses this by defining a neutral and clean background and imposing restrictions on background elements. For example, inadequate details in the original prompt led to distracting background elements that interfered with the scene's focus. The optimized prompt defined a neutral and uncluttered background to emphasize the piano and bench, preventing distractions and ensuring the main subjects remain the focal point in the generated image.

Partial Object Generation: Partial Object Generation happens when parts of objects in the generated image are missing. The optimized prompt addresses this by detailing the object's overall structure and each part's specifics, clarifying the connections between parts, and repeatedly emphasizing the object's completeness. For example, the original prompt caused the generation of a bicycle with a missing rear wheel. The optimized prompt specifies the "complete structure of a bicycle with two wheels" and repeatedly emphasizes that "all components, including handlebars, seat, pedals, and both wheels, are fully intact and firmly attached" to ensure no part is omitted in the generated image.

Object Occlusion Errors: Object Occlusion Errors occur when key parts of objects in the generated image are inappropriately blocked by other elements. The optimized prompt addresses this by explicitly defining the spatial hierarchy between objects and emphasizing the visibility of critical items. For example, the original prompt caused the woman's face to be partially obscured by the vase in the foreground. The optimized prompt specifies the "woman positioned in the foreground with a clear, unobstructed view of her face" and adjusts the spatial arrangement by stating "the vase placed behind the woman" ensuring key elements remain visible and the intended focus is maintained in the generated image.

Unwanted Brand Elements: Unwanted Brand Elements occur when brand logos or identifiers appear inappropriately in the generated image. The optimized prompt addresses this by explicitly stating the exclusion of any brand characteristics and emphasizing their absence. In this case, the original image features undesired brand symbols such as "NEFE" on the paintbrush, which were not specified in the original prompt. The optimized prompt explicitly excludes brand names or symbols, leading to cleaner results without unwanted visual elements.

Temporal Ambiguity Errors: Temporal Ambiguity Errors occur when the time setting in the generated image is unclear or inaccurately represented. The optimized prompt addresses this by explicitly specifying the exact time point or time period to eliminate ambiguity. For example, the original prompt inadequately linked the scene to a clear midnight context, creating ambiguity regarding the setting. In the optimized prompt, midnight context details were reinforced with references to the

moon's alignment, object illumination, and atmospheric serenity, ensuring a precise and unambiguous temporal setting in the generated image.

Seasonal Element Errors: Seasonal Element Errors occur when elements related to seasons in the generated image are illogical or inconsistent. The optimized prompt addresses this by explicitly specifying the exact season and detailing natural characteristics, climate conditions, and typical activities associated with that season. For example, the original prompt led to confusion over specific items associated with each season, resulting in misplaced elements like pumpkins in spring and summer images. The optimized prompt explicitly rejects inappropriate additional objects and emphasizes relevant seasonal motifs and colors, ensuring the generated image accurately reflects the intended season.

Facial Expression Errors: Facial Expression Errors occur when the facial expressions of characters in the generated image do not align with the intended emotions described in the prompt. The optimized prompt addresses this by providing detailed descriptions of facial features and utilizing environmental contrasts to highlight the desired expression. For example, the original prompt failed to fully convey the fierce expression of fiery vengeance, particularly in the eyes and mouth area. The optimized prompt intensely highlighted key facial features with flames to enhance the skull's menacing and vengeful expression, ensuring the generated image accurately reflects the intended emotion through detailed facial rendering and environmental emphasis.

Transparency Errors: Transparency Errors occur when the transparency of objects in the generated image is depicted unreasonably. The optimized prompt addresses this by emphasizing the transparent effects of objects and providing details on how light refracts and reflects through them, as well as how other objects are reflected. For example, the astronaut's helmet does not correctly reflect or have transparency showing the lunar landscape. The improved prompt focuses on the helmet's transparency property, ensuring its integration with the lunar landscape.

Background Inconsistency Errors: Background Inconsistency Errors happen when the style and elements of the background in the generated image are not unified. The optimized prompt addresses this issue by emphasizing the need for background consistency and setting an overall style theme. For instance, the original prompt failed to ensure the entire scene, including the background, maintained visual coherence. The optimized prompt establishes "a historical 15th-century European setting" and stresses the importance of keeping the background consistent with this theme, thereby achieving visual harmony in the generated image.

Contrast Errors: Contrast Errors arise when the overall contrast in the generated image is inappropriate, leading to poorly defined distinctions between elements. The optimized prompt resolves this by explicitly specifying the desired contrast and reinforcing related descriptions. For example, the lack of emphasis on visual contrast between apples and leaves resulted in less defined distinctions in the images. The optimized prompt highlights the contrast between the circular leaves and square apples to improve visual discrepancy, ensuring the generated image reflects the intended clarity and distinction through enhanced contrast.

Color Disharmony Errors: Color Disharmony Errors occur when colors in the generated image clash or fail to create a harmonious visual effect. The optimized prompt addresses this by emphasizing the need for overall color coordination and providing a harmonious color scheme. For example, the lack of emphasis on color harmony between elements led to disjointed visual tone representations. By emphasizing color harmony and warm tones, the optimized prompt established better visual and thematic coherence.

Emotional Tone Errors: Emotional Tone Errors occur when the overall image fails to convey the intended emotional tone. The optimized prompt addresses this by elaborately describing the emotional atmosphere and integrating elements like color, lighting, and composition, while emphasizing emotional keywords. For example, the original prompt failed to capture the intended sadness in the scene of a solitary figure by the window. The optimized prompt counters this by specifying "a melancholic atmosphere with soft blue and gray tones," ensuring the generated image aligns with the desired emotional impact through cohesive use of color and lighting.

Object Boundary Errors: Object Boundary Errors occur when the boundaries of objects in the generated image are unclear or incomplete. The optimized prompt addresses this by detailing the object's contours, edge characteristics, and contrast with the background, thereby emphasizing clear boundaries. For instance, the original prompt resulted in a tree whose edges blended ambiguously

with the background, making the boundary unclear. The optimized prompt specifies that the tree should have “crisp, well-defined edges with high contrast against the sky” ensuring the object stands out distinctly in the generated image.