

VISUALPROMPTER: SEMANTIC-AWARE PROMPT OPTIMIZATION WITH VISUAL FEEDBACK FOR TEXT-TO-IMAGE SYNTHESIS

Anonymous authors

Paper under double-blind review

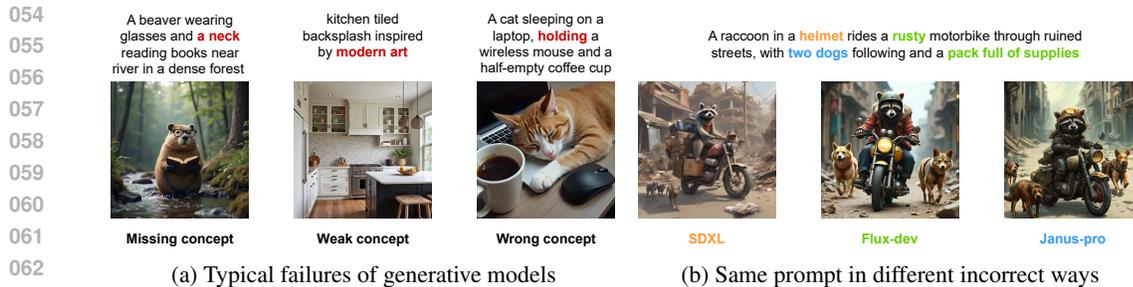
ABSTRACT

The notable gap between user-provided and model-preferred prompts poses a significant challenge for generating high-quality images with text-to-image models, compelling the need for prompt engineering. Current studies on prompt engineering can effectively enhance the style and aesthetics of generated images. However, they often neglect the semantic alignment between generated images and user descriptions, resulting in visually appealing but content-wise unsatisfying outputs. In this work, we propose VisualPrompter, a novel training-free prompt engineering framework that refines user inputs to model-preferred sentences. VisualPrompter utilizes an automatic self-reflection module that identifies absent concepts in the generated images, followed by a target-specific prompt optimization mechanism which revises the prompts in a fine-grained manner. By deconstructing prompts, introducing new elements at the atomic semantic level, and then reassembling them, our model is able to maintain semantic consistency and integrity throughout the optimization process. Extensive experiments demonstrate the effectiveness of VisualPrompter, which achieves new state-of-the-art performance on multiple benchmarks for text-image alignment evaluation. Additionally, our framework features a plug-and-play design, making it highly adaptable to various generative models.

1 INTRODUCTION

Text-to-image (T2I) generation task requires generating realistic images that are consistent with textual descriptions. Recently, diffusion-based generative models (Ho et al., 2020; Rombach et al., 2022) have demonstrated remarkable capabilities in generating vivid images from textual descriptions. However, they still face challenges in correctly representing key concepts within user inputs, as shown in Figure 1a. This issue arises from the significant discrepancy between user input prompts and the prompts preferred by the models Hei et al. (2024). To be specific, novice users often provide brief, coarse-grained descriptions, whereas models tend to perform better with detailed, fine-grained prompts, as such data are commonly used during training. Thus, T2I models may struggle to generate satisfactory results when directly fed with user inputs.

Given that a well-crafted prompt can significantly enhance the quality of generated images (Brown et al., 2020), researchers have explored prompt engineering to automatically refine user input prompts (Hao et al., 2023; Rosenman et al., 2024). For instance, Best Prompts (Pavlichenko & Ustalov, 2023) simply appends a few empirically effective keywords to improve image quality. BeautifulPrompt (Cao et al., 2023) finetune a large language model to serve as a prompt engineer and incorporate reinforcement learning to leverage visual feedback. Despite effectiveness, these methods suffer from three major limitations. First, current studies on T2I prompt engineering (Hao et al., 2023; Rosenman et al., 2024) primarily focused on enhancing the style and aesthetics of generated images, neglecting or even compromising the alignment between texts and images. Second, most works (Liu & Chilton, 2022; Pavlichenko & Ustalov, 2023) apply similar modification to all prompts, lacking fine-grained, case-specific adjustments tailored to individual inputs, resulting in limited performance improvement. Third, existing methods exhibit limited generality (Cao et al., 2023; Rosenman et al., 2024), as they are typically designed for a single specific diffusion model. However, as illustrated in Figure 1b, different models have varying prompt preferences and may



064 Figure 1: Existing T2I generative models often fail to correctly draw key concepts within the pro-
065 vided prompts. Different models also fail differently on the same prompt.

066
067
068 interpret the same prompt in different incorrect ways, making these methods difficult to extend to
069 other models. These limitations highlights the importance of leveraging the outputs of generative
070 models as model-specific feedback for effective optimization.

071 In this paper, we introduce VisualPrompter, a novel and training-free framework that provides fine-
072 grained, case-specific, and model-preferred optimization for distinct user inputs. To make clear the
073 optimization target, our model incorporates a SELF-Reflection module (SERE) to identify the miss-
074 ing concepts that are included in the input prompts but not reflected in the generated images. SERE
075 breaks down the prompts into atomic concepts using a large language model and verifies if each
076 concept is reflected in the generated images using a Visual-Language Model (VLM). Subsequently,
077 with the missing concepts as targets, a Target-Specific Prompt Optimization module (TSPO) is em-
078 ployed to refine the user prompts. By enriching the missing concepts while preserving the existing
079 concepts, TSPO ensures that the refined prompts are both model-preferred and semantically faithful
080 to the user’s original intent. Furthermore, we introduce another LLM to incorporate appropriate
081 aesthetic keywords into sentences, thereby improving the visual quality of optimized images.

082 Notably, by breaking down the user prompts into fine-grained concepts, our method allows case-
083 specific detailed prompt optimization, which is not only more interpretable but also more power-
084 ful. This capability enables the model to assess key attributes like objects, properties, and rela-
085 tions, thereby refining and expanding vague or fragmented user inputs into well-structured, model-
086 preferred prompts. Moreover, our method allows different prompt optimization for different dif-
087 fusion models according to the distinct answers based on their generated images. In this way, our
088 VisualPrompter can be compatible with various generative models, obtaining a universal and robust
089 performance improvement.

090 Experiments show that VisualPrompter outperforms prior prompt engineering works on two text-
091 image alignment evaluation benchmarks, demonstrating the effectiveness of our approach. Images
092 generated from texts optimized by VisualPrompter also achieve higher CLIP scores than those pro-
093 duced by previous prompt engineering methods. Besides this improved semantic fidelity, our opti-
094 mization also enhances the aesthetic quality, contributing to more visually appealing results. Fur-
095 thermore, our model exhibits notable performance improvements across various diffusion models,
096 highlighting its extensive applicability and adaptability.

097 Our contributions are summarized as follows:

- 098
- 099 • We propose an innovative text-to-image prompt engineering method, named Visual-
100 Prompter. Our approach is capable of generating prompts that effectively align with both
101 model preferences and user intent.
 - 102 • We propose a feedback-driven optimization system that leverages the visual feedback from
103 VLM to strategically enhance outputs from diverse generative models.
 - 104 • Our method operates at the atomic semantic level to analyze and refine the prompt, fully
105 preserving its original meaning while incorporating appropriate new content.
 - 106 • Extensive experiments demonstrate the effectiveness of our VisualPrompter, which signifi-
107 cantly outperforms current state-of-the-art prompt engineering methods in multiple bench-
marks.

2 RELATED WORK

2.1 TEXT-TO-IMAGE SYNTHESIS

Diffusion Models (Ho et al., 2020; Song et al., 2021) have recently attained unprecedented success in image synthesis. These models generate images from random noise through a series of denoising steps, achieving high visual fidelity and diversity. The introduction of classifier-free guidance (Ho & Salimans, 2022) facilitates the generation of images from brief text descriptions by leveraging text encoders such as CLIP (Radford et al., 2021), making text-to-image synthesis possible within the diffusion framework. Another significant innovation is the latent diffusion model (LDM) (Rombach et al., 2022), which implements the diffusion process in latent space. LDM is capable of generating images with exceptional visual fidelity, while significantly reducing the required computational resources. Furthermore, the emergence of open-source diffusion models (Esser et al., 2024; Podell et al., 2024) has brought text-to-image generation into the public spotlight. Therefore, a well-designed prompt engineering framework can be immensely beneficial for novice users.

2.2 PROMPT ENGINEERING

Prompt engineering aims to optimize the interaction between humans and AI systems by crafting effective input prompts that elicit desired outputs (Liu et al., 2023; Sahoo et al., 2024). In the field of text-to-image generation, prompt engineering involves carefully selecting and combining phrases to achieve high-quality and satisfactory synthesized images. Early studies (Liu & Chilton, 2022; Pavlichenko & Ustalov, 2023) focus on automatically searching for keywords that significantly influence the style and quality of generated images through mining techniques. With the rapid advancement of large language models, several studies (Cao et al., 2023; Wang et al., 2023a; Rosenman et al., 2024) have explored their utilization as prompt engineers, particularly through fine-tuning and reinforcement learning. These models are capable of producing detailed and varied content prompts, based on short phrases provided by users.

To achieve precise control, other methods (Brade et al., 2023; Feng et al., 2024) focus on the interaction with human. PromptCharm (Wang et al., 2024b) provides convenient attention adjustments for the keywords within prompts. Prompt Expansion (Datta et al., 2024) outputs a set of expanded text prompts and images, providing users with multiple optimization directions for selection. Although these methods can improve the visual quality of generated images, they often neglect the semantic consistency with user-provided descriptions, which significantly limits their practicality in real-world scenarios.

3 VISUALPROMPTER

In this section, we present our VisualPrompter in detail, which is a semantic-aware, plug-and-play prompt engineering method for text-to-image generation task.

3.1 OVERVIEW

The target of our VisualPrompter is to automatically optimize user input prompts into model-preferred prompts. The principal strength of our approach stems from its atomic semantic level prompt optimization, which offers more precise and reliable control over semantics compared to direct expansion by models. Figure 2 illustrates the overall framework of our proposed method.

Given a user input prompt, a SELF-REFlection module (SERE) is employed to identify the specific aspects that require enhancement, which analyzes the image generated by the diffusion model based on the input text. By thoroughly evaluating the semantic information in the prompt through question generation and answering, we can effectively identify the missing concepts. Notably, the missing concepts are those textual descriptions that are included in user inputs but are not reflected in the generated images. To obtain prompts that remain faithful to user inputs while aligning with model preferences, we employ a Target-Specific Prompt Optimization module (TSPO) to refine the user prompt based on the identified missing concepts. TSPO first strategically enriches the attributes of the missing concepts to simulate model-preferred patterns. It then combines these expanded atomic concepts to form a syntactically complete and semantically fluent result.

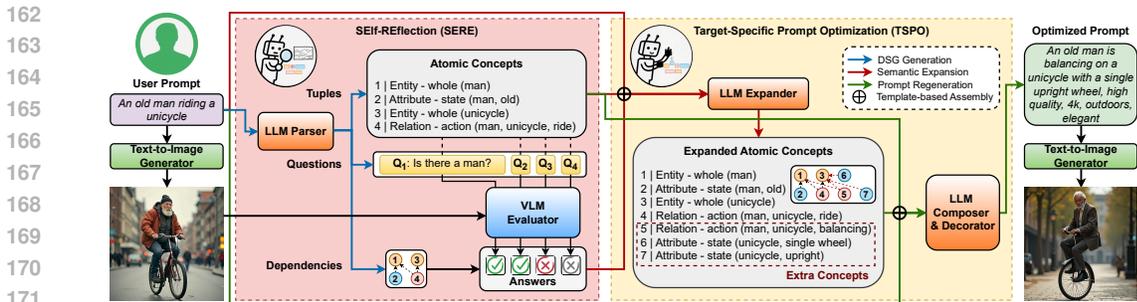


Figure 2: Refining pipeline of our VisualPrompter. The optimization starts with LLM-driven VLM-verifying semantic evaluation, followed by LLM-controlled concept expansion and sentence composition. Our approach emulates the chain-of-thought reasoning in human prompt refinement. All the utilized models do not require any extra training.

3.2 SELF-REFLECTION MODULE

Our self-reflection module utilizes multi-step reasoning to obtain explicit visual feedback by evaluating the generated images against sentence-derived atomic concept questions. This mechanism ensures a fine-grained evaluation of the alignment between textual descriptions and visual content, effectively pinpointing discrepancies where the generative model fails to capture specific aspects of the user input. An example of this fine-grained evaluation is shown in Figure 2.

Question Generation. For detailed question generation, we adopt Davidsonian Scene Graph (DSG) (Cho et al., 2024) to identify atomic concepts within the input prompts. A DSG is a structured representation comprising a set of general questions. Each question corresponding to an atomic concept in the given prompt. These questions are interconnected through directed edges, which capture the entailment dependencies among them. We employ a Large Language Model (LLM) to generate the DSG, and its detailed construction process is elaborated in Appendix A.

Question Answering. Upon completing the construction of DSG, we employ a Visual-Language Mode (VLM) as the visual question answering model to evaluate the generated image against the questions in DSG. The VLM is instructed to provide a binary response for each question, determining if the corresponding concept appears in the generated image. The dependency relationships within the DSG are utilized for question pruning, a process that eliminates the need for redundant assessments. Specifically, if a particular concept is judged to be missing, all subsequent concepts dependent on it are automatically regarded as missing as well. The answers enable a comprehensive and fine-grained semantic analysis of the generated images, which will be employed as semantic-aware information to refine the prompt.

3.3 TARGET-SPECIFIC PROMPT OPTIMIZATION

With those identified missing concepts by our self-reflection module, we can perform targeted optimization of the input texts, reducing unnecessary modifications and expansions. We introduce a novel, training-free optimization approach that effectively manages prompt adjustments.

Prompt Regeneration. Rather than directly rewriting the sentences, VisualPrompter first utilizes an LLM to expand the generated atomic concepts based on the responses to the questions from SERE module. This expansion process specifically targets the identified missing concepts, enriching them with additional detail such as attributes, actions, and spatial relationships. Our observations reveal that augmenting the missing concepts can nudge the user input toward the distribution of model-preferred prompts, providing a plug-and-play lever for optimizing various generative model. These additional elements are also expressed as atomic concepts, ensuring consistency and clarity in representation. Notably, if no missing concepts are detected, no further modifications will be made.

Subsequently, VisualPrompter generates a new prompt from the expanded atomic concepts with LLM. The original prompts and concepts are also provided to guide the synthesis, ensuring that the generated prompt remains contextually aligned with the initial input. TSPO module retains

216 textual descriptions that are already accurately reflected in the generated images to avoid unneces-
217 sary modifications, thereby preserving original user intent. By adopting this two-step optimization
218 methodology, our model is able to produce more accurate and reliable prompts.

219 **Prompt Decoration.** To further enrich the diversity and aesthetic quality of the generated content,
220 we augment the optimized results with several aesthetic keywords that are automatically selected by
221 an LLM. By providing example keywords such as *high-quality*, *detailed*, and *fantasy*, we leverage
222 the LLM to automatically generate rich and diverse keywords that do not conflict with the input
223 sentences. These stylistic keywords are seamlessly integrated into the texts or appended at the end
224 in a comma-separated format.

225 226 4 EXPERIMENTS

227 In this section, we first present involved benchmarks and our experimental configurations. Then, we
228 compare VisualPrompter with three previous SOTA prompt engineering works both quantitatively
229 and qualitatively. Human evaluation is conducted to underscore the preeminence of our approach.
230 Furthermore, we select several widely-used text-to-image generation applications to demonstrate the
231 effectiveness of our model in open-world scenarios.

232 233 4.1 BENCHMARKS AND EVALUATION METRICS

234 **Benchmarks.** For our evaluation, we utilize two comprehensive benchmarks: DSG-1k (Cho et al.,
235 2024) and TIFA v1.0 (Hu et al., 2023). Both of them contain human-written prompts from various
236 established text-to-image benchmarks. The DSG-1k benchmark comprises 1,060 prompts from
237 10 publicly available prompt datasets, encompassing a wide range of skills and writing styles. It
238 provides 8,182 questions associated with these prompts, facilitating a multifaceted evaluation of
239 generated images. Meanwhile, TIFA v1.0 contains 4,081 diverse prompts from 4 datasets. As the
240 original TIFA v1.0 benchmark contains multiple-choice and free-form questions, we further refine
241 these questions and represent them in DSG structure. Ultimately, the TIFA benchmark used in our
242 experiments contains 19,233 general questions.

243 **Metrics.** To evaluate the text-image alignment, we adopt a robust and innovative approach (Yarom
244 et al., 2023; Cho et al., 2023) which leverages question generation and answering techniques
245 (Deutsch et al., 2021; Min et al., 2023). This VLM-as-judge metric demonstrates strong correlations
246 with human judgments (Hu et al., 2023), making it particularly suitable for fine-grained evaluation
247 tasks. We also employ the CLIP Score (Hessel et al., 2021) to evaluate the semantic alignment
248 between textual descriptions and generated images at feature level. For aesthetic assessment, we
249 employ the Aesthetic Score (Schuhmann et al., 2022), a well-established metric for objective image
250 quality assessment, reducing subjectivity in human judgments.

251 252 4.2 EXPERIMENTAL SETTINGS

253 Our VisualPrompter leverages a dual-model architecture that requires an LLM for DSG generation
254 and prompt optimization, connected by a VLM for comprehensive image semantic evaluation. We
255 employ the state-of-the-art open-sourced Qwen2 (Yang et al., 2024) as our primary language model
256 and the Qwen2-VL (Wang et al., 2024a) as our visual question answering model. Each language
257 generation task in our pipeline is initiated by feeding the Qwen2 with a meticulously designed
258 preamble and 23 manually curated samples from TIFA dataset, ensuring consistent model output.

259 To thoroughly evaluate our model’s performance, we conduct extensive testing across three promi-
260 nent diffusion models and one autoregressive generative model, including Stable Diffusion v1.5,
261 Stable Diffusion v2.1 (Rombach et al., 2022), Flux-dev (BlackForestLabs, 2024), and Janus-Pro
262 (Chen et al., 2025). Our evaluation utilizes prompts from both the DSG-1k and TIFA benchmarks.
263 We compare our method with several open-sourced prompt engineering methods, including Neuro-
264 Prompts (Rosenman et al., 2024), Promptist (Hao et al., 2023) and BeautifulPrompt (Cao et al.,
265 2023). All of these methods utilize supervised fine-tuning and reinforcement learning to adjust
266 LLMs for optimizing prompts. Our approach is implemented using PyTorch. All experiments are
267 conducted on a single A100 with 40GB memory.

Table 1: Summarized results on DSG benchmark and TIFA benchmark. Reported scores are based on the percentage of “yes” answers to the questions, while the best scores are highlighted in boldface. Our VisualPrompter demonstrates notable improvement in semantic consistency evaluations.

Methods	DSG benchmark				TIFA benchmark				Average
	SD 1.5	SD 2.1	Flux-dev	Janus-Pro	SD 1.5	SD 2.1	Flux-dev	Janus-Pro	
Baseline	67.5	72.1	79.1	79.5	75.0	80.4	87.9	85.7	78.4
NeuroPrompts	58.4	68.7	75.5	81.7	62.8	75.5	84.0	89.8	74.6
Promptist	65.1	68.7	76.9	78.0	71.4	77.2	85.3	86.9	76.2
BeautifulPrompt	47.9	49.6	51.5	55.3	50.9	53.6	57.4	62.0	53.5
VisualPrompter	69.5	77.0	84.3	82.6	80.7	82.8	93.8	93.7	83.0

Table 2: Semantic evaluation across all categories of the DSG benchmark using Flux-dev.

Model	Category of source texts									
	whoops	localized	posescript	vrđ	countbench	midjourney	tifa160	draw text	stanford	diffusion db
Flux	82.2	85.1	73.3	85.8	72.1	56.7	88.6	83.3	89.1	68.9
NeuroPrompts	75.3	86.1	74.8	84.8	70.3	49.6	85.7	75.6	88.2	58.5
Promptist	79.1	84.1	75.4	81.9	73.3	56.6	85.4	81.2	85.2	61.4
BeautifulPrompt	56.4	53.0	61.2	64.5	44.7	38.6	58.2	40.1	45.7	48.7
VisualPrompter	86.5 (+4.3)	89.7 (+3.6)	75.9 (+0.5)	92.4 (+6.6)	81.7 (+8.4)	65.7 (+9.0)	93.1 (+4.5)	88.1 (+4.8)	92.8 (+3.7)	72.1 (+3.2)

4.3 SEMANTIC ACCURACY EVALUATION

We compare VisualPrompter with several state-of-the-art prompt engineering methods on two semantic consistency evaluation benchmarks, and summarize the results in Table 1. We refine the original text prompts with each method, and use the refined prompts to generate images with different generative models. We then use Qwen2-VL-7B to assess whether the generated images contain the specified semantic information. We quantify the results by measuring the average answering accuracy, which has been demonstrated to be effective by DSG (Cho et al., 2024).

As illustrated in Table 1, our model achieves significant improvements across all evaluation benchmarks and all generative models. Notably, VisualPrompter consistently outperforms the four baseline generative models in terms of average performance scores. The results also show that VisualPrompter achieves higher performance gains when integrated with Flux-dev and Janus-Pro, which can be attributed to their state-of-the-art generative capabilities, particularly in handling long sentences and intricate details. This indicates that VisualPrompter can better collaborate with high-capacity generators to produce results with more accurate semantics. It also demonstrates the strong adaptability of VisualPrompter, enabling its effective application across various scenarios.

Surprisingly, previous methods not only fail to preserve the semantic consistency during optimization but also exhibit varying degrees of decline in this aspect. Our analysis reveals that both NeuroPrompts and Promptist are unable to modify the original descriptions, even they utilize an LLM for prompt optimization. This limitation prevents them from effectively refining those model-dispreferred sentences. Additionally, they often introduce irrelevant keywords that are inconsistent with the overall context of the prompts, thereby compromising the semantic fidelity of the generated images. In contrast, BeautifulPrompt is able to adjust the original prompts and add contextually relevant details. However, it tends to omit critical information during the modification process, which significantly deviates from the user intent.

In Table 2, we present the semantic evaluation results across all categories of the DSG benchmark using Flux-Dev. Additional results from other generative models, as well as those from the TIFA benchmark, can be found in Appendix E. These results confirm that our approach effectively reinforces semantic components in the input prompts and ensures their accurate representation in the generated images. Our model demonstrates consistent improvements across specific categories, indicating its strong capability to comprehend and enhance key elements such as objects, attributes, and relationships.

We visualize the generation results of various generative models, as shown in Figure 3. We also use Qwen2 for zero-shot prompt optimization as a comparison. The results demonstrate that our model is capable of producing more elegant prompts that result in more visually compelling images. It

324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377



Figure 3: Comparison of prompts generated by large language model and different prompt engineering methods. Images in the two rows correspond to outputs from Stable Diffusion v2.1 and Flux-dev, respectively.

successfully corrects the deficiencies of the original images, indicating that our optimized prompts are preferred by generative models. Additional visual examples are presented in Appendix F.

4.4 TEXT-IMAGE RELEVANCE EVALUATION

To further validate the effectiveness of VisualPrompeter, we evaluate the CLIP Score between the optimized prompts and their corresponding generated images. We report the mean scores of all the prompts from the DSG benchmark, generating images with those different generative models. The results are illustrated in Table 3.

As shown in Table 3, prompts optimized by VisualPrompeter achieve the highest scores across all the generative models. The results indicate that the prompts produced by VisualPrompeter can be effectively understood by the generative models, generating images with improved semantic coherence. Optimized prompts from Promptist and BeautifulPrompt also achieve relatively high CLIP Score compared to baselines, whereas NeuroPrompts performs the worst. This is primarily because NeuroPrompts offers only limited keyword options, which significantly restricts the diversity of the resulting prompts. In contrast, we leverage LLMs to generate aesthetic keywords that align well with the sentence content, leading to greater overall coherence and harmony in the prompts.

Additionally, we also notice that Promptist and BeautifulPrompt achieve higher CLIP scores than baseline with Stable Diffusion V1.5 and Janus-Pro, while their performance drops with Stable Diffusion v2.1 and Flux-dev. The performance gap arises from their specific optimization for Stable Diffusion V1.5. Another reason is that Janus-Pro’s autoregressive design enables stronger semantic understanding, allowing it to better handle complex sentences. The discrepancy highlights their limited generalization capability across different generative models. In contrast, our prompt engineering framework functions in a model-agnostic way, which does not rely on any specific generative model, demonstrating the broad compatibility and robustness of VisualPrompeter.

Table 3: Results of CLIP Score between optimized prompts and generated images on DSG benchmark. Baseline represents the results on images from original prompts.

Methods	Generative Models				Mean
	SD v1.5	SD v2.1	Flux	Janus-Pro	
Baseline	31.53	31.88	31.58	31.83	31.71
NeuroPrompts	25.61	25.26	24.56	26.21	25.41
Promptist	32.14	31.61	31.49	31.85	31.77
BeautifulPrompt	31.70	31.38	31.35	32.10	31.63
VisualPrompeter	32.21	32.50	33.81	32.22	32.69

Table 4: Results of Aesthetic Score for generated images on DSG benchmark. Baseline represents the results on images from original prompts.

Methods	Generative Models			Mean
	SD v1.5	SD v2.1	Flux	
Baseline	5.30	5.48	5.77	5.52
NeuroPrompts	6.21	6.03	6.40	6.21
Promptist	5.93	5.86	6.25	6.01
BeautifulPrompt	5.97	5.98	6.48	6.14
VisualPrompter	5.66	5.73	6.03	5.81
Ours + NeuroPrompts	6.13	6.15	6.44	6.24

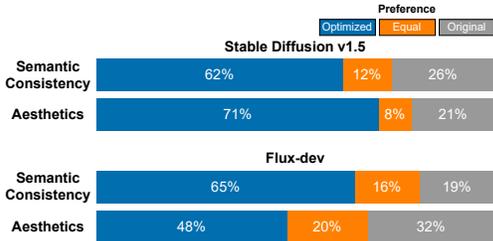


Figure 4: Human evaluation results. From left to right, the segments indicate preference for images from optimized prompts, equal preference for both results, and preference for images from original prompts.

4.5 AESTHETIC EVALUATION

Recent studies (Hao et al., 2023; Podell et al., 2024) have revealed that conventional image quality metrics, such as FID (Heusel et al., 2017) and CLIP Score, do not exhibit a positive correlation with visual aesthetics. To alleviate the burden of labor-intensive human evaluation, the Aesthetic Score (Podell et al., 2024) has emerged as an effective metric, as it is specifically designed to align with human perceptions of beauty, leveraging insights from large-scale datasets. Thus, previous studies have placed greater emphasis on this metric and integrated it into their reinforcement learning frameworks. We also evaluate our method with Aesthetic Score to show the visual elegance of optimized images. The results are summarized in Table 4.

The experimental results on aesthetics show that VisualPrompter achieves modest improvements in aesthetic quality, which is not as substantial as those in previous studies. This discrepancy primarily stems from the tendency of our method to classify the original prompts as descriptions of realistic photographs, which results in an inherent lack of imaginative or fantastical elements among the optimized prompts. This also indicates that our prompt decorator is overly simplistic for enhancing aesthetics. To address this problem, we employ NeuroPrompts as the replacement of our decorator, which adds additional stylistic keywords while preserving the original sentence. This integrated approach achieves optimal performance on average in the aesthetic evaluation. However, we observe a notable decline in semantic consistency, compared to the results without the involvement of NeuroPrompt. It implies that adopting the Aesthetic Score as a training metric may inadvertently encourage the model to optimize for superficial features at the expense of semantic integrity, highlighting the need for more balanced and semantically meaningful training guidance.

4.6 USER STUDY

To enhance the credibility of the above quantitative results, we additionally conduct a human evaluation experiment based on Stable Diffusion v1.5 and Flux-dev. We compute the average preference distribution and report the results in Figure 4. In detail, we generate image pairs using both the original prompts and the optimized prompts derived from VisualPrompter. A panel of 10 independent annotators is tasked with selecting the most desirable images from semantic and aesthetic aspects. Each annotator is provided with 50 samples randomly selected from both benchmarks.

The results of human evaluation indicate that annotators generally prefer images generated by optimized prompts in both semantic and aesthetic aspects, which demonstrates the superiority of our approach. The optimized prompts slightly outperform the original prompts in terms of aesthetic on Flux-dev, owing to its robust generative capability, which can effectively fill in intricate details.

4.7 ABLATION STUDY

Module Impact Evaluation. To systematically analyze the impact of different components in VisualPrompter, we conduct an ablation study to evaluate the individual contributions of the self-reflection module and the target-specific optimization module in Figure 2. Specifically, we compare

Table 5: Impact of different modules in Visual-Prompter. We report the Semantic Accuracy / Aesthetic Score based on the DSG benchmark.

Visual Feedback	Prompt Modification Approach		
	None	Qwen	DSG
w/o	72.1 / 5.48	67.8 / 5.32	71.9 / 5.69
w/	73.8 / 5.54	73.0 / 5.45	77.0 / 5.73

Table 6: Inference time comparison.

T2I Models	Prompt Modification Approach			
	NeuroPrompt	Promptist	Qwen3-32B	Ours
SD v1.5	8.5s	7.0s	21.3s	10.7s
FLUX-dev	20.3s	18.8s	33.1s	34.3s

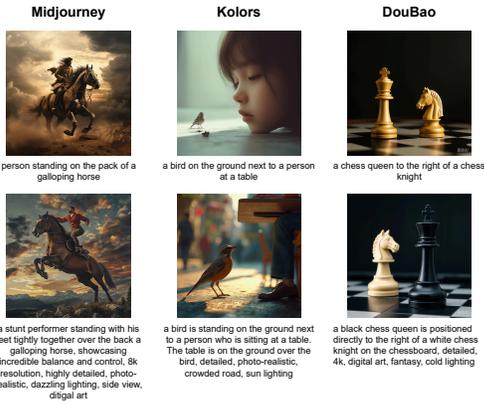


Figure 5: Adaptation for online generators. The second row presents results optimized by VisualPrompter.

two approaches for prompt refinement: direct prompts optimization using Qwen and constructing prompts from expanded concepts based on DSG. For the self-reflection module, visual feedback is leveraged to refine the targeted location of the given prompts. Additionally, we account for the impact of regeneration, where visual feedback is employed to evaluate the original images. Prompts that result in defective images will be used to regenerate images without any modifications.

As shown in Table 5, our fine-grained semantic extraction and composition approach achieves better results than directly using LLM for optimization. We observe that Qwen often produce prompts with semantically irrelevant details. These long sentences increase the difficulty for the diffusion models to comprehend them, leading to worse results than the original prompts. Another observation is that the regeneration of image actually exerts a non-negligible impact on the semantic evaluation metric. Despite this influence, our VisualPrompter achieves superior results, demonstrating its outstanding capability in enhancing semantic consistency.

Inference time. We compare the inference times of several optimization methods and summarized the results in Table 6. We calculate the total time from when a prompt is provided to when a complete image is generated. Our model uses Qwen2 1.5 B for fast DSG generation and concepts integration. As shown in Table 6, while our VisualPrompter is marginally slower compared with existing optimization models, the difference falls within tolerable limits for practical applications.

Online Generation Evaluation. To show the robust adaptability of VisualPrompter, we select several online generative models, including Midjourney (Midjourney, 2021), Kolors (Kolors, 2024) and DouBao (DouBao, 2023). We provide them with human-writing prompts and utilize the feedback to further optimize the prompts. The results are presented in Figure 5. Each image is selected from three samples with the same prompts.

5 CONCLUSION

In this paper, we propose VisualPrompter, a training-free prompt engineering framework that improves text-to-image generation by refining user prompts to model-preferred sentences. We reveal that prompts generated by existing LLM-based methods often struggle to maintain semantic consistency with the original intent and consequently produce images that deviate semantically. To tackle this issue, VisualPrompter operates at the atomic semantic level to analyze and revise the prompt. It leverages model-specific insights obtained through visual reflection to perform targeted expansion and modification of the initial input, thereby ensuring semantic consistency throughout the optimization process. Extensive experiments demonstrate the effectiveness of our VisualPrompter, which achieves state-of-the-art performance in enhancing semantic consistency while maintaining superior aesthetic quality. Compared with previous methods, VisualPrompter exhibits better generalization capabilities, which can be easily adapted to various diffusion models. We intend to extend this feedback-driven optimization system to multimodal generative models in our future work.

REFERENCES

- 486
487
488 BlackForestLabs. Flux. <https://github.com/black-forest-labs/flux>, 2024. Ac-
489 cessed: 2025-03-01.
- 490
491 Stephen Brade, Bryan Wang, Maurício Sousa, Sageev Oore, and Tovi Grossman. Promptify: Text-
492 to-image generation through interactive prompt exploration with large language models. In *Pro-*
493 *ceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, UIST*
494 *2023*, pp. 96:1–96:14. ACM, 2023.
- 495
496 Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhari-
497 wal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal,
498 Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M.
499 Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin,
500 Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford,
501 Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in*
Neural Information Processing Systems 33, NeurIPS 2020, 2020.
- 502
503 Tingfeng Cao, Chengyu Wang, Bingyan Liu, Ziheng Wu, Jinhui Zhu, and Jun Huang. Beautiful-
504 prompt: Towards automatic prompt engineering for text-to-image synthesis. In *Proceedings of*
505 *the 2023 Conference on Empirical Methods in Natural Language Processing: EMNLP 2023*, pp.
1–11. Association for Computational Linguistics, 2023.
- 506
507 Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and
508 Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model
509 scaling. *CoRR*, abs/2501.17811, 2025.
- 510
511 Jaemin Cho, Abhay Zala, and Mohit Bansal. Visual programming for step-by-step text-to-image
512 generation and evaluation. In *Advances in Neural Information Processing Systems, NeurIPS*
2023, 2023.
- 513
514 Jaemin Cho, Yushi Hu, Jason M. Baldridge, Roopal Garg, Peter Anderson, Ranjay Krishna, Mo-
515 hit Bansal, Jordi Pont-Tuset, and Su Wang. Davidsonian scene graph: Improving reliability in
516 fine-grained evaluation for text-to-image generation. In *The Twelfth International Conference on*
Learning Representations, ICLR 2024. OpenReview.net, 2024.
- 517
518 Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep
519 reinforcement learning from human preferences. In *Advances in Neural Information Processing*
520 *Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pp. 4299–4307,
521 2017.
- 522
523 Siddhartha Datta, Alexander Ku, Deepak Ramachandran, and Peter Anderson. Prompt expansion
524 for adaptive text-to-image generation. In *Proceedings of the 62nd Annual Meeting of the Associ-*
525 *ation for Computational Linguistics, ACL 2024*, pp. 3449–3476. Association for Computational
Linguistics, 2024.
- 526
527 Daniel Deutsch, Tania Bedrax-Weiss, and Dan Roth. Towards question-answering as an automatic
528 metric for evaluating the content quality of a summary. *Trans. Assoc. Comput. Linguistics*, 9:
529 774–789, 2021.
- 530
531 DouBao. Doubao. <https://www.doubao.com/chat/create-image>, 2023. Accessed:
2025-03-01.
- 532
533 Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam
534 Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English,
535 and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In
536 *Forty-first International Conference on Machine Learning, ICML 2024*. OpenReview.net, 2024.
- 537
538 Ling Fan, Harry Jiannan Wang, Kunpeng Zhang, Zilong Pei, and Anjun Li. Towards an automatic
539 prompt optimization framework for AI image generation. In *25th International Conference on*
Human-Computer Interaction, HCII 2023, volume 1836 of *Communications in Computer and*
Information Science, pp. 405–410. Springer, 2023.

- 540 Yingchaojie Feng, Xingbo Wang, Kamkwai Wong, Sijia Wang, Yuhong Lu, Minfeng Zhu, Baicheng
541 Wang, and Wei Chen. Promptmagician: Interactive prompt engineering for text-to-image creation.
542 *IEEE Trans. Vis. Comput. Graph.*, 30(1):295–305, 2024.
- 543 Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-
544 shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computa-*
545 *tional Linguistics and the 11th International Joint Conference on Natural Language Processing,*
546 *ACL/IJCNLP 2021*, pp. 3816–3830. Association for Computational Linguistics, 2021.
- 547 Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair,
548 Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural In-*
549 *formation Processing Systems 27: Annual Conference on Neural Information Processing Systems*
550 *2014*, pp. 2672–2680, 2014.
- 551 Ziyu Guo, Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Rui Huang, Haoquan Zhang, Manyuan
552 Zhang, Jiaming Liu, Shanghang Zhang, Peng Gao, Hongsheng Li, and Pheng-Ann Heng. Can we
553 generate images with cot? let’s verify and reinforce image generation step by step, 2025. URL
554 <https://arxiv.org/abs/2501.13926>.
- 555 Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. Optimizing prompts for text-to-image generation.
556 In *Advances in Neural Information Processing Systems 36, NeurIPS 2023*, 2023.
- 557 Nailei Hei, Qianyu Guo, Zihao Wang, Yan Wang, Haofen Wang, and Wenqiang Zhang. A user-
558 friendly framework for generating model-preferred prompts in text-to-image synthesis. In *Thirty-*
559 *Eighth AAAI Conference on Artificial Intelligence, AAAI 2024*, pp. 2139–2147. AAAI Press,
560 2024.
- 561 Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A
562 reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference*
563 *on Empirical Methods in Natural Language Processing, EMNLP 2021*, pp. 7514–7528. Associa-
564 tion for Computational Linguistics, 2021.
- 565 Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter.
566 Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in*
567 *Neural Information Processing Systems 30: Annual Conference on Neural Information Process-*
568 *ing Systems 2017*, pp. 6626–6637, 2017.
- 569 Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022.
- 570 Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances*
571 *in Neural Information Processing Systems 33, NeurIPS 2020*, 2020.
- 572 Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A.
573 Smith. TIFA: accurate and interpretable text-to-image faithfulness evaluation with question an-
574 swering. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023*, pp. 20349–
575 20360. IEEE, 2023.
- 576 Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International*
577 *Conference on Learning Representations, ICLR 2014*, 2014.
- 578 Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-
579 a-pic: An open dataset of user preferences for text-to-image generation. In *Advances in Neural*
580 *Information Processing Systems 36, NeurIPS 2023*, 2023.
- 581 Hyung-Kwon Ko, Gwanmo Park, Hyeon Jeon, Jaemin Jo, Juho Kim, and Jinwook Seo. Large-scale
582 text-to-image generation models for visual artists’ creative works. In *Proceedings of the 28th*
583 *International Conference on Intelligent User Interfaces, IUI 2023*, pp. 919–933. ACM, 2023.
- 584 Kolors. Kolors. <https://app.klingai.com/cn/>, 2024. Accessed: 2025-03-01.
- 585 Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In
586 *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the*
587 *11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021*, pp.
588 4582–4597. Association for Computational Linguistics, 2021.

- 594 Youwei Liang, Junfeng He, Gang Li, Peizhao Li, Arseniy Klimovskiy, Nicholas Carolan, Jiao Sun,
595 Jordi Pont-Tuset, Sarah Young, Feng Yang, Junjie Ke, Krishnamurthy Dj Dvijotham, Kather-
596 ine M. Collins, Yiwen Luo, Yang Li, Kai J. Kohlhoff, Deepak Ramachandran, and Vidhya Naval-
597 pakkam. Rich human feedback for text-to-image generation. In *IEEE/CVF Conference on Com-
598 puter Vision and Pattern Recognition, CVPR 2024*, pp. 19401–19411. IEEE, 2024.
- 599 Zhanhao Liang, Yuhui Yuan, Shuyang Gu, Bohan Chen, Tiankai Hang, Mingxi Cheng, Ji Li, and
600 Liang Zheng. Aesthetic post-training diffusion models from generic preferences with step-by-step
601 preference optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition,
602 CVPR 2025*, pp. 13199–13208. Computer Vision Foundation / IEEE, 2025.
- 604 Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-
605 train, prompt, and predict: A systematic survey of prompting methods in natural language pro-
606 cessing. *ACM Comput. Surv.*, 55(9):195:1–195:35, 2023.
- 607 Vivian Liu and Lydia B. Chilton. Design guidelines for prompt engineering text-to-image generative
608 models. In *CHI '22: CHI Conference on Human Factors in Computing Systems, New Orleans,
609 2022*, pp. 384:1–384:23. ACM, 2022.
- 611 Midjourney. Midjourney. <https://www.midjourney.com/home/>, 2021. Accessed: 2024-
612 12-01.
- 613 Sewon Min, Kalpesh Krishna, Xixi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer,
614 Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of fac-
615 tual precision in long form text generation. In *Proceedings of the 2023 Conference on Empiri-
616 cal Methods in Natural Language Processing, EMNLP 2023*, pp. 12076–12100. Association for
617 Computational Linguistics, 2023.
- 618 Jonas Oppenlaender. A taxonomy of prompt modifiers for text-to-image generation. *Behav. Inf.
619 Technol.*, 43(15):3763–3776, 2024.
- 621 Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin,
622 Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser
623 Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan
624 Leike, and Ryan Lowe. Training language models to follow instructions with human feedback.
625 In *Advances in Neural Information Processing Systems 35, NeurIPS 2022*, 2022.
- 626 Nikita Pavlichenko and Dmitry Ustalov. Best prompts for text-to-image models and how to find
627 them. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Devel-
628 opment in Information Retrieval, SIGIR 2023*, pp. 2067–2071. ACM, 2023.
- 630 Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe
631 Penna, and Robin Rombach. SDXL: improving latent diffusion models for high-resolution image
632 synthesis. In *The Twelfth International Conference on Learning Representations, ICLR 2024*.
633 OpenReview.net, 2024.
- 634 Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agar-
635 wal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya
636 Sutskever. Learning transferable visual models from natural language supervision. In *Proceed-
637 ings of the 38th International Conference on Machine Learning, ICML 2021*. PMLR, 2021.
- 639 Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi
640 Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text
641 transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.
- 642 Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the
643 few-shot paradigm. In *CHI '21: CHI Conference on Human Factors in Computing Systems, 2021*,
644 pp. 314:1–314:7. ACM, 2021.
- 645 Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-
646 resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer
647 Vision and Pattern Recognition, CVPR 2022*, pp. 10674–10685. IEEE, 2022.

- 648 Shachar Rosenman, Vasudev Lal, and Phillip Howard. Neuroprompts: An adaptive framework to
649 optimize prompts for text-to-image generation. In *Proceedings of the 18th Conference of the*
650 *European Chapter of the Association for Computational Linguistics, EACL 2024*, pp. 159–167.
651 Association for Computational Linguistics, 2024.
- 652 Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed
653 Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans,
654 Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion
655 models with deep language understanding. In *Advances in Neural Information Processing Sys-*
656 *tems 35, NeurIPS 2022, 2022*.
- 657 Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha.
658 A systematic survey of prompt engineering in large language models: Techniques and applica-
659 tions, 2024.
- 660 Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi
661 Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski,
662 Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jit-
663 sev. LAION-5B: an open large-scale dataset for training next generation image-text models. In
664 *Advances in Neural Information Processing Systems 35, NeurIPS 2022, 2022*.
- 665 Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th*
666 *International Conference on Learning Representations, ICLR 2021*. OpenReview.net, 2021.
- 667 Fu-Yun Wang, Yunhao Shui, Jingtian Piao, Keqiang Sun, and Hongsheng Li. Diffusion-npo: Nega-
668 tive preference optimization for better preference aligned generation of diffusion models. In *The*
669 *Thirteenth International Conference on Learning Representations, ICLR 2025*. OpenReview.net,
670 2025.
- 671 Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu,
672 Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng
673 Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model’s
674 perception of the world at any resolution, 2024a.
- 675 Yunlong Wang, Shuyuan Shen, and Brian Y. Lim. Reprompt: Automatic prompt editing to refine
676 ai-generative art towards precise expressions. In *Proceedings of the 2023 CHI Conference on*
677 *Human Factors in Computing Systems, CHI 2023*, pp. 22:1–22:29. ACM, 2023a.
- 678 Zhijie Wang, Yuheng Huang, Da Song, Lei Ma, and Tianyi Zhang. Promptcharm: Text-to-image
679 generation through multi-modal prompting and refinement. In *Proceedings of the CHI Conference*
680 *on Human Factors in Computing Systems, CHI 2024*, pp. 185:1–185:21. ACM, 2024b.
- 681 Zijie J. Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and
682 Duen Horng Chau. Diffusiondb: A large-scale prompt gallery dataset for text-to-image gener-
683 ative models. In *Proceedings of the 61st Annual Meeting of the Association for Computational*
684 *Linguistics, ACL 2023*, pp. 893–911. Association for Computational Linguistics, 2023b.
- 685 Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein.
686 Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery.
687 In *Advances in Neural Information Processing Systems 36, NeurIPS 2023, 2023*.
- 688 Tsung-Han Wu, Long Lian, Joseph E. Gonzalez, Boyi Li, and Trevor Darrell. Self-correcting llm-
689 controlled diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recogni-*
690 *tion, CVPR 2024*, pp. 6327–6336. IEEE, 2024.
- 691 Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better
692 aligning text-to-image models with human preference. In *IEEE/CVF International Conference on*
693 *Computer Vision, ICCV 2023*, pp. 2096–2105. IEEE, 2023.
- 694 An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li,
695 Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang,
696 Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren
697 Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang,
698

Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024.

Michal Yarom, Yonatan Bitton, Soravit Changpinyo, Roei Aharoni, Jonathan Herzig, Oran Lang, Eran Ofek, and Idan Szepktor. What you see is what you read? improving text-image alignment evaluation. In *Advances in Neural Information Processing Systems 36, NeurIPS 2023*, 2023.

Shih-Ying Yeh, Sang-Hyun Park, Giyeong Oh, Min Song, and Youngjae Yu. TIPO: text to image with text presampling for prompt optimization, 2024. URL <https://arxiv.org/abs/2411.08127>.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 3813–3824. IEEE, 2023.

Weixia Zhang, Kede Ma, Jia Yan, Dexiang Deng, and Zhou Wang. Blind image quality assessment using A deep bilinear convolutional neural network, 2019.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models, 2021.

A AUTOMATIC CONSTRUCTION OF DSG

Davidsonian Scene Graph (DSG) (Cho et al., 2024) is a set of questions in the form of directed acyclic graph, which has been used for text-image alignment evaluation. As shown in the right of Figure 6, each question in DSG is about an atomic concept, which represents a specific and indivisible semantic unit through a tuple. By feeding the generated images and the related questions into a visual question answering model, we can effectively identify and analyze the missing concepts present in the images.

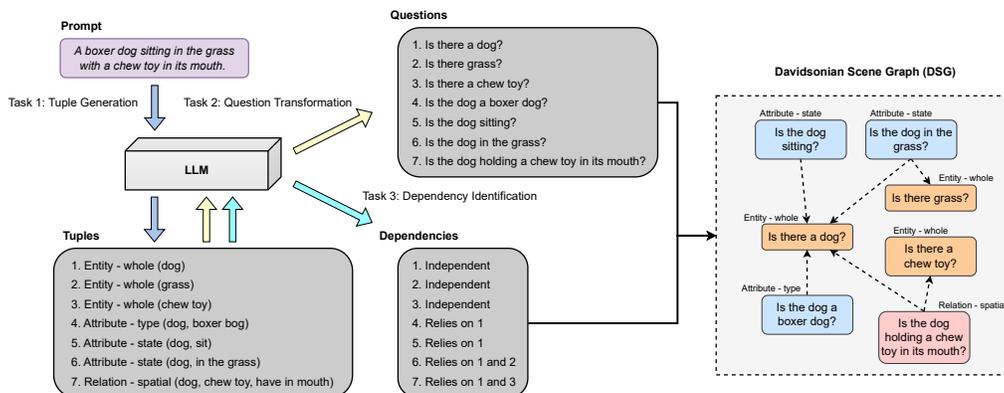


Figure 6: The automatic construction of DSG with an LLM.

Large language models (LLMs) have been employed to construct DSG from textual descriptions automatically. To meticulously construct a DSG that is both comprehensive and non-redundant, our approach adopts a three-step pipeline for the overall generation process, as illustrated in Figure 6. Specifically, an input prompt is first decomposed into detailed concepts represented as tuples, which include entities, attributes, and actions in the text. Based on these tuples of atomic concepts, LLMs are able to generate corresponding general interrogative sentences and determine the dependency relationship among them. Finally, the questions and dependencies are combined to form the DSG for further use.

756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774

Preamble

Task: I will give several examples that contain the input and output. Then I will give a prompt, please generate skill-specific tuples from the prompt according to the specified task requirements and provided examples. The last element of a tuple should be enclosed in parentheses.

First, extract all entities from the prompt and describe them using tuples that start with "entity". If the entity represents a part of another entity, use "part" as the second element of the tuple; otherwise, use "whole". And the last element of the tuple is the name of the entity. For entity that is a part of a whole one, the name should include its main entity.

Second, extract all the descriptions of styles and describe them using tuples that start with "other". Descriptions with specific quantities should also be recorded.

Third, extract all attributes of the entities such as color, material, shape and size. Descriptive phrase of appearance that contain verbs is also treated as an attribute. However, descriptions of location is not considered in this task. Describe them using tuples that start with "attribute". For the second element of the tuple, specify the category of the attribute. And for the final element, provide the entity and its attribute.

Last, extract the spatial and action relationships between entities and describe them using tuples or tuples that start with "relation". For spatial relationship, use "spatial" as the second element, followed by a third element that details the two entities and their spatial relation. For action-based relationship, use a tuple to describe it. And the last element describes two entities and the action relationship.

Only give me those tuples without the prompt. Do not generate tuples describing the same thing among all the tasks. Do not generate tuples that are not explicitly described in the prompt. output format: id | tuple

Examples

input:
PROMPT
A person wearing sunglasses is riding a bike.
output:
1 | entity - whole - (person)
2 | entity - part - (person's sunglasses)
3 | entity - whole - (bike)
4 | attribute - state - (person, wearing sunglasses)
5 | relation - action - (person, bike, riding)
..... (another 22 samples)

Test Input

input:
PROMPT
An infant is sitting on a bed wearing glasses and reading a book.
output:
1 | entity - whole - (infant)
2 | entity - whole - (bed)
3 | entity - part - (glasses)
4 | entity - whole - (book)
5 | attribute - state - (infant, sitting)
6 | attribute - state - (infant, wearing glasses)
7 | relation - spatial - (infant, bed, on)
8 | relation - action - (infant, book, reading)

Output

1 | entity - whole - (infant)
2 | entity - whole - (bed)
3 | entity - part - (glasses)
4 | entity - whole - (book)
5 | attribute - state - (infant, sitting)
6 | attribute - state - (infant, wearing glasses)
7 | relation - spatial - (infant, bed, on)
8 | relation - action - (infant, book, reading)

Figure 7: Example of tuple generation.

775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803

Preamble

Task: I will give several examples that contain the input and output. Then I will give a prompt, skill-specific tuples and tuples after supplementation. Please provide me with a complete, natural sentence based on the given prompt, which incorporates the content of the new tuples.

Here are some rules for these tuples.

Tuples that start with "entity" describe the entity in the given prompt. If the entity represents a part of another entity, use "part" as the second element of the tuple; otherwise, use "whole". And the last element of the tuple is the name of the entity. For entity that is a part of a whole one, the name should include its main entity.

Tuples that start with "other" describe the total style of the image. Descriptions with specific quantities should also be recorded.

Tuples that start with "attribute" describe the attribute of the entity in them. For the second element of the tuple, specify the category of the attribute. And for the final element, provide the entity and its attribute.

Tuples that start with "relation" or "action" describe the relation and action of the entities. For spatial relationship, use "spatial" as the second element, followed by a third element that details the two entities and their spatial relation. For action-based relationship, use a tuple to describe it. And the last element describes two entities and the action relationship.

You should only give me the rewritten text. Do not add any irrelevant content that is not in the new tuples.

Examples

input:
PROMPT
A red motorcycle parked by paint chipped doors.
TUPLES
1 | entity - whole (motorcycle)
2 | entity - whole (doors)
3 | attribute - color (motorcycle, red)
4 | attribute - state (door, paint chipped)
5 | relation - spatial (motorcycle, door, next to)
6 | attribute - state (motorcycle, parked)
REWRITTEN_TUPLES
1 | entity - whole (motorcycle)
2 | entity - whole (doors)
3 | attribute - color (motorcycle, red)
4 | attribute - state (doors, paint chipped)
5 | relation - spatial (motorcycle, doors, next to)
6 | attribute - state (motorcycle, parked)
7 | attribute - state (motorcycle, shiny)
8 | other - count (doors, ==2)
9 | attribute - color (wall, white)
10 | entity - whole (wall)
11 | attribute - color (wall, white)
12 | relation - spatial (doors, wall, attached to)
output:
A red and shiny motorcycle is parked next to two old, paint-chipped doors that are attached to a white wall.
..... (another 22 samples)

Test Input

input:
PROMPT
A man riding his bike of rock cliffs.
TUPLES
1 | entity - whole (man)
2 | entity - whole (bike)
3 | entity - whole (rock cliffs)
4 | action - (man, bike, ride)
5 | relation - spatial (man, rock cliffs, on)
REWRITTEN_TUPLES
1 | entity - whole (man)
2 | entity - whole (bike)
3 | entity - whole (rock cliffs)
4 | action - (man, bike, ride)
5 | relation - spatial (man, rock cliffs, on)
6 | attribute - state (man, riding)
7 | attribute - state (bike, moving)
8 | attribute - state (rock cliffs, rugged)
9 | attribute - state (man, helmet, wearing)
10 | relation - spatial (bike, man, on)
output:
A man is riding his bike on rugged rock cliffs. He is wearing a helmet, and the bike is moving on the cliffs.

Output

A man is riding his bike on rugged rock cliffs. He is wearing a helmet, and the bike is moving on the cliffs.

Figure 8: Example of prompt regeneration.

804
805
806
807
808
809

Preamble

Task: I will give several examples that contain the input and output. And I will give a prompt, skill-specific tuples and a yes or no list, please add more tuples with additional information which do no conflict with the previous information.

First, rewrite all the given tuples. Here are some rules for these tuples.

Tuples that start with "entity" describe the entity in the given prompt. If the entity represents a part of another entity, use "part" as the second element of the tuple; otherwise, use "whole". And the last element of the tuple is the name of the entity. For entity that is a part of a whole one, the name should include its main entity.

Tuples that start with "other" describe the total style of the image. Descriptions with specific quantities should also be recorded.

Tuples that start with "attribute" describe the attribute of the entity in them. For the second element of the tuple, specify the category of the attribute. And for the final element, provide the entity and its attribute.

Tuples that start with "relation" or "action" describe the relation and action of the entities. For spatial relationship, use "spatial" as the second element, followed by a third element that details the two entities and their spatial relation. For action-based relationship, use a tuple to describe it. And the last element describes two entities and the action relationship.

Second, if the list specifies "no", expand the corresponding tuple with the same id. Generate several additional detailed descriptions such as objects, attributes and actions and write them after the tuples with continuing sequentially ids. You can create content that is not present in the given prompt.

Each tuple is preceded by a numerical id. If there is no false answer in the answer list, just give back the input tuples is ok. You should only give me the tuples. Do not give explanations. output format: id | tuple

Examples

input:
PROMPT
A red motorcycle parked by paint chipped doors.
TUPLES
1 | entity - whole (motorcycle)
2 | entity - whole (doors)
3 | attribute - color (motorcycle, red)
4 | attribute - state (door, paint chipped)
5 | relation - spatial (motorcycle, door, next to)
6 | attribute - state (motorcycle, parked)
ANSWERS
1 | yes
2 | no
3 | yes
4 | no
5 | no
6 | yes
output:
1 | entity - whole (motorcycle)
2 | entity - whole (doors)
3 | attribute - color (motorcycle, red)
4 | attribute - state (doors, paint chipped)
5 | relation - spatial (motorcycle, doors, next to)
6 | attribute - state (motorcycle, parked)
7 | attribute - state (doors, old)
8 | attribute - state (doors, old)
9 | other - count (doors, ==2)
10 | entity - whole (wall)
11 | attribute - color (wall, white)
12 | relation - spatial (doors, wall, around)
..... (another 22 samples)

Test Input

input:
PROMPT
Three security guards wearing black uniform looking through binoculars, in three different directions, shot on white
TUPLES
1 | entity - whole (security guards)
2 | other - count (security guards, ==3)
3 | entity - part (security guards' uniform)
4 | attribute - color (security guards' uniform, black)
5 | entity - part (security guards' binoculars)
6 | relation - action (security guards, binoculars, look through)
7 | attribute - color (security guards, in three different directions)
ANSWERS
1 | yes
2 | no
3 | yes
4 | yes
5 | yes
6 | no
7 | yes
output:
1 | entity - whole (security guards)
2 | other - count (security guards, ==3)
3 | entity - part (security guards' uniform)
4 | attribute - color (security guards' uniform, black)
5 | entity - part (security guards' binoculars)
6 | relation - action (security guards, binoculars, look through)
7 | attribute - state (security guards, in three different directions)
8 | relation - action (security guard 1, binoculars, look north)
9 | relation - action (security guard 2, binoculars, look east)
10 | relation - action (security guard 3, binoculars, look south)
11 | attribute - state (security guards, standing)
12 | entity - whole (background)
13 | attribute - color (background, white)

Output

1 | entity - whole (security guards)
2 | other - count (security guards, ==3)
3 | entity - part (security guards' uniform)
4 | attribute - color (security guards' uniform, black)
5 | entity - part (security guards' binoculars)
6 | relation - action (security guards, binoculars, look through)
7 | attribute - state (security guards, in three different directions)
8 | relation - action (security guard 1, binoculars, look north)
9 | relation - action (security guard 2, binoculars, look east)
10 | relation - action (security guard 3, binoculars, look south)
11 | attribute - state (security guards, standing)
12 | entity - whole (background)
13 | attribute - color (background, white)

Figure 9: Example of concepts expansion.

To ensure consistent and standardized outputs from the large language model (LLM), for each step in the process, we provide the LLM with a well-designed preamble and multiple manually annotated samples. This methodology guides the LLM to generate tuple annotations in a uniform format for

Table 7: Class and examples of aesthetic keywords.

class	Samples
quality	best quality, 4k, 8k, highres, masterpiece, fantasy art, highly detailed
style	photo-realistic, oil painting, portraits, digital art, landscape, impressionist, anime, concept artists, cyberpunk
background	blue sky, crowded road, beautifully decorated wall, high mountain
light	studio lighting, soft lighting, sun lighting, diffuse lighting, cold lighting
aesthetics	beautiful, elegant, stunning, lovely, majestic, charming, graceful

new inputs, thereby enhancing the reliability and scalability of DSG generation across diverse textual inputs. We assign a numerical identifier to each tuple to distinguish between them and ensure the integrity of data. An example of tuple generation is shown in Figure 7. The preambles and annotated examples for other tasks will be included in our code release.

For the generation of questions and dependencies, we provide the LLM with both the prompts and tuples to enhance comprehension. Any invalid results will be regenerated multiple times to ensure accuracy.

B TARGET-SPECIFIC OPTIMIZATION

Here we provide examples for the two key technologies in our target-specific optimization module: concepts expansion and prompt regeneration. We utilize the LLM engineering method used in DSG generation above. Examples of prompt templates are shown in Figure 8 and Figure 9.

C KEYWORDS FOR DECORATION

To enhance the aesthetics of generated images, we employ an LLM decorator to incorporate keywords which have been demonstrated to substantially improve the aesthetic appeal of the images. To constrain the generation scope of the LLM, we provide several keyword examples in the task preamble and instruct the LLM decorator to select appropriate keywords that align with the prompt content. These suitable keywords are appended at the end of the input sentence, separated by commas. We collect some keywords from the results of NeuroPrompts and BeautifulPrompt. These keywords are systematically organized according to their descriptive categories, including image quality, artistic style, background description, environmental illumination and perspective. A selection of examples is illustrated in Table 7.

For each class of keywords, we guide the LLM to select or create 0 to 2 words and incorporate them into the input prompts. This approach yields sentences that are both detailed and enriched with aesthetic keywords. Our experimental findings indicate that the inclusion of these aesthetic keywords does not substantially affect the semantic consistency between generated images and textual descriptions.

D EVALUATION RELIABILITY WITH DSG

We adopt the Davidsonian Scene Graph (DSG) (Cho et al., 2024) as a stable metric for text-image alignment. DSG decomposes prompts into atomic, dependency-structured questions whose accuracy yields a precise measurement of semantic consistency. The alignment score is computed as the proportion of "yes" responses from a VLM, ensuring that images with stronger textual fidelity receive higher scores. Notably, DSG demonstrates near-perfect reliability: questions achieve 98.3% precision and 96.0% recall against human annotations, while state-of-the-art VLMs answer these questions with 82.6% accuracy in entity judgments. These results substantiate the reliability of DSG, enabling our VisualPrompter to deliver dependable defect detection and targeted optimization. Furthermore, DSG's dependency-aware design inherently prunes invalid queries such as asking about an attribute when the object is absent, thereby reducing noise and reinforcing measurement stability across diverse prompts and models. For additional experiments and implementation details validating DSG's stability, please refer to Cho et al. (2024).

Table 8: Detailed results on TIFA benchmark. Reported scores represent the percentage of “yes” answers.

Model	Category of source texts				Average
	coco	drawbench	partiprompt	paintskill	
SD v1.5	81.8	65.2	72.5	58.5	75.0
NeuroPrompts	69.7	49.5	58.2	52.4	62.8
Promptist	78.1	62.4	67.9	57.8	71.4
BeautifulPrompt	56.3	50.8	45.9	43.1	50.9
VisualPrompter	87.5 (+5.7)	70.5 (+5.3)	77.0 (+4.5)	66.8 (+8.3)	80.7 (+5.7)

(a) Semantic evaluation on Stable Diffusion v1.5.

Model	Category of source texts				Average
	coco	drawbench	partiprompt	paintskill	
Flux	92.4	80.8	86.2	77.2	87.9
NeuroPrompts	88.6	74.2	80.3	79.2	84.0
Promptist	90.0	75.6	82.3	78.2	85.3
BeautifulPrompt	63.2	56.1	49.8	56.3	57.4
VisualPrompter	96.2 (+3.8)	87.6 (+6.8)	90.8 (+4.6)	94.7 (+5.5)	93.8 (+5.9)

(c) Semantic evaluation on Flux-dev.

Model	Category of source texts				Average
	coco	drawbench	partiprompt	paintskill	
SD v2.1	86.2	69.1	77.9	67.5	80.4
NeuroPrompts	81.4	64.5	72.5	64.3	75.5
Promptist	82.3	67.1	74.8	67.3	77.2
BeautifulPrompt	58.9	50.3	48.9	46.8	53.6
VisualPrompter	89.0 (+2.8)	74.5 (+5.4)	79.6 (+1.7)	69.7 (+2.2)	82.8 (+2.4)

(b) Semantic evaluation on Stable Diffusion v2.1.

Model	Category of source texts				Average
	coco	drawbench	partiprompt	paintskill	
Janus-Pro	90.8	85.4	80.2	81.1	85.7
NeuroPrompts	92.9	85.2	87.9	83.9	89.7
Promptist	90.4	79.4	84.8	81.1	86.9
BeautifulPrompt	67.1	60.3	56.6	57.4	62.0
VisualPrompter	95.4 (+2.5)	88.2 (+2.8)	91.9 (+4.0)	93.8 (+9.9)	93.7 (+4.0)

(d) Semantic evaluation on Janus-Pro.

Table 9: Detailed results on DSG benchmark. The best scores are highlighted in boldface. Our VisualPrompter stands out as the only method that demonstrates notable improvement in semantic consistency evaluation.

Model	whoops	localized	posescript	vrd	Category of source texts					Average	
					countbench	midjourney	tifa160	draw text	stanford		diffusion db
SD v1.5	69.8	70.9	65.9	66.7	68.0	61.5	77.1	55.2	65.6	68.6	67.5
NeuroPrompts	57.8	62.9	58.3	62.2	54.1	50.3	64.6	47.6	62.0	60.6	58.4
Promptist	73.7	68.0	61.8	64.7	66.2	57.9	70.5	56.1	65.1	63.3	65.1
BeautifulPrompt	48.0	48.3	52.8	53.7	43.7	46.3	50.1	41.5	38.9	54.5	47.9
VisualPrompter	76.5 (+2.8)	75.7 (+4.8)	61.5 (-4.4)	71.4 (+4.7)	72.3 (+4.3)	63.4 (+1.9)	77.8 (+0.7)	54.5 (-1.6)	63.6 (-2.0)	73.2 (+4.6)	69.5 (+2.0)

(a) Semantic evaluation on Stable Diffusion v1.5.

Model	whoops	localized	posescript	vrd	Category of source texts					Average	
					countbench	midjourney	tifa160	draw text	stanford		diffusion db
SD v2.1	76.9	77.7	66.9	70.6	69.5	63.0	81.6	64.1	73.3	71.6	72.1
NeuroPrompts	70.8	76.4	62.2	74.5	66.4	58.7	74.5	63.8	66.1	70.1	68.7
Promptist	71.5	73.3	65.8	68.1	71.2	59.8	75.9	67.1	64.6	65.7	68.7
BeautifulPrompt	54.7	47.3	55.2	56.4	47.1	44.6	52.4	41.3	43.4	52.1	49.6
VisualPrompter	82.0 (+5.1)	80.6 (+2.9)	71.1 (+4.2)	77.5 (+3.0)	79.1 (+9.6)	69.6 (+6.6)	83.8 (+2.2)	73.3 (+6.2)	74.2 (+0.9)	74.9 (+3.3)	77.0 (+4.9)

(b) Semantic evaluation on Stable Diffusion v2.1.

Model	whoops	localized	posescript	vrd	Category of source texts					Average	
					countbench	midjourney	tifa160	draw text	stanford		diffusion db
Janus-Pro	86.1	87.2	69.8	82.4	73.7	65.6	87.2	79.9	85.9	72.1	79.5
NeuroPrompts	86.2	88.7	76.7	86.4	76.0	68.3	88.1	85.9	86.8	70.0	81.7
Promptist	84.6	80.4	68.1	80.3	76.2	65.6	88.3	78.9	82.5	69.2	78.0
BeautifulPrompt	66.3	54.2	59.1	62.4	54.7	45.2	61.0	45.5	44.3	56.8	55.3
VisualPrompter	91.2 (+5.0)	86.6 (-2.1)	74.2 (-2.5)	86.9 (+0.5)	79.0 (+2.8)	69.8 (+1.5)	90.8 (+2.5)	85.8 (-0.1)	80.1 (-6.7)	76.9 (+4.8)	82.6 (+0.9)

(c) Semantic evaluation on Janus-Pro.

E DETAILED RESULTS ON BENCHMARKS

The benchmarks used in our study, namely TIFA v1.0 (Hu et al., 2023) and DSG (Cho et al., 2024) (as introduced in the paper), consist of multiple constituent datasets. We have systematically evaluated VisualPrompter’s performance across all sub-datasets and compiled the comprehensive results in Table 8 and Table 9.

From Table 9, our model demonstrates more pronounced improvements across prompts from *whoops*, *localized* and *countbench* datasets, which care more about object attributes and spatial relationships. The results demonstrate that our model is capable of effectively enhancing semantic elements in the prompt, ensuring their representation in the generated images. We also notice that there are still 3 sub-datasets in the detailed tables on which our method exhibits a slight performance

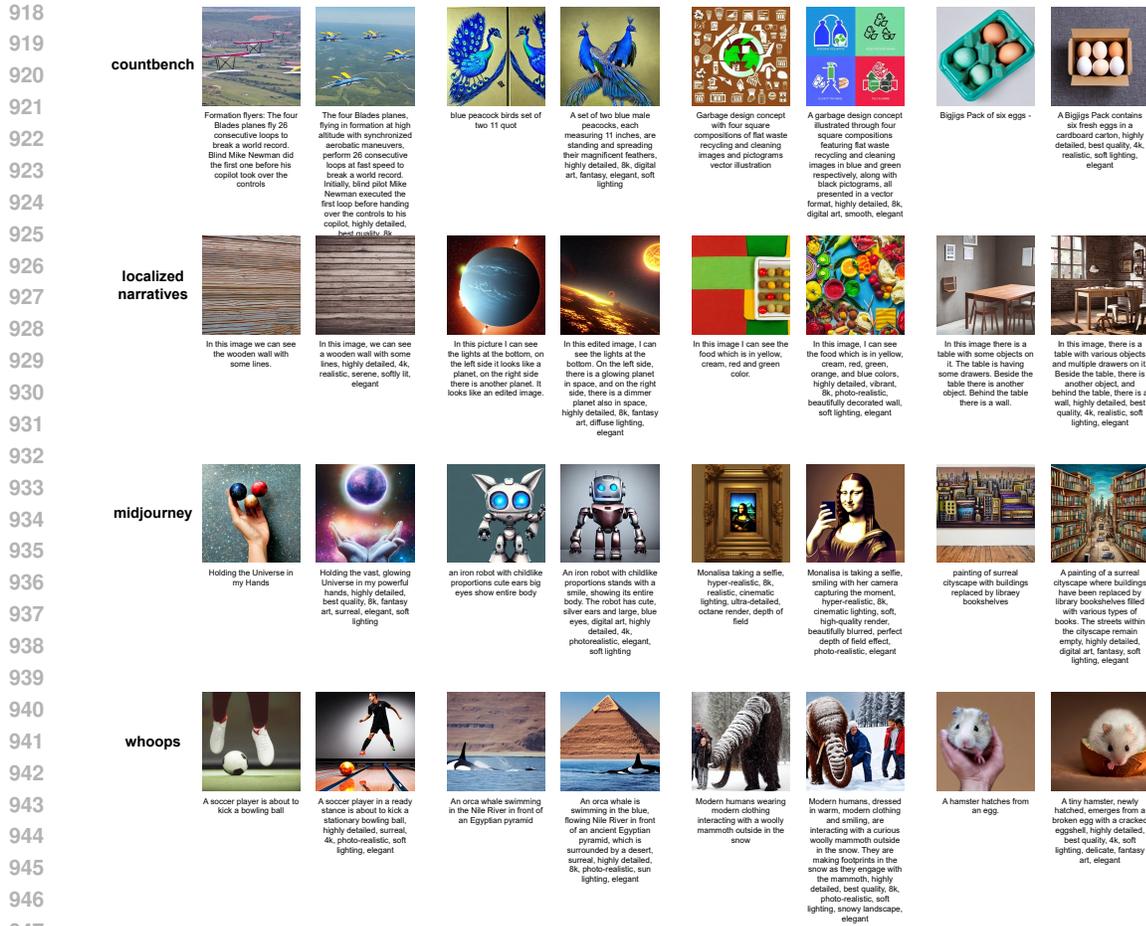


Figure 10: Examples of images generated by Stable Diffusion v1.5. Each line corresponds to the results of prompts derived from distinct sub-datasets. The right image in each pair is generated from the prompt optimized by VisualPrompter, as detailed below the image.

decline. We attribute this fact to the limitation of the Stable Diffusion v1.5, which is not sufficiently capable of comprehending human poses and processing long textual descriptions.

The experimental results demonstrate that our method achieves significant improvements across all subcategories of the TIFA benchmark, indicating its strong capability in optimizing naturally described sentences. On the DSG benchmark, our model shows consistent performance gains in most subcategories, which further validates its generalization ability across different evaluation metrics.

While our approach delivers robust performance overall, we observe marginal performance decreases when applied to SD v1.5 and Janus-Pro models on the DSG benchmark. This can be attributed to two main factors: (1) SD v1.5’s limited capacity in processing moderately complex sentences often leads to substantial detail loss, and (2) Janus-Pro’s superior language understanding capability makes it particularly responsive to NeuroPrompt-optimized sentences, creating a slightly higher result. We plan to conduct more in-depth investigations to develop better optimization strategies in our future work.

F MORE CASES OF VISUALPROMPTER

We provide more results for our optimization approach. As shown in Figure 10, Figure 11, and Figure 12. Each image pair consists of one generated from the original prompt and another from the prompt optimized by our VisualPrompter, with the corresponding prompt text displayed beneath each image. The visualization results convincingly demonstrate that our VisualPrompter exhibits

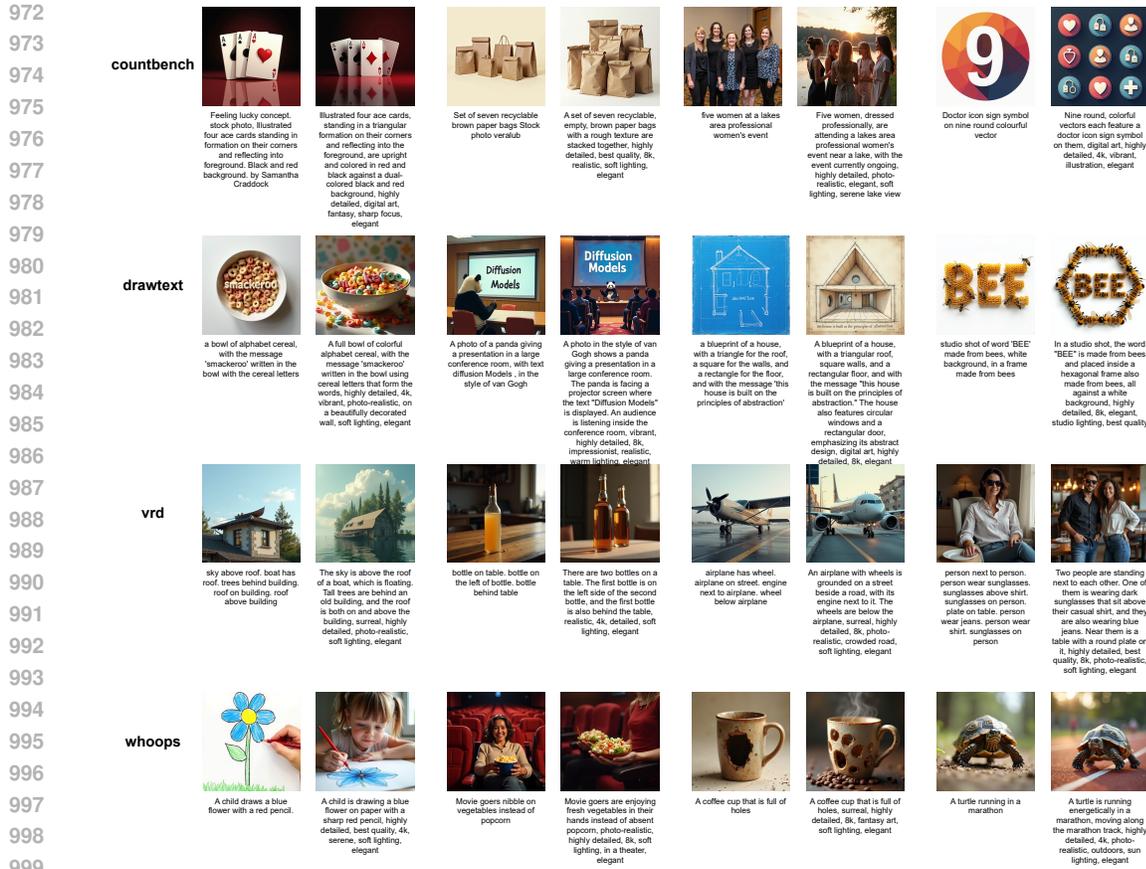


Figure 11: Examples of images generated by FLUX-dev. Each line corresponds to the results of prompts derived from distinct sub-datasets. The right image in each pair is generated from the prompt optimized by VisualPrompter, as detailed below the image.

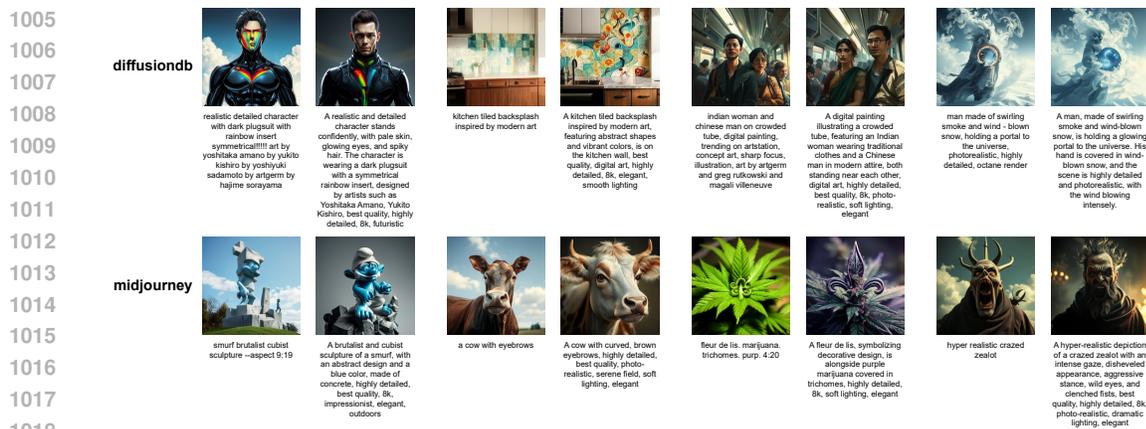


Figure 12: Examples of images generated by Janus-Pro. Each line corresponds to the results of prompts derived from distinct sub-datasets. The right image in each pair is generated from the prompt optimized by VisualPrompter, as detailed below the image.

robust adaptability to various prompts and substantiates superior performance in rectifying multiple types of generation artifacts.



Figure 13: Some failure cases of VisualPrompter. The two rows of images are from Flux-dev and Janus-pro, respectively.

Table 10: Impact of random seed. We report the mean and standard deviation of semantic accuracy based on the DSG benchmark.

T2I Models	Prompt Modification Approach		
	None	Qwen3-32B	VisualPrompter
SD v1.5	67.4 ± 0.13	60.9 ± 0.43	69.8 ± 0.56
FLUX-dev	79.2 ± 0.08	79.7 ± 0.21	84.3 ± 0.12

G FAILURE CASES OF VISUALPROMPTER

Despite its effectiveness, our VisualPrompter still exhibits certain limitations. A few typical failure cases are shown in Figure 13. Most failure cases stem from the inherent constraints of the generative model, particularly its suboptimal accuracy in multi-object generation, which occasionally leads to irrecoverable artifacts in the final output. Additionally, some errors arise from the VLM’s judgment, where semantically subtle or ambiguous prompts are incorrectly validated, preventing VisualPrompter from identifying and optimizing these issues. Nevertheless, we anticipate that advancements in VLM capabilities will progressively mitigate these challenges, further enhancing the robustness of our model.

H INFLUENCE OF RANDOM SEED

To investigate the impact of random seeds, we conduct extensive experiments on two text-to-image generative models using five different random seeds and present the comprehensive results in Table 10. The experimental outcomes show remarkably consistent performance across all trials, with only minimal deviations observed. This stability clearly demonstrates that the influence of random seeds on our results is statistically negligible.

The observed robustness can be attributed to two key factors. First, the DSG benchmark incorporates an extensive set of over a thousand diverse prompts, providing sufficient data volume to effectively average out any potential randomness. Second, while random variations might affect certain aspects like layout generation, they have minimal impact on semantic content representation. This is particularly relevant as our VisualPrompter framework is specifically designed to enhance and complement semantic information processing, further mitigating any potential variability from random initialization.

I POTENTIAL TOWARD MULTIMODAL ANALYSIS

To evaluate the broader applicability of our approach, we integrate ControlNet (Zhang et al., 2023) with human pose estimation, enabling image generation conditioned on both skeletal inputs and

1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095

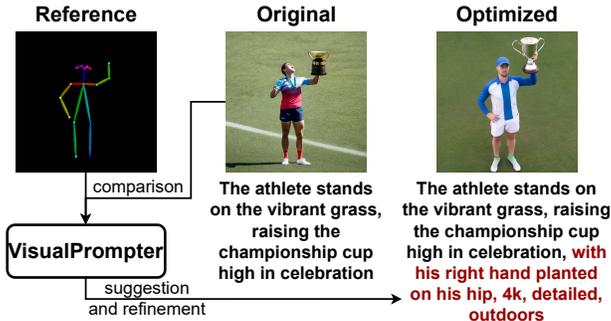


Figure 14: Multimodal application of VisualPrompter.

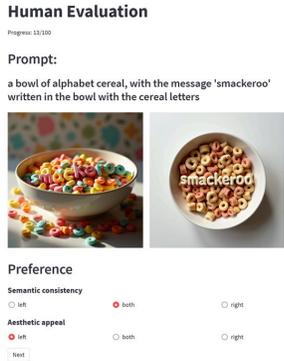


Figure 15: Screenshot of human evaluation interface.

1096 textual prompts. We quantitatively assess skeleton-to-image consistency using our VisualPrompter’s
1097 VLM module, which additionally provides refinement suggestions for improved alignment.

1098 As illustrated in Figure 14, our framework demonstrates promising potential for multimodal anal-
1099 ysis—effectively bridging visual structural data (skeletons) and linguistic descriptions. This multi-
1100 modal capability suggests significant versatility in handling diverse input modalities. We identify
1101 this as a pivotal research direction and plan to systematically explore: (a) cross-modal interaction
1102 mechanisms, (b) scalability to additional modalities, and (c) quantitative evaluation frameworks,
1103 which will constitute a cornerstone of our future work.

1104
1105 **J USER STUDY SETTINGS AND MORE COMPARISON**

1106 We provide a simple test system for our human evaluation experiment based on *streamlit*, as shown in
1107 Figure 15. Departing from the experiments in previous text-to-image prompt engineering studies that
1108 only focus on aesthetic evaluation, our assessment framework specifically incorporates an analysis
1109 of semantic alignment between generated visual content and the corresponding textual prompts,
1110 which measures how well the generated images align with the user intent.

1111 As shown in Figure 15, a raw prompt and a pair of images are given to the participants for each
1112 test. The images pair consist of an image generated by the original prompt above and another image
1113 generated by the optimized prompt. In our experiment, the optimized prompts are invisible to the
1114 participants. To eliminate potential bias, images in each pair are randomly assigned to left or right
1115 positions. Participants are required to evaluate two key aspects individually: the visual consistency
1116 with the given prompt and the aesthetic appeal of each image. Both provide essential insights into
1117 the overall quality of the visual representation.

1118 Table 11: Comparison of VisualPrompter with other optimized methods. User preferences are mea-
1119 sured as the percentage of choices for VisualPrompter, the competing model, or a tie, respectively.

1120
1121
1122

T2I Models	Aspects	Methods in Comparison		
		NeuroPrompts	Promptist	BeautifulPrompt
SD v1.5	Semantic	58% / 24% / 18%	52% / 27% / 21%	60% / 26% / 14%
	Aesthetics	41% / 43% / 16%	44% / 39% / 17%	35% / 46% / 19%
FLUX-dev	Semantic	44% / 30% / 26%	43% / 39% / 18%	44% / 31% / 25%
	Aesthetics	34% / 46% / 20%	37% / 41% / 22%	27% / 49% / 24%

1123
1124
1125
1126
1127
1128
1129
1130 To evaluate the semantic consistency and aesthetic quality of our model, we conduct the user study
1131 above based on Stable Diffusion v1.5 and FLUX-Dev, comparing it with other optimized models
1132 rather than the original generators. As illustrated in Table 11, the results demonstrate a clear user
1133 preference for our model in terms of semantic consistency. Furthermore, it achieves competitive
performance on aesthetic metrics, underscoring its comprehensive capability.

Table 12: Summarized results on DSG benchmark using Llama-Vision. Reported scores are based on the percentage of “yes” answers to the questions, while the best scores are highlighted in boldface.

Methods	DSG benchmark				Average
	SD 1.5	SD 2.1	Flux-dev	Janus-Pro	
Baseline	67.2	72.0	79.4	79.5	74.5
NeuroPrompts	59.0	69.5	77.2	81.4	71.8
Promptist	65.9	68.9	77.6	78.2	72.7
BeautifulPrompt	49.4	51.1	53.4	57.0	52.7
VisualPrompter	70.1	77.9	85.4	82.9	79.1

Table 13: Summarized results on TIFA benchmark using Llama-Vision. Reported scores are based on the percentage of “yes” answers to the questions, while the best scores are highlighted in boldface.

Methods	TIFA benchmark				Average
	SD 1.5	SD 2.1	Flux-dev	Janus-Pro	
Baseline	76.1	81.0	88.4	85.6	82.8
NeuroPrompts	64.3	77.2	85.5	89.8	79.2
Promptist	72.7	79.0	86.9	86.8	81.4
BeautifulPrompt	53.0	55.9	62.8	63.5	58.8
VisualPrompter	81.7	83.4	94.3	94.5	88.5

K ROBUSTNESS OF EVALUATION

We notice that the use of the same VLM for both the internal optimization feedback and the final evaluation may introduce potential bias. To address this concern and validate the robustness of our reported performance, we conduct an additional experiment where the results are re-evaluated using another VLM. We employ Llama-3.2-11B-Vision for a comparative analysis, as it is both open-source and readily accessible. As summarized in Table 12 and Table 13, the evaluation scores between the two VLMs are highly consistent. Despite minor fluctuations in evaluation scores, the relative performance ranking of all optimization methods remains identical under both VLM assessors, underscoring that our comparative conclusions are robust to the choice of evaluator.

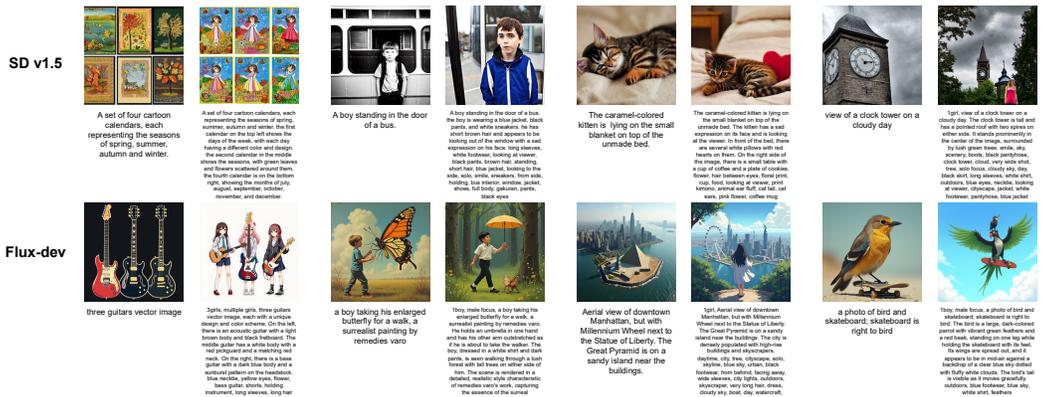


Figure 16: Examples of images optimized by TIPO from Stable Diffusion V1.5 and Flux-dev.

L ADDITIONAL METHODS FOR COMPARISON

To situate our work within the broader landscape of prompt optimization techniques and provide a more comprehensive comparison, we extend our evaluation to include TIPO (Yeh et al., 2024), a

Table 14: Detailed results of TIPO on different generative models. The value in parentheses indicates the performance gap between TIPO and our VisualPrompter.

Generators	Category of source texts										Average
	whoops	localized	posescript	vr4	countbench	midjourney	tifa160	draw text	stanford	diffusion db	
SD v1.5	50.3	54.0	60.5	60.2	49.1	45.8	62.7	44.1	56.5	55.6	54.4 (-15.1)
SD v2.1	62.3	61.8	65.2	63.5	60.9	52.9	69.3	54.6	66.2	60.4	62.1 (-14.9)
Flux-dev	72.0	80.7	73.1	82.1	63.5	39.0	78.9	70.9	83.9	51.5	70.1 (-14.2)
Janus-Pro	73.5	77.0	66.3	80.9	63.5	47.4	79.2	70.5	80.3	59.3	70.3 (-12.3)

(a) Semantic evaluation on DSG benchmark.

Generators	Category of source texts				Average
	coco	drawbench	partiprompt	paintskill	
SD v1.5	65.8	55.2	59.2	48.1	61.3 (-19.4)
SD v2.1	71.4	54.6	64.4	57.2	66.6 (-16.2)
Flux-dev	83.9	69.0	75.6	76.0	79.5 (-14.3)
Janus-Pro	83.3	70.8	75.8	74.5	79.1 (-14.6)

(b) Semantic evaluation on TIFA benchmark.

lightweight pre-trained prompt optimizer that refines prompts by sampling from a learned target sub-distribution. Our empirical findings suggest that TIPO learns a highly specialized sub-distribution from its training data, heavily favoring the introduction of human subjects and portrait details, as shown in Figure 16. While effective for its target domain, this bias causes it to systematically alter the semantic content of generic prompts in a way that frequently deviates from user intent. Quantitative results in Table 14 confirm this limitation, with TIPO underperforming our model by a large margin on both DSG benchmark and TIFA benchmark. These results demonstrate that the TIPO model is not suitable as a universal prompt optimization model and has limited applicability. This highlights a key strength of our training-free, feedback-driven approach, whose model-agnostic design does not inherit such distributional biases, enabling more generalizable and semantically faithful prompt optimization.

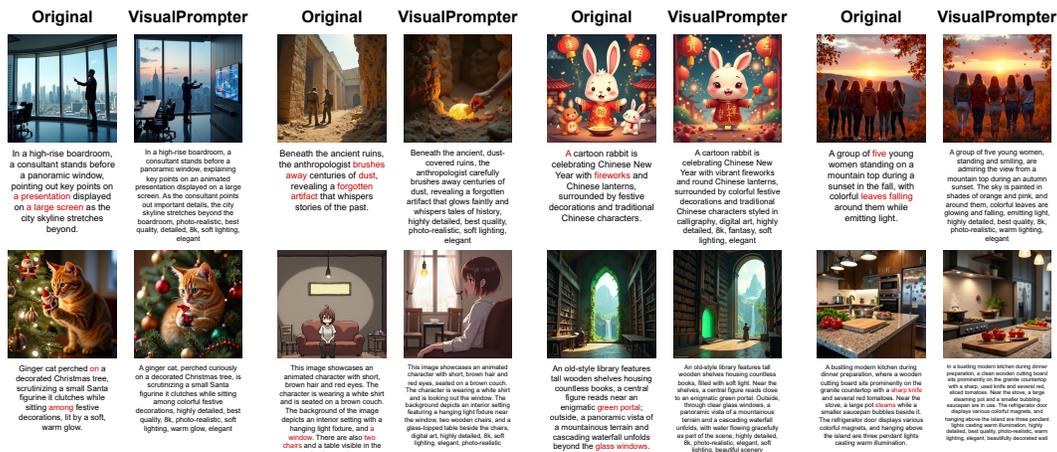


Figure 17: Visual comparison between the original Flux-dev (left) and our optimized model (right) on complex prompts.

M COMPARISON ON COMPLEX PROMPTS

To investigate performance on complex prompts, we collected additional prompts describing intricate scenes and generated images using Flux-dev, a generative model capable of handling complex scenarios. We then compared these results with outputs from our VisualPrompter-optimized prompts. As shown in Figure 17, our model demonstrates a clear ability to correct missing or er-

1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295



Figure 18: Examples of multilingual optimization based on Chinese, validating the effectiveness of the proposed method across languages. We employ Kolors (Kolors, 2024) as the image generator.

reoneous semantics in the generated images. We also observed that the capability of the underlying generative model is crucial. Since long prompts already contain substantial detail, our optimization approach remains constrained when dealing with structures that are inherently challenging for the base generative model. We plan to continue research in this direction.

To validate the effectiveness of our method in optimizing multilingual text-to-image generation, we use Chinese text accepted by Qwen for testing. Since Stable Diffusion v1.5 and FLux-dev do not support Chinese, we select the text-to-image model Kolors (Kolors, 2024) as our subject and modify the prompt templates in VisualPrompter to support Chinese input. Visualization examples are shown in Figure 18. The results indicate that our method can also enhance Chinese text-to-image generation. This effectiveness stems from our language-agnostic atomic semantic structures and analytical approach, allowing for the processing of different languages supported by the underlying LLM.

N DISCLOSURE OF LLM USE

The authors confirm that the use of LLMs, which was limited exclusively to polishing sentence structures and correcting grammatical errors, and was not involved in generating ideas, contributions, data, or any substantive content.