EvoStealer: Differential Evolution for Prompt Template Stealing Against Text-to-Image Synthesis

Anonymous ACL submission

Abstract

Prompt trading has emerged as a significant intellectual property concern in recent years, where vendors entice users by showcasing sample images before selling prompt templates that can generate similar images. This work investigates a critical security vulnerability: attackers can steal prompt templates using only a limited number of sample images. To investigate this threat, we introduce **PRISM**, a prompt-stealing benchmark consisting of 50 templates and 450 images, organized into Easy and Hard difficulty levels. To identify the vulnerabity of VLMs to prompt stealing, we propose **EvoStealer**, a novel template stealing method that operates without model fine-tuning by leveraging differential evolution algorithms. The system first initializes population sets using multimodal large language models (MLLMs) based on predefined patterns, then iteratively generates enhanced offspring through MLLMs. During evolution, EvoStealer identifies common features across offspring to derive generalized templates. Our comprehensive evaluation conducted across open-source (INTERNVL2-26B) and closed-source models (GPT-4O and GPT-4O-MINI) demonstrates that EvoStealer's stolen templates can reproduce images highly similar to originals and effectively generalize to other subjects, significantly outperforming baseline methods with an average improvement of over 10%. Moreover, our cost analysis reveals that EvoStealer achieves template stealing with negligible computational expenses.

1 Introduction

Recent advancements in text-to-image generation (Liu et al., 2024a; Cao et al., 2024), particularly in multimodal large language models (MLLMs) (Liu et al., 2024a; Wang et al., 2024) and diffusion models (Ho et al., 2020; Sohl-Dickstein et al., 2015), have significantly improved image generation performance. However, crafting the perfect prompt to produce desired output images

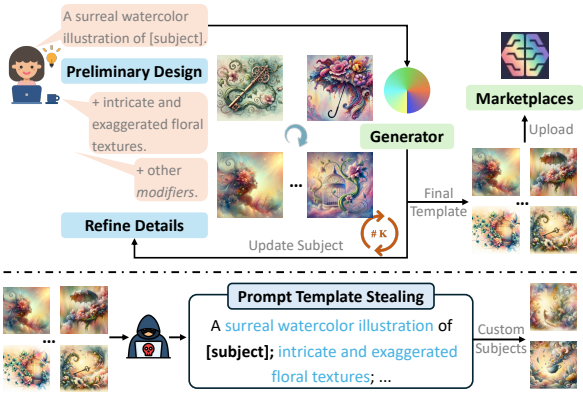


Figure 1: Top: Illustrating the legitimate development of text-to-image prompt templates. Bottom: Depicting unauthorized extraction of proprietary prompt templates

results remains a meticulous process that requires significant expertise and time investment (Refer to Figure 1(top)). This challenge has catalyzed the emergence of prompt trading, a novel business model exemplified by platforms like PromptBase[1] and LaPrompt[2]. On these platforms, creators upload meticulously crafted prompt templates (viewable post-purchase) alongside multiple sample images (publicly visible). Customers attracted to these samples can purchase the template, then merely modify the subject specification to generate new images that preserve the original stylistic elements. In this context, the platform's copyright and security vulnerabilities raise significant concerns. If attackers reverse-engineer the proprietary templates by analyzing the visible samples, they could significantly compromise sellers' intellectual property rights and threaten the platform's business model (See Figure 1 (bottom)). We term this attack *prompt template stealing*.

Existing methods for prompt stealing attacks (Shen et al., 2024; Sha and Zhang, 2024; Naseh et al., 2024) focus on reconstructing individ-

[1]https://promptbase.com/
[2]https://laprompt.com/

ual prompts for each sampled image, rather than recovering a general prompt template for the entire group of sampled images. As a result, the prompts reconstructed by these methods are specific to each image and lack generalizability, which limits their applicability in practical scenarios, as illustrated in Figure 1. For example, in the case of the woman image located in Figure 1, a stolen prompt might include the "golden sun" as a distinctive element. Nevertheless, a comparison with the other three images demonstrates that the "golden sun" is not a shared characteristic among them.

To fill this gap, we build a new and comprehensive benchmark named PRISM, comprising 50 prompt templates stratified across two difficulty levels (Easy and Hard) and spanning 9 distinct subjects, sourced from a specialized prompt trading platform. Utilizing DALL·E 3, we generated 450 images, with each group methodically partitioned into 5 in-domain and 4 out-of-domain images to systematically evaluate both model fitting capability and generalization performance.

Besides, we introduce EvoStealer, a novel template stealing methodology derived from the differential mutation algorithm in evolutionary computation. Our approach strategically leverages mutation and crossover operations within the search space to effectively mitigate overfitting and circumvent local optima, precisely aligning with template stealing objectives. We integrate large language models (LLMs) spanning both open-source and closed-source domains, specifically utilizing InternVL2-26B, GPT-4o, and GPT-4o-mini. By combining these models with a differential evolution algorithm, we generate prompt templates characterized by exceptional stability and robust generalization capabilities. Comprehensive experimental evaluations are conducted across easy and hard difficulty levels. The results demonstrate EvoStealer's remarkable performance: the methodology efficiently reproduces images highly similar to original templates while simultaneously exhibiting strong cross-subject generalizability. This enables large-scale image generation maintaining consistent stylistic characteristics.

Our main contributions are as follows:

(1) To the best of our knowledge, this is the first systematic study on prompt template stealing, revealing its severity as an emerging security threat and empirically demonstrating its significant risk to intellectual property protection;

(2) This study introduces PRISM, the first benchmark for prompt template stealing, and EvoStealer, a plug-and-play attack framework that requires no fine-tuning, significantly improving practicality and scalability;

(3) We conducted extensive experiments on both open-source models (INTERNVL2-26B) and closed-source models (GPT-4O, GPT-4O-MINI), with results validating the effectiveness of EvoStealer.

2 Related Work

We discuss two lines of related work: the text-to-image prompt stealing attacks and the evolutionary algorithms in LLMs.

2.1 Text-to-Image Prompt Stealing Attack

Prompt stealing attacks, or prompt extraction attacks, aim to infer the input from a model's output. A successful attack infringes on intellectual property and poses significant risks to prompt trading platforms in the era of LLMs. However, such attacks are more challenging in text-to-image generation due to the greater uncertainty in image generation compared to text. CLIP Interrogator employs CLIP (Radford et al., 2021) to extract the subject and then selects phrases matching the target image from predefined sets (Udo and Koshinaka, 2023). Shen et al. (2024) fine-tunes 2 models to extract image subjects and modifiers separately, combining them for the attack. Building on this, Naseh et al. (2024) employ GPT-4V to iteratively optimize the prompt, resulting in higher quality. Unlike these works, EvoStealer targets the extraction of a generalizable prompt template, offering greater practical value compared to stealing individual prompts.

2.2 Evolutionary Algorithms in LLMs

Recent researches combining evolutionary algorithms with LLMs have demonstrated strong and stable performance across various tasks (Yang et al., 2023; Liu et al., 2023). Some studies leverage the rich domain knowledge and powerful text analysis capabilities of LLMs to accelerate the search process in evolutionary algorithms, particularly in tasks involving complex reasoning (Meyerson et al., 2024; Liu et al., 2024b; Lange et al., 2024; Brahmachary et al., 2024) and interpretability (Chiquier et al., 2025). Conversely, some studies capitalize on the stability of evolutionary algorithms to utilize LLMs for generating higher-quality prompt words (Xu et al., 2022; Prasad et al., 2022; Guo

et al., 2023; Fernando et al., 2023). In this paper, we use evolutionary algorithms to progressively generate style descriptors that closely resemble multiple example images, thereby achieving prompt template stealing.

3 Data Consturction

In this section, we introduce the threat model of prompt template stealing, providing a detailed description of the attacker's existing conditions, constraints, and objectives. We then detail our methodology for developing PRISM, a comprehensive benchmark designed to realistically simulate this attack scenario. The specifics are presented below.

3.1 Threat Model

The attack scenario is grounded in real-world applications. Attackers have access to two pieces of information from the prompt trading platform: 9 sample images and the generative model (e.g., DALL-E 3 [3] or Midjourney [4]). While attackers can interact with the model via an API, they are not privy to its internal parameters. Their objectives are twofold: first, to generate images that closely resemble, or even replicate, the sample images by using the same subject with the stolen prompt template; and second, to alter the subject within the template and generate images that retain the same style as the sample images.

3.2 Benchmark Construction

Currently, no specialized benchmark exists for prompt template stealing research. To address this gap, we introduce PRISM, a novel benchmark comprising 50 freely available prompt templates sourced from PromptBase and LaPrompt. These templates are divided into two equal groups of 25 templates each, categorized as "Easy" and "Hard" based on complexity. Each group encompasses 9 distinct subject categories. We utilize DALL·E 3 as our generation model, combining each prompt template with the 9 subjects to produce 450 unique images. To ensure quality control, we implemented a comprehensive manual review process focusing on two key criteria: subject-prompt alignment and stylistic consistency across template-generated images. For each group, we designated the first 5 generated images as in-domain data to assess similarity between original and stolen prompt. The remaining 4 images serve as out-of-domain data, enabling evaluation of prompt template generalization capabilities across diverse subjects. For comprehensive details on the benchmark construction methodology, please refer to Appendix A.

[3] https://openai.com/index/dall-e-3/

[4] https://www.midjourney.com/home

4 EvoStealer

In this section, we introduce the three main steps of EvoStealer: Image Element Extraction, Differential Evolution, and Fitness Function. The details are presented below.

4.1 Image Element Extraction

High-quality prompts for text-to-image generation typically consist of a subject and several modifiers (Liu and Chilton, 2022; Oppenlaender, 2024). The subject defines the object or scene depicted in the image, such as *"a woman with a flower crown"* or a more intricate description like *"Woodland creatures gather around a shimmering pond, surrounded by trees and glowing flowers, creating a peaceful scene"*. The modifiers specify the style of the image, including aspects such as artistic style and resolution. While multimodal models can accurately identify simple subjects, they often misinterpret complex subjects, mistakenly treating parts of the subject as style modifiers. For instance, in the case of the complex subject "*peaceful scene*", the model may misinterpret "*peaceful*" as a style modifier, contaminating the intended description.

To address this issue, we define an image element extraction pattern: <Subject, Modifiers, Supplements>. The subject describes the object or scene, while the modifiers are categorized into four types: *Artistic Style, Visual Composition and Structure, Aesthetic and Emotional Atmosphere*, and *Medium and Material*. For further details, please refer to Appendix B. We have imposed the aforementioned constraints on the modifiers to ensure that the model describes the image solely from the relevant perspectives within the four predefined categories. This restriction contributes to the stability and controllability of modifier extraction. However, such a constraint may limit the model's ability to fully capture the diversity of style features. To mitigate this limitation, we incorporate supplements as a compensatory measure. Supplements encompass descriptions outside the four categories and can include individual words, phrases, or even sentences, such as "*radiating lines suggesting motion*" or "*subtle transitions between colors*".

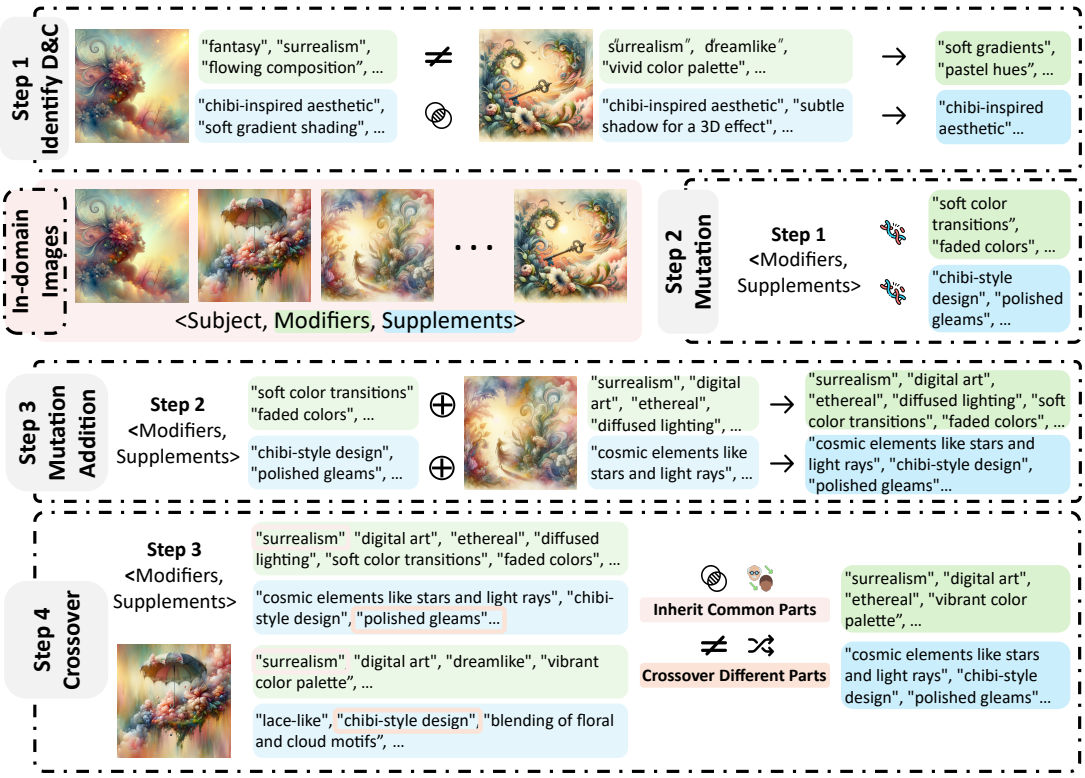


Figure 2: The key steps of EvoStealer in differential evolution, including the identification of differences and commonalities, mutation, mutation addition, and crossover operations.

4.2 Differential Evolution

Firstly, we introduce the theory of differential evolution. The process of generating offspring through the differential evolution algorithm is represented using numerical vectors. Initially, each vector in the population is sequentially selected as the base vector, denoted as α. Then, three individuals, x_1, x_2, x_3, are randomly chosen from the population to perform the mutation operation. Specifically, the difference between x_2, x_3 is calculated, and this difference undergoes mutation. The mutated difference is then added to x_1 to produce a new vector, denoted as β. The mutation operation is mathematically expressed as: $\beta = x_1 + F(x_2 - x_3)$, where F represents the mutation factor, which controls the magnitude of the mutation. Finally, a crossover operation is applied to the vectors α and β to generate the offspring.

Figure 2 illustrates the differential evolution process implemented in EvoStealer. In Step 1, we differentiate between modifiers and supplements due to their distinct characteristics, particularly in terms of controllability and unpredictability. For modifiers, we focus on identifying the differences between the two sets, while for supplements, we concentrate on their common components. This approach is grounded in the understanding that the uncontrollability of supplements introduces unique features specific to individual images. Additionally, supplements typically contain more tokens than modifiers, which results in a greater influence on the visual representation of the image and, consequently, on the generalization ability of the template. In Step 2, we randomly select an image from the in-domain dataset to influence the mutation process. This strategy serves two purposes: first, it helps filter out modifiers that do not align with the image (e.g., in the case of a surrealistic style image, modifiers such as "cartoon style" are excluded); second, the image, serving as a mutation variable, introduces additional contextual information. As mentioned earlier, the initial version of EvoStealer directly derives image element extraction, which results in an over-reliance on the quality of this extraction. By incorporating the image in Step 2, we enable the population to gain valuable information that may otherwise be overlooked, thus mitigating the drawback of over-dependence on image element extraction. In Step 3, no modifications are made, and the two components are simply combined to generate the mutated description. In Step 4, in contrast to the direct crossover used in genetic

Method	DINO	CLIP_{img}	CLIP_{txt}	SigLIp_{img}	SigLIp_{txt}	Average	Human Evaluation
Easy Benchmark							
BLIP 2 (Li et al., 2023)	62.07	79.38	48.35	82.32	52.69	64.96	3.42
CLIP Interrogator	69.93	82.76	54.14	85.86	62.59	70.86	4.02
PromptStealer (Shen et al., 2024)	63.73	77.90	49.21	82.73	61.93	67.10	3.78
EvoStealer (INTERNVL2-26B)	74.68	84.46	68.94	87.88	**74.93**	78.18	4.32
EvoStealer (GPT-4O-MINI)	73.87	84.79	72.12	88.38	71.80	78.19	4.30
EvoStealer (GPT-4O)	**75.83**	**85.30**	**74.41**	**89.14**	72.75	**79.49**	4.52
Hard Benchmark							
BLIP 2 (Li et al., 2023)	61.16	76.67	46.04	80.51	50.74	63.02	3.24
CLIP Interrogator	66.45	78.26	54.62	82.45	60.78	68.51	3.66
PromptStealer (Shen et al., 2024)	60.01	75.58	47.10	79.20	59.71	64.32	3.48
EvoStealer (INTERNVL2-26B)	70.16	80.63	63.02	84.66	68.14	73.32	4.17
EvoStealer (GPT-4O-MINI)	**71.05**	81.02	67.64	84.88	69.00	74.72	4.12
EvoStealer (GPT-4O)	69.24	**81.34**	**70.61**	**85.28**	**69.27**	**75.15**	4.24

Table 1: The overall evaluation results for the in-domain data, with the bolded values indicating the best scores.

algorithms, EvoStealer first identifies the common parts between the two individuals. When generating the new offspring, the common parts are fully inherited, while only the differing parts undergo the crossover operation. This design approach strikes a balance between the exploration and exploitation of the algorithm, facilitating effective exploration while ensuring that generalization constraints are preserved.

4.3 Fitness Function

The fitness function is employed to assess the quality of offspring, with those exhibiting higher fitness scores being retained for progression to the next iteration. While the fitness function does not directly influence the offspring generation, it guides the search direction throughout the evolution process. Our fitness function incorporates both the semantic similarity of the text and the style similarity of the image. Specifically, for each offspring (i.e., a prompt template), we sequentially replace the subject within the template and calculate its semantic similarity with the ground truth. Additionally, we randomly select a subject and use the target model (DALL·E 3) to generate the corresponding image, subsequently calculating the similarity between this generated image and the corresponding image from the in-domain dataset. The mathematical formulation is as follows:

$$F = \frac{1}{n}\sum_{i=1}^{n}\left(\lambda\left(\frac{\mathbf{T}_{\text{off}}(i)\cdot\mathbf{I}_{\text{gt}}(i)}{\|\mathbf{T}_{\text{off}}(i)\|\|\mathbf{I}_{\text{gt}}(i)\|}\right)\right) + (1-\lambda)\left(\frac{\mathbf{I}_{\text{off}}\cdot\mathbf{I}_{\text{gt}}}{\|\mathbf{I}_{\text{off}}\|\|\mathbf{I}_{\text{gt}}\|}\right)$$

Where off and gt denote the offspring and ground truth, respectively, and $\mathbf{T}$ and $\mathbf{I}$ represent the text and image embeddings. The parameter λ serves as a balance factor to weight the two similarity measures.

5 Experiments

We employ PRISM to evaluate the vulnerability of image generation models to prompt template stealing. Following recent works (Shen et al., 2024; Naseh et al., 2024; Huang et al., 2024), we employ subject similarity, style similarity, and semantic similarity metrics to evaluate the performance of image generation models against prompt template stealing (Section 5.3). These metrics demonstrate higher agreement with human annotations than previous approaches. Additionally, we conduct human evaluation to measure the quality of prompt stealing.

5.1 Baselines

Our baselines encompass models for both caption generation (BLIP-2) and prompt stealing attack (CLIP Interrogator and PromptStealer).

- **BLIP-2**: BLIP-2 (Li et al., 2023) is a multimodal model that aligns text with images using a lightweight Querying Transformer to connect a frozen image encoder with LLMs. In this study, we employ the BLIP-2-opt-2.7b model to generate image descriptions.
- **CLIP Interrogator**: CLIP Interrogator [5] uses CLIP to generate image descriptions, incorporating prompts from preset categories such as artists, flavors, and mediums. It encodes

[5] https://github.com/pharmapsychotic/clip-interrogator/tree/main

Method	DINO	CLIP_{img}	CLIP_{txt}	SigLIp_{img}	SigLIp_{txt}	Average	Human Evaluation
Easy Benchmark							
CLIP Interrogator	64.02	78.72	53.95	82.98	63.73	68.68	3.80
PromptStealer (Shen et al., 2024)	60.53	75.53	51.37	81.19	61.16	65.96	3.64
EvoStealer (INTERNVL2-26B)	72.93	83.13	68.63	**87.24**	**74.54**	77.29	4.36
EvoStealer (GPT-4O-MINI)	74.53	83.60	71.87	85.28	73.30	77.71	4.47
EvoStealer (GPT-4O)	**75.14**	**83.91**	**74.18**	85.75	73.53	**79.10**	4.60
Hard Benchmark							
CLIP Interrogator	62.23	69.90	51.66	75.19	58.51	63.50	3.74
PromptStealer (Shen et al., 2024)	58.53	70.42	45.29	74.38	55.07	60.74	3.46
EvoStealer (INTERNVL2-26B)	68.92	78.96	61.29	83.37	67.87	72.08	4.32
EvoStealer (GPT-4O-MINI)	67.76	79.55	66.91	84.13	68.84	73.44	4.35
EvoStealer (GPT-4O)	**67.00**	**80.50**	**69.27**	**84.55**	**69.79**	**74.22**	4.48

Table 2: The overall evaluation results for the out-of-domain data, with the bolded values indicating the best scores.

both the image and text with the CLIP model, calculates their similarity, and generates the most matching description.

- **PromptStealer**: PromptStealer (Shen et al., 2024) consists of two modules: the Subject Generator, fine-tuned on BLIP to extract image subjects, and the Modifier Detector, a multi-class classifier that selects style modifiers based on similarity to predefined categories. The final prompt is generated by concatenating the subject and selected modifiers.

5.2 Experimental Settings

Due to the inherent difficulties in subject identification and replacement within BLIP-2-generated prompts, its evaluation is limited to in-domain data only. For both CLIP Interrogator and PromptStealer methods, we first extract subjects and modifiers from 5 in-domain samples and concatenate them to create prompts. We then randomly select a prompt and systematically replace its subject with subjects from the out-of-domain group. For PromptStealer, we maintain a threshold value of 0.6. In EvoStealer's implementation, we extract prompt templates from in-domain data and perform sequential subject substitutions, using 9 different subjects to generate the final prompts. Both the population size and generation count are set to 5, with the temperature parameter set to 0 to ensure consistent results. We employ SigLIP (Zhai et al., 2023) for fitness score calculations and set λ to 0.5. All image generation is performed using DALL·E 3 with a resolution of 1024×1024 and standard quality settings.

5.3 Evaluation Metric

We adopt the evaluation framework proposed by Huang et al. (2024) and employ the following metrics to assess the performance of EvoStealer and baseline methods:

- **Subject Similarity**: To evaluate the similarity between subjects in paired images, we utilize the self-supervised model DINO (Oquab et al., 2023), as subject comparison is a crucial aspect of image similarity assessment.
- **Style Similarity**: To measure style consistency, we employ CLIP and SigLIP to extract style features from images generated using stolen prompts and compare them with the original images.
- **Semantic Similarity**: To assess prompt similarity, we compute the cosine similarity between embeddings of the stolen and target prompts, generated using CLIP and SigLIP.
- **Human Evaluation**: We recruit 3 external evaluators to rate the similarity between generated and target images on a scale of 1-5, where higher scores indicate greater similarity. For each group, we randomly sample 2 images from both in-domain and out-of-domain categories and calculate average scores. For out-of-domain samples, the evaluation focuses exclusively on style similarity.

5.4 Main Results

Tables 1 and 2 present comparative performance evaluations between EvoStealer and baseline methods using both in-domain and out-of-domain data.

The results demonstrate that EvoStealer consistently outperforms baseline approaches across all evaluation metrics.

Performance on in-domain data. EvoStealer outperforms other methods in both the Easy and Hard categories. For example, EvoStealer (GPT-4o) leads the second-best method, CLIP Interrogator, by 8.63% and 6.64% on the two datasets, demonstrating its ability to generate more accurate prompt templates and better stealing performance. Notably, EvoStealer excels in textual semantic comparison, as its prompts are significantly more effective. CLIP Interrogator and PromptStealer rely on simple concatenation of [subject] and modifiers, limiting variability. Additionally, CLIP's length restriction hampers modifier extraction. In contrast, EvoStealer generates diverse templates iteratively, avoiding these limitations. This aligns with the findings of Naseh et al. (2024).

Performance on out-of-domain data. As shown in Table 2, EvoStealer outperforms other methods, especially on out-of-domain data, where it demonstrates a larger advantage compared to in-domain data. For instance, EvoStealer (GPT-4o) leads by more than 10% across all data types, indicating better generalization of stolen templates to different subjects. Furthermore, as seen in Table 1, EvoStealer's performance on out-of-domain data shows minimal degradation, while CLIP Interrogator and PromptStealer experience average degradations of 3.60% and 2.36%, respectively. This is due to EvoStealer's effective template stealing by extracting common features across multiple images.

Comparison of performance across different models. As shown in Tables 1 and 2, GPT-4o outperforms the other models, followed by GPT-4o-mini, with InternVL2-26B performing slightly worse. However, the performance differences among these models are minimal. This is primarily due to EvoStealer's reliance on the models' text and image analysis capabilities, indicating that EvoStealer is highly compatible and not dependent on a specific multimodal model.

6 Analysis

In this section, we analyze the effects of EvoStealer's components, the iteration number, and the experimental costs.

Method	InDom.	OutDom.	Average
Ours	77.32	76.67	77.00
w/o. supp.	73.56	74.85	74.21
w/o. img.	75.89	75.57	75.73

Table 3: Results of the ablation study: Impact of omitting supplements in the extraction pattern (w/o supp.) and excluding image similarity in the fitness function (w/o img.), with the model employed being GPT-4.

6.1 Ablation Study

We remove the supplements from the extracted templates and the image similarity evaluation from the fitness function to examine their impact on EvoStealer. The results are shown in Table 3. As observed, removing either module results in decreased performance, with a more significant drop when supplements are removed—specifically, an average similarity reduction of 2.79%. This is because supplements provide additional details, such as image features and style information. As noted in Section 4.2, supplements are longer than individual modifiers, so their removal has a more pronounced effect on visual performance. A comparison of the performance before and after removing supplements is provided in Appendix D. Removing the image similarity evaluation from the fitness function causes a performance decrease of 1.27%, suggesting that including the comparison between the generated and target images in the fitness function helps guide the evolutionary process and accelerate convergence.

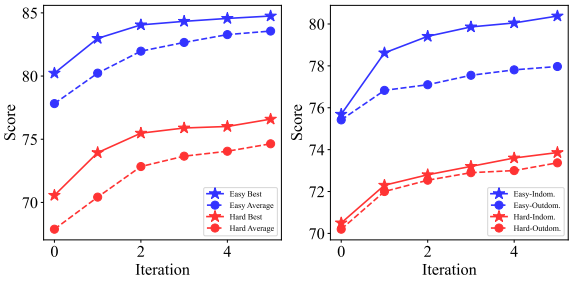


Figure 3: The convergence curve of EvoStealer, with the left half showing changes in fitness score and the right half depicting performance changes of the optimal prompt template for in-domain and out-of-domain data.

6.2 Effect of Number of Iterations

We select 10 groups of easy and 10 groups of hard cases to examine EvoStealer's convergence (we use GPT-4o as the analysis model), with results shown in Figure 3. The left section of the figure

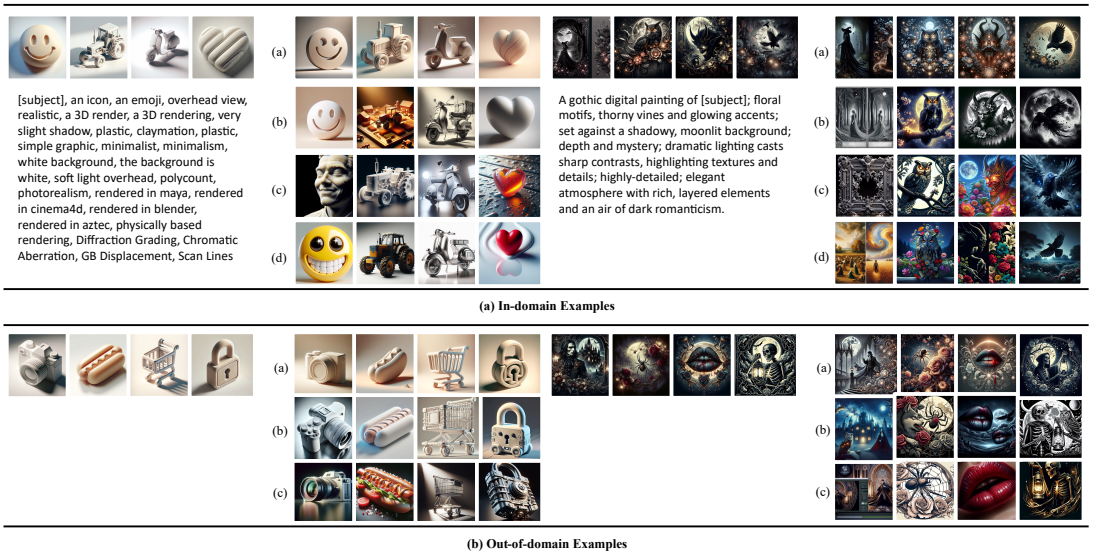


Figure 4: The attack results of EvoStealer compared to three baseline methods on both easy and hard examples. (a)-(d) represent EvoStealer, CLIP-Interrogator, PromptStealer, and BLIP2, respectively.

displays changes in the fitness score as evolution progresses, while the right section shows changes in the scores of the optimal templates for both in-domain and out-of-domain data. We observe that as evolution progresses, both the optimal and average fitness scores gradually increase, indicating that EvoStealer generates offspring with higher adaptability. The performance of the prompt templates steadily improves for both in-domain and out-of-domain data. Two examples are provided in Appendix E

6.3 Cost Analysis

To assess the practicality of EvoStealer, we analyzed the cost of stealing a prompt template. The primary overhead of EvoStealer consists of three components: population initialization, differential evolution (including the fitness function), and image synthesis. A detailed cost estimation process is provided in Appendix F. The results indicate that EvoStealer requires 144 API calls, generates 34 images (including 9 final synthesized images), and consumes approximately 119.1k tokens, amounting to a total cost of $1.70. While this is lower than the platform's pricing range of $3–9, the cost advantage is not substantial. However, as demonstrated in the ablation study in Section 6.1, costs can be further reduced by using open-source models or omitting image similarity calculations in the fitness function, enabling near-zero-cost stealing. Although this cost-reduced version performs slightly worse than the full EvoStealer model, it still significantly outperforms alternative approaches.

7 Case Study

To clearly demonstrate EvoStealer's advantages over baseline methods, we select an easy and a hard example for case study, with the results shown in Figure 4. The results show that, on in-domain data, EvoStealer generates images that closely match the style of the original images, with all four synthesized images maintaining stylistic consistency. In contrast, the four images generated by the other baseline methods exhibit significant style variation. On out-of-domain data, EvoStealer maintains the same style as in-domain images, successfully achieving subject generalization. In contrast, the other baseline methods fail to generalize. Additionally, we analyze three distinct failure cases (see Appendix G for details).

8 Conclusion

This paper investigates prompt template stealing—whether attackers can extract generalizable templates that maintain stylistic consistency using minimal sample images. To explore this scenario, we provide PRISM, a two-tier benchmark consisting of 50 templates and 450 images, organized into Easy and Hard difficulty levels. We also introduce EvoStealer, a template stealing method that combines differential evolution algorithms with MLLMs, enabling template stealing without the need for fine-tuning. Extensive experiments and analysis validate its effectiveness and practicality.

9 Limitations

The current implementation of EvoStealer and benchmark presents several methodological limitations:

1. EvoStealer's MLLM-based design offers simplified implementation without fine-tuning requirements and maintains robust performance across open datasets. However, this approach inherently limits the system's maximum performance to the capabilities of the underlying MLLMs.

2. Resource constraints restricted our benchmark to DALL·E-3 generated images, excluding other prominent models like Midjourney and Stable Diffusion. Nevertheless, the current benchmark adequately evaluates stealing method performance, with planned expansion to additional models in future work.

3. The benchmark's single-subject design facilitates comparative analysis but does not address multi-subject templates in real-world applications—a limitation to be addressed in subsequent research.

10 Ethical Considerations

EvoStealer's ability to extract prompt templates from minimal image examples enables attackers to generate multiple stylistically similar images through minor template modifications, posing significant risks to creators' intellectual property. This research highlights this security vulnerability, as understanding such threat models is essential for developing effective countermeasures.

While watermarking offers some protection, its implementation on trading platforms presents practical challenges. Watermarks can obscure image details, potentially deterring buyers or leading to customer dissatisfaction when purchased prompts fail to meet expectations. Our findings suggest that limiting the number of displayed images to 2-4 examples provides a simple yet effective defensive strategy.

Future research should prioritize developing robust protection mechanisms to safeguard both creators' rights and the integrity of the AI-generated content marketplace.

References

Shuvayan Brahmachary, Subodh M Joshi, Aniruddha Panda, Kaushik Koneripalli, Arun Kumar Sagotra, Harshil Patel, Ankush Sharma, Ameya D Jagtap, and Kaushic Kalyanaraman. 2024. Large language model-based evolutionary optimizer: Reasoning with elitism. *arXiv preprint arXiv:2403.02054*.

Pu Cao, Feng Zhou, Qing Song, and Lu Yang. 2024. Controllable generation with text-to-image diffusion models: A survey. *arXiv preprint arXiv:2403.04279*.

Mia Chiquier, Utkarsh Mall, and Carl Vondrick. 2025. Evolving interpretable visual classifiers with large language models. In *European Conference on Computer Vision*, pages 183–201. Springer.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2023. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2309.16797*.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2023. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532*.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. 2024. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818.

Robert Lange, Yingtao Tian, and Yujin Tang. 2024. Large language models as evolution strategies. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 579–582.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Fei Liu, Xi Lin, Zhenkun Wang, Shunyu Yao, Xialiang Tong, Mingxuan Yuan, and Qingfu Zhang. 2023. Large language model for multi-objective evolutionary optimization. *arXiv preprint arXiv:2310.12541*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024a. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Shengcai Liu, Caishun Chen, Xinghua Qu, Ke Tang, and Yew-Soon Ong. 2024b. Large language models as evolutionary optimizers. In *2024 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE.

Vivian Liu and Lydia B Chilton. 2022. Design guidelines for prompt engineering text-to-image generative models. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pages 1–23.

Elliot Meyerson, Mark J Nelson, Herbie Bradley, Adam Gaier, Arash Moradi, Amy K Hoover, and Joel Lehman. 2024. Language model crossover: Variation through few-shot prompting. *ACM Transactions on Evolutionary Learning*, 4(4):1–40.

Ali Naseh, Katherine Thai, Mohit Iyyer, and Amir Houmansadr. 2024. Iteratively prompting multimodal llms to reproduce natural and ai-generated images. *arXiv preprint arXiv:2404.13784*.

Jonas Oppenlaender. 2024. A taxonomy of prompt modifiers for text-to-image generation. *Behaviour & Information Technology*, 43(15):3763–3776.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. 2022. Grips: Gradient-free, edit-based instruction search for prompting large language models. *arXiv preprint arXiv:2203.07281*.

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

Zeyang Sha and Yang Zhang. 2024. Prompt stealing attacks against large language models. *arXiv preprint arXiv:2402.12959*.

Xinyue Shen, Yiting Qu, Michael Backes, and Yang Zhang. 2024. Prompt stealing attacks against {Text-to-Image} generation models. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 5823–5840.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR.

Honori Udo and Takafumi Koshinaka. 2023. Image captioners sometimes tell more than images they see. *arXiv preprint arXiv:2305.02932*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Hanwei Xu, Yujun Chen, Yulun Du, Nan Shao, Yanggang Wang, Haiyu Li, and Zhilin Yang. 2022. Gps: Genetic prompt search for efficient few-shot learning. *arXiv preprint arXiv:2210.17041*.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986.

A Data Collection

Our data collection and preprocessing pipeline consists of several systematic steps. Initially, we collect 100 free templates (excluding specific images) from PromptBase and LaPrompt, subsequently generating 900 corresponding images using DALL-E 3. Through manual curation, we eliminate templates that prove challenging to reproduce, such as those containing specific artistic style descriptors (e.g., Arkhip Kuindzhi). We then perform deduplication to remove images with highly similar styles and subjects, thereby preventing evaluation bias from data redundancy. The subsequent quality control process encompasses two primary aspects: subject alignment verification and style consistency assessment. When anomalous data is identified, we regenerate images using DALL-E 3 until they meet our quality criteria. Finally, we categorize our dataset into Easy and Hard classifications. The Hard category is characterized by: uncommon modifiers, abstract subject descriptions and rich image details. The token distribution of the complete dataset is illustrated in Figure 5, while Table 4 presents a detailed breakdown of token statistics for both Easy and Hard categories.

B Extraction Pattern Detail

Describing the style of an image requires including different perspectives. The style description of EvoStealer includes four categories: *Artistic Style, Visual Composition and Structure, Aesthetic and Emotional Atmosphere*, and *Medium and Material*.

- Artistic Style: Include Genre, Era or Historical Style, Cultural and Technological Style.
- Visual Composition and Structure: Include Composition and Layout, Form and Structure, Scale, Movement, Perspective, Pattern and Ornamentation and Detail Level.

Number	Easy		Hard	
	Subject	Modifier	Subject	Modifier
Min.	1	23	1	16
Max.	27	154	24	107
Avg.	8.03	65.00	3.60	43.64

Table 4: Token Statistics for Easy and Hard Benchmarks.

- Aesthetic and Emotional Atmosphere: Include Tone and Atmosphere, Emotional Atmosphere, Lighting and Shadow Effects,
- Medium and Material: Include Medium, Material, Technique, Texture, Surface, Color Palette, Brushwork, Line Quality, Strokes, Layering, Transparency, Opacity and Resolution.

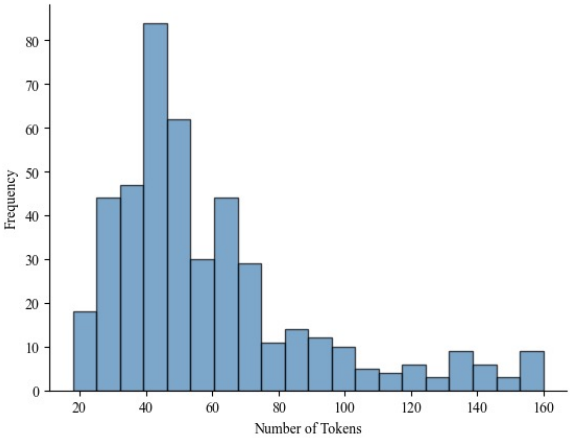


Figure 5: Token frequency distribution of the dataset

C Human Evaluation

We implement a rigorous human evaluation protocol using a blinded manual scoring approach. Each evaluator is presented with the original benchmark images alongside extracted results from all methods, comprising two in-domain and two out-of-domain images per set. To maintain objectivity, evaluators are blinded to the generation methods and conduct their assessments independently, without inter-evaluator communication. The evaluation criteria are differentiated by image category:

- For in-domain data: Evaluators assess both subject matter and stylistic similarity to measure template reproduction fidelity
- For out-of-domain data: Evaluation focuses exclusively on stylistic similarity to assess template generalization capability

Images are rated using a 5-point Likert scale, with higher scores indicating greater similarity. Final results are reported as mean scores across all evaluators. The detailed scoring criteria are presented below.

1. **Completely Different:** The generated image exhibits no discernible similarities to the original, presenting entirely distinct content and stylistic elements.
2. **Barely Similar:** While minimal thematic or elemental commonalities may exist between the original and generated images, they demonstrate significant divergence in both content and stylistic execution.
3. **Somewhat Similar:** The generated image maintains recognizable correspondence to the original's content or subject matter, although notable stylistic variations are present.
4. **Closely Similar:** The generated image demonstrates substantial fidelity to the original's content and subject matter, with only minor compositional variations.
5. **Very Similar:** The generated image achieves near-identical reproduction, maintaining high fidelity to the original's content, style, and intricate details.

D Ablation Comparison

Our extraction template incorporates controllable Subjects and Modifiers, complemented by a flexible Supplements module designed to address potential gaps in subject and modifier extraction. Figure 6 demonstrates the impact of the Supplements module on EvoStealer's effectiveness.

The first case study illustrates how the Supplements module enhances feature detection. While analyzing images individually may cause oversight of shared characteristics—such as the presence of petals in 'a floating umbrella covered in flowers'—the Supplements module successfully captures these overlooked elements in Subject, thereby improving extraction accuracy. In the second case, the module demonstrates its ability to detect visual attributes that are overlooked by predefined modifier categories, such as 'dark yellow tone' within the 'Visual Composition and Structure' categories.

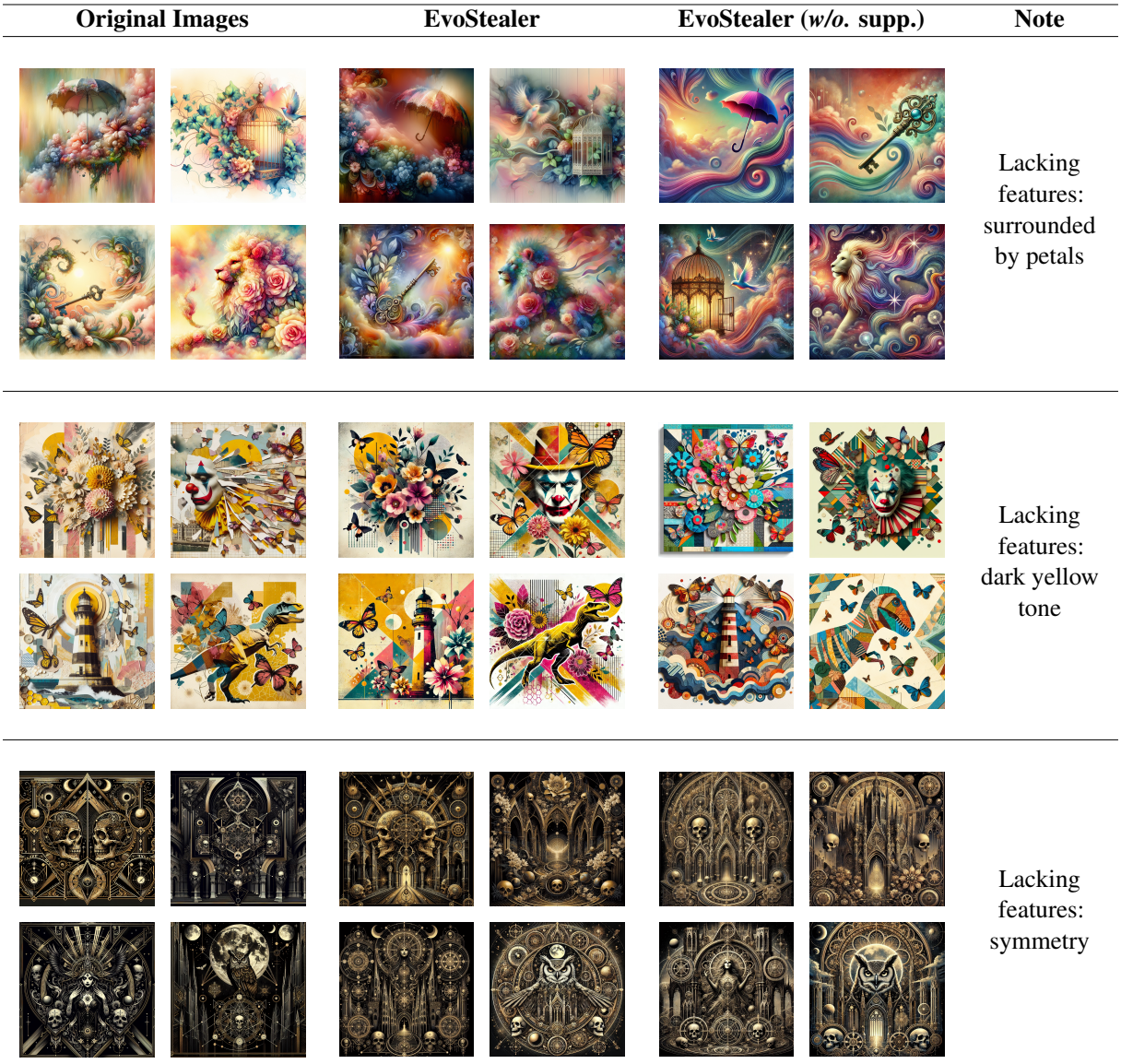


Figure 6: Three examples are used to demonstrate the impact of removing supplements. "w/o. supp." represents the removal of supplements extracted from the pattern.

The third case exemplifies the module's capacity to identify fundamental aesthetic properties like symmetry, which fall outside established modifier categories. These examples highlight how the Supplements module's flexibility enables the detection of additional key features, ultimately enhancing the quality of image generation.

E Evolution Progress

Figure 7 presents the iterative results of EvoStealer on in-domain data across two distinct styles. The figure demonstrates that with each iteration, the generated images progressively converge toward the ground truth style. This progression indicates that EvoStealer successfully refines the quality of style descriptors throughout its iterative process, resulting in images that increasingly approximate the target style. The visual comparison clearly illustrates the algorithm's capacity to incrementally improve stylistic fidelity through successive refinements.

F Cost Estimate

The execution process of EvoStealer comprises three main stages: population initialization, differential evolution (including the fitness function), and image synthesis. We assess the cost from three perspectives: API call frequency, token consumption, and image generation. While API calls and image generation can be accurately and directly measured, token consumption is estimated. Given the instability of the model's output, only the input

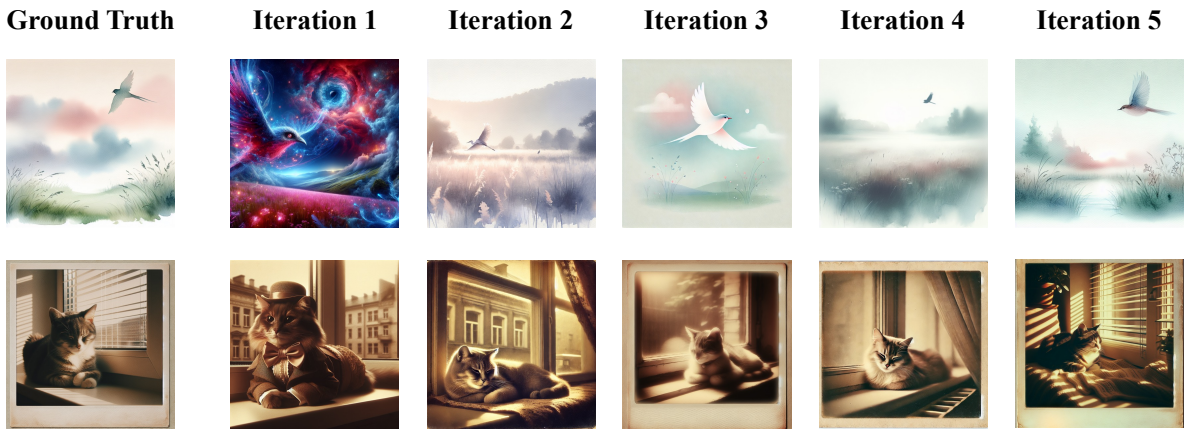


Figure 7: Results of each evolutionary cycle.

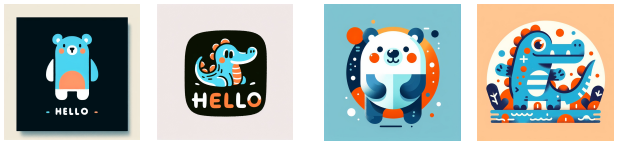
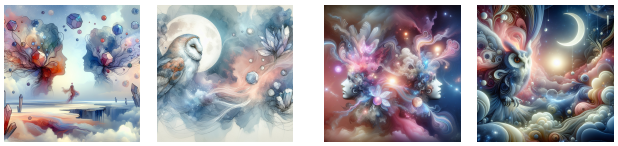
Original Template	Subject	Original Images	*w/o.* supp.
"Act as you are Arkhip Kuindzhi, and you are living during Industrial Revolution time, you have access only to drawing colors and tools available that time. Create image of [subject]."	"cat", "train"		
"[subject], logo, funny children's hand drawn style, doodles, minimalism, cute character, in pastel colors including orange and bright blue on a clean black background, hand writing 'hello' with a bold character underneath."	"bear", "alligator"		
"[subject] / Dreamlike Ethereal Illustrations / Watercolor-Like Techniques / Loose Expressive Brushstrokes / Cool Pastel Shades / Floating Crystalline Structures / Organic Surreal Shapes / Delicate Flowing Line Work / Tranquil Imaginary Worlds"	"Two Heads Are Better Than One", "Moonlit Owl"		

Figure 8: Three failure cases in EvoStealer.

portion is estimated. For this analysis, we evaluate the cost of stealing a prompt template using EvoStealer, based on GPT-4o.

During the population initialization phase, EvoStealer performs two key operations: image element extraction (which generates <subject, modifiers, supplements> triples) and initial template synthesis. On average, this requires 10 calls to GPT-4o, consuming 1.6k tokens, with an estimated cost of approximately $0.04. In the differential evolution phase, EvoStealer performs operations such as difference and commonality identification,

mutation, mutation addition, and crossover. Additionally, for each offspring, template synthesis and image generation are required for both creation and evaluation. On average, this phase involves 125 API calls, consumes 117.5k tokens, and generates 25 images, resulting in a total cost of approximately $1.30. In the image synthesis phase, only the optimal template is used to generate 9 images. This requires 9 API calls and 9 image generations, totaling $0.36. Thus, the overall cost amounts to approximately $1.70.

G Failure Cases

In this section, we will examine several typical failure cases. These failures stem either from the complexity of the images themselves and vague descriptions, or from the inherent limitations of the current EvoStealer method. Figure 8 illustrates representative examples.

A primary limitation is the system's inadequate interpretation of specific artistic styles. Analysis of PromptBase and LaPrompt platforms reveals that many prompt templates incorporate stylistic modifiers, such as "Arshile Gorky style," "Disney style," and "Renaissance style." However, the system struggles to accurately identify and replicate the distinctive characteristics of individual artists' techniques or historical artistic movements, resulting in significant stylistic disparities between generated and source images.

A second limitation concerns text recognition capabilities. The current EvoStealer implementation lacks explicit protocols for extracting textual elements from images. Despite MLLMs' inherent text recognition capabilities, this functionality remains underutilized in the present version—a limitation scheduled for address in future iterations.

The third limitation involves comprehensive detail preservation. When processing images with complex color palettes and rich content, EvoStealer may fail to capture fine-grained features, leading to degraded quality in the resultant prompt templates.