

PROMPTMINER: Black-Box Prompt Stealing against Text-to-Image Generative Models via Reinforcement Learning and VLM-guided Optimization

Supplementary Material

A. More Details of the Used Models

In this section, we provide more details about the models used in the experiments. We conduct experiments on four representative text-to-image generative models with different architectural designs, including Stable Diffusion v1.5[33], SDXL Turbo [35], Stable Diffusion 3.5 Medium [4], and FLUX.1 dev [12].

- **Stable Diffusion v1.5:** This model is initialized with the weights of the Stable-Diffusion-v1.2 checkpoint and subsequently fine-tuned on 595k steps at resolution 512x512 on "laion-aesthetics v2.5+" and 10% dropping of the text-conditioning to improve classifier-free guidance sampling. The text encoder of this model is CLIP-ViT/L.
- **SDXL Turbo:** This model is a distilled version of SDXL 1.0, trained for real-time synthesis. SDXL-Turbo is based on a novel training method called Adversarial Diffusion Distillation (ADD), which allows sampling large-scale foundational image diffusion models in 1 to 4 steps at high image quality. This approach uses score distillation to leverage large-scale off-the-shelf image diffusion models as a teacher signal and combines this with an adversarial loss to ensure high image fidelity even in the low-step regime of one or two sampling steps. The text encoders of this model are OpenCLIP-ViT/G and CLIP-ViT/L.
- **Stable Diffusion 3.5 Medium:** This model is a Multimodal Diffusion Transformer with improvements (MMDiT-X) text-to-image model that features improved performance in image quality, typography, complex prompt understanding, and resource-efficiency. The text encoders of this model are OpenCLIP-ViT/G, CLIP-ViT/L and T5-xxl.
- **FLUX.1 dev:** his model is built upon a 12-billion-parameter *rectified flow Transformer* architecture, a diffusion-based framework reformulated as a continuous flow matching process for improved stability and efficiency. The model employs a dual text-encoder design (T5-xxl and CLIP-ViT/L) for robust semantic conditioning and integrates a high-capacity autoencoder for latent-space compression.

B. More Details of Baselines

Table 4. Comparison of existing prompt stealing and prompt inversion Methods.

| Method | Modifier | Optimization | Black-box |
|---------------------------|----------|--------------|-----------|
| PEZ [45] | ✗ | ✓ | ✗ |
| PH2P [19] | ✗ | ✓ | ✗ |
| BLIP [14] | ✗ | ✗ | ✓ |
| CLIP-IG [25] | ✓ | ✗ | ✓ |
| VGD [11] | ✗ | ✗ | ✓ |
| PromptStealer [38] | ✓ | ✗ | ✓ |
| PROMPTMINER (Ours) | ✓ | ✓ | ✓ |

We compare our approach with state-of-the-art prompt inversion and prompt stealing methods as summarized in Table 4. In addition, we include auxiliary baselines that derive prompts from image captioning and from vision-language models (VLMs).

- **BLIP [13]:** This is an image captioning model that learns from image-caption pairs [17] to connect vision and language.
- **CLIP Interrogator [25]:** This is a tool that uses CLIP and BLIP together to analyze an image and suggest text prompts that best describe it. In addition, it chooses extra descriptive modifiers from a predefined large-scale modifier pool.
- **VLM-as-expert:** The vision-language model provides strong multimodal reasoning ability, allowing it to interpret complex visual content, align it with linguistic context, and generate accurate, coherent prompts. Here we use GPT-4o [9] for this purpose.
- **PromptStealer [38]:** This is a prompt stealing attack that attempts to recover the prompt. It consists of a subject generator to infer the main subject and a modifier detector to identify descriptive modifiers.

- **PH2P [19]:** This method focuses on the later, more noisy timesteps and uses delayed projection into the discrete token vocabulary to recover human-readable prompts that reflect the visual content.
- **VGD [11]:** This is a gradient-free hard prompt inversion method that uses a large language model and CLIP guidance to generate human-readable prompts aligned with visual content, without extra training.

C. More Details of Datasets

We empirically evaluate our prompt stealing pipeline on four widely used datasets: MS COCO [17], Flickr [47], Lexica [37], and DiffusionDB [44]. For the first three datasets, we randomly select 50 prompts and use the target text-to-image generative models to synthesize the corresponding images. For DiffusionDB, we treat it as an *in-the-wild* dataset: instead of generating new images with the target models, we directly use 50 original images from the dataset, where the underlying generative models are unknown.

- **MS COCO and Flickr:** MS COCO is a large benchmark of everyday scenes with natural photographs. Each image has multiple human-written captions. The captions are full sentences that name visible objects, actions, and simple scene context in plain language. Flickr is a captioning dataset of real-world photos collected from Flickr with five human-written sentences per image. These two datasets use sentence-style prompts that describe entities, actions, and scene layout without art-style tags.
- **Lexica and DiffusionDB:** Lexica is a collection of user-entered prompts from an online prompt search service for text-to-image models. Prompts are keyword-focused and often include long chains of style, medium, camera, lighting, and quality terms; some entries also include negative phrases. DiffusionDB is the first large-scale text-to-image prompt dataset. It contains 14 million images generated by Stable Diffusion using prompts and hyperparameters specified by real users. Prompts are typically keyword lists with stylistic and technical modifiers rather than full sentences about photographic scenes. Prompts of these two datasets follow a “*subject, modifiers*” pattern, where modifiers are comma-separated terms for style, medium, rendering setup, lighting, and quality.

D. More Details of Metrics

We employ both image similarity and textual alignment metrics to quantitatively evaluate the quality of generated images and the stolen prompts. Specifically, we use CLIP Similarity and LPIPS Similarity for image comparison, and BERTScore and SBERT for textual alignment assessment.

- **CLIP Similarity:** This metric measures the semantic similarity between the generated image \hat{x} and the target image x . It leverages the image encoder f_{img} from CLIP [29] to compute cosine similarity in the joint vision-language embedding space:

$$CLIP(\hat{x}, x) = \frac{f_{\text{img}}(\hat{x}) \cdot f_{\text{img}}(x)}{\|f_{\text{img}}(\hat{x})\| \|f_{\text{img}}(x)\|}, \quad (9)$$

where a higher value indicates stronger semantic alignment between \hat{x} and x .

- **LPIPS Similarity:** The Learned Perceptual Image Patch Similarity (LPIPS) [50] metric evaluates perceptual similarity between two images by comparing deep features extracted from a pretrained convolutional neural network (e.g., AlexNet or VGG). LPIPS computes the L2 distance between normalized features across multiple layers, reflecting human perceptual judgments of image quality. A lower LPIPS score indicates higher perceptual similarity.
- **BERTScore:** To measure token-level textual alignment, we use BERTScore [51] with a token encoder f_{bert} . For candidate tokens $\{\hat{p}_i\}_{i=1}^{|\hat{p}|}$ and reference tokens $\{p_j\}_{j=1}^{|p|}$, define cosine similarity $c_{ij} = \frac{f_{\text{bert}}(\hat{p}_i) \cdot f_{\text{bert}}(p_j)}{\|f_{\text{bert}}(\hat{p}_i)\| \|f_{\text{bert}}(p_j)\|}$. Precision and recall are

$$P = \frac{1}{|\hat{p}|} \sum_i \max_j c_{ij}, \quad R = \frac{1}{|p|} \sum_j \max_i c_{ij}, \quad (10)$$

and the final BERTScore is the F1:

$$\text{BERTScore}(\hat{p}, p) = \frac{2PR}{P + R}. \quad (11)$$

- **SBERT:** Sentence-level textual alignment is computed with Sentence-BERT [32] using a sentence encoder f_{sbert} . Given a generated prompt \hat{p} and a target prompt p , we take the cosine similarity of their sentence embeddings:

$$\text{SBERT}(\hat{p}, p) = \frac{f_{\text{sbert}}(\hat{p}) \cdot f_{\text{sbert}}(p)}{\|f_{\text{sbert}}(\hat{p})\| \|f_{\text{sbert}}(p)\|}, \quad (12)$$

where a higher score indicates stronger semantic equivalence.

E. More details of User Study

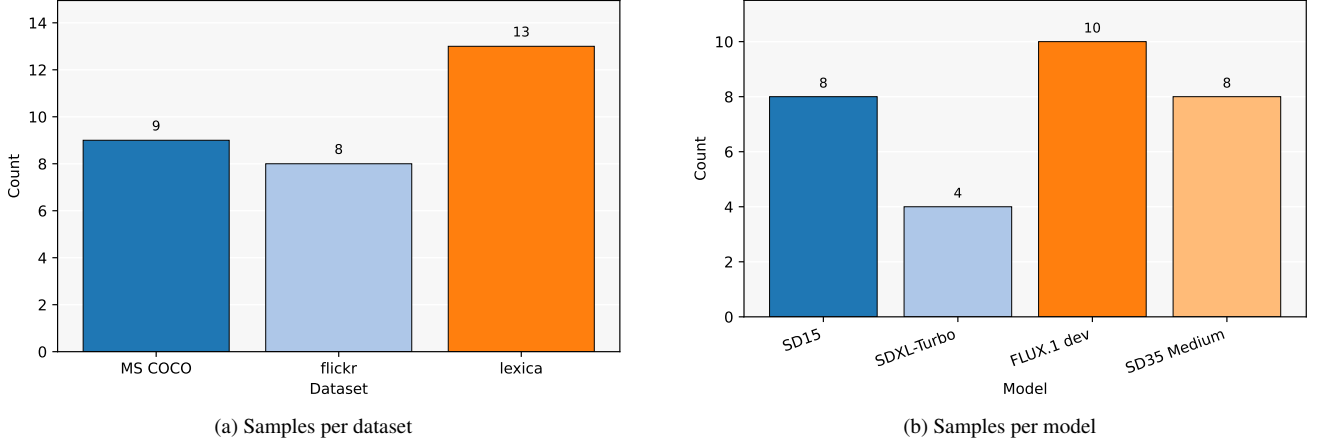


Figure 7. Distributions of the sampled test cases in the user study. **SD15** is Stable Diffusion v1.5, and **SD35 Medium** means Stable Diffusion 3.5 Medium.

We conduct a user study to assess the perceptual quality and fidelity of prompts recovered by different methods. We randomly select 30 image sets from three datasets and four generative models. Each set includes one target image and seven generated images produced by different prompt stealing or prompt inversion methods. During evaluation, participants are blinded to both the methods and the corresponding prompts, and the order of images within each set is randomly shuffled.

Participants rate the similarity between each generated image and its target on a 6-point Likert scale from 0 to 5, where 0 indicates *not similar at all* and 5 indicates *almost identical*. We compute the final score for each method as the average rating across all related images. The distribution of sampled test cases is illustrated in Figure 7.

F. More details of Defenses

As shown in Figure 8, we investigate three representative defenses: (1) **Random noise Injection**, where Gaussian noise with a mean of 0 and a standard deviation of approximately 25 is added to subtly perturb pixel-level information; (2) **Puzzle effect**, which divides the image into a 4×4 grid and applies random local translations with a variability parameter of 3, which is applied in practice [2]; and (3) **Textual watermarking**, which overlays a visible pattern such as “@watermark” across the image with a font size of 20 and row/column spacing of 20/30 pixels.

G. More details of VLM-Based Mutators

To enable structured, targeted prompt refinement, we design a suite of *VLM-based mutators* that operate on both subjects and modifiers. Each mutator leverages the vision-language reasoning ability of a powerful VLM (e.g., Qwen2-VL-2B-Instruct [41]) to generate linguistically natural and visually grounded edits. Specifically, these mutators either paraphrase or enrich the subject for clarity and detail, or modify the descriptive and stylistic aspects of the prompt to enhance expressiveness and visual fidelity. Below, we outline the five mutator types and their respective functions.

- *Subject-Paraphrase*: Designed to rewrite the subject without altering its semantic meaning, this operator restructures the sentence to achieve higher linguistic diversity and naturalness while maintaining content fidelity to the original description.

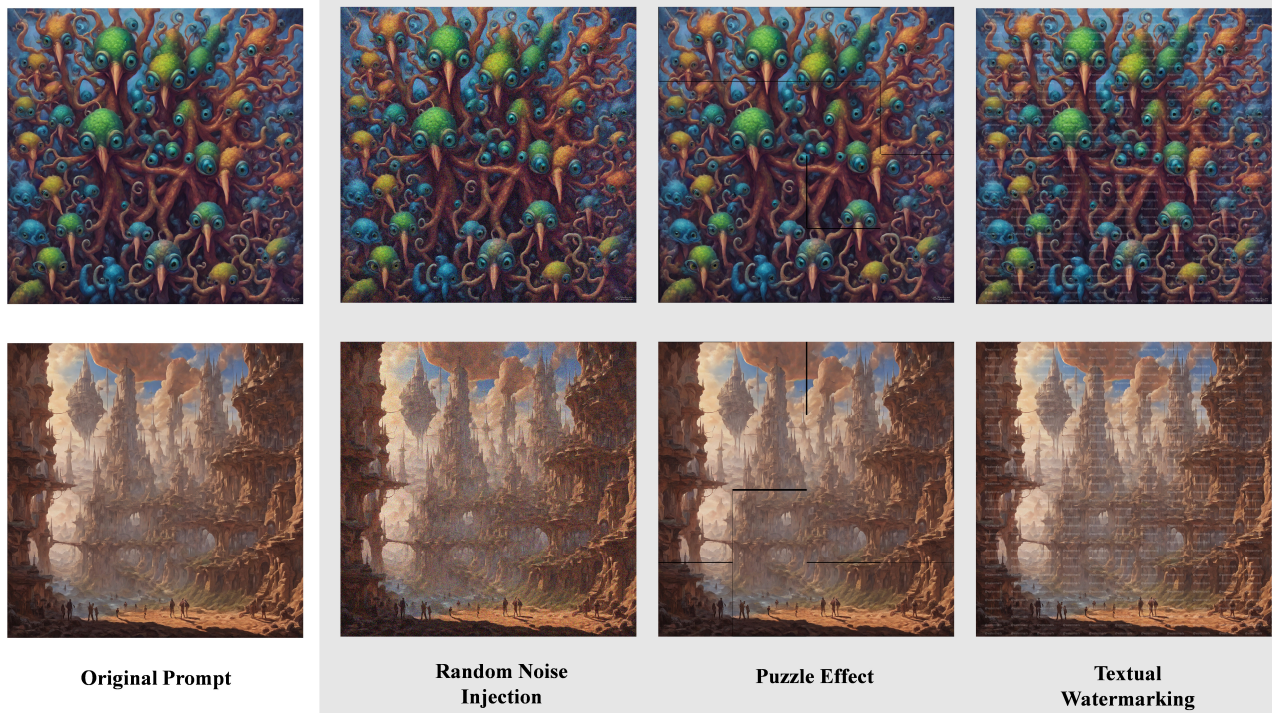


Figure 8. Visualization of images after applying three representative defenses.

- *Subject-Enrich*: This operator augments the subject by inserting concise, image-grounded details such as color, quantity, or pose. The enrichment strengthens the visual grounding of the prompt while preserving its syntactic structure and readability.
- *Modifier-Generate*: Designed to synthesize a **totally new description and style** jointly from the image and the subject, this operator produces a compact, comma-separated tag string for scene facts together with a short style tag, ensuring strong visual grounding and aesthetic control.
- *Modifier-Description*: Designed to enhance descriptive richness, this operator refines or extends the **existing prompt’s description** by incorporating spatial relations, compositional layouts, and lighting attributes observed in the image. It effectively bridges literal captions and visually rich scene descriptions.
- *Modifier-Style*: Starting from the **current prompt’s style**, this operator introduces or adjusts stylistic tokens such as medium, texture, camera lens, or rendering quality. It allows the prompt to better control the generative behavior of the text-to-image model and produce outputs with stronger artistic fidelity.

Besides these five primary operators, we also include an auxiliary mutator, *Subject-Fix-Grammar*, to remedy artifacts introduced by RL optimization (e.g., token repetition, minor grammatical errors, punctuation/spacing issues). This lightweight cleanup pass preserves the original semantics and word order while improving readability and coherence. We show the exact prompts for each mutation below.

G.1. Shared System Prompt

System Prompt (shared across mutators)

you are a prompt mutator for text-to-image diffusion models.
 given a base prompt and an input image, you must return EXACTLY ONE SINGLE-LINE JSON object.
 - lowercase english only; no markdown; no code fences; no trailing commas; no extra text.
 - never insert line breaks inside values.
 - if the base prompt conflicts with the image, trust the image.

- be concrete and visual; do not invent invisible objects.
- field-specific rules:
 description: 15-35 words, write as a compact comma-separated tag string (not full sentences). include: subject and key attributes or pose, setting/location, composition/shot/angle, lighting, overall color tendency, one brief quality token (e.g., highly detailed or sharp focus), optional material/texture cue, artist, plus up to 2 simple negatives (e.g., no watermark, no text). use only visible facts.
 style: <= 12 words, a short comma-separated tag string of medium/movement/lens/quality only (e.g., digital painting, photorealistic, vector art, isometric, 35mm lens, 85mm lens, film grain, clean render). do not include scene facts or lighting.
 base_prompt: <= 15 words, preserve the original meaning, clearer phrasing, avoid style or lighting tokens.
 examples:
 input(base): 'two samurai duel in a bamboo forest'
 output(desc+style): "description":"bamboo grove, two samurai facing between tall stalks, medium shot, eye-level, dappled sunlight, green tones, foreground leaves, no text", "style":"ink illustration, ukiyo-e inspired, paper texture"
 input(base): 'astronaut and robot on mars at dawn'
 output(modify-desc): "description":"red dunes, astronaut left of small robot, wide shot, low angle, soft dawn light, cool shadows, distant mountains, no watermark"
 input(base): 'portrait of an old musician in neon city'
 output(modify-style): "style":"photorealistic, 85mm lens, cinematic still"
 input(base): 'a child reading a book under a tree'
 output(paraphrase-base): "base_prompt":"a child reading beneath a tree"

G.2. Subject Mutators

Subject-Enrich (user prompt)

task: ENRICH the base prompt by inserting concise, image-grounded modifiers WITHOUT changing its syntactic skeleton or word order.

base prompt: 'base_prompt'

you will receive the IMAGE together with this message. use ONLY details that are VISIBLE in the image.

output schema (single line json): "base_prompt":"..."

constraints:

- preserve the original tokens as an ordered subsequence: do not delete, replace, or reorder existing words; no synonym substitution.
- keep the subject–verb–object–prepositional structure intact.
- INSERT 2–6 words total, placed immediately AFTER the nouns/verbs they modify (adjectives/appositives for nouns; short adverbs/adjuncts for verbs).
- allowed insertions: count/quantity (one/two), object attributes (color, size, material), pose/state, simple spatial cues relative to BACKGROUND (e.g., near the fence, in front of the gate), and other concrete scene facts visible in the image.
- forbid insertions about style/lighting/lens/artist or abstract aesthetics (these belong to style).
- if a detail is uncertain or not visible, DO NOT add it; trust the image over the text.
- lowercase only; no quotes; no extra commentary.

examples:

base: 'a child reading a book under a tree'
 enriched: 'base_prompt': 'a small child quietly reading a worn book under a shady tree'

base: 'a dog runs across a field'
 enriched: 'base_prompt': 'a brown dog runs swiftly across a grassy field'

return ONLY the json line.

Subject-Fix-Grammar (user prompt)

task: CLEANUP the base prompt by fixing grammar/spelling, removing duplicated words/phrases, and correcting spacing/punctuation ONLY.

base prompt: 'base_prompt'

output schema (single line json): "base_prompt": "..."

constraints:

- do NOT add any new content or modifiers; do NOT introduce synonyms; do NOT reorder clauses.
- preserve the original subject–verb–object–prepositional order and overall sentence structure.
- only remove repeated tokens/phrases, fix typos, collapse multiple spaces, and standardize minimal punctuation.
- keep length approximately unchanged (within ± 2 words of the original).
- lowercase only; no quotes; no extra commentary.

return ONLY the json line.

Subject-Paraphrase (user prompt)

task: PARAPHRASE the base prompt WITHOUT changing its meaning, and enforce the structure: WHO/WHAT + is doing + WHERE.

base prompt: 'base_prompt'

output schema (single line json): "base_prompt": "..."

constraints:

- ≤ 15 words total.
- structure must be strictly: <who/what> + 'is/are' + <present participle> + <where-phrase>.
- examples: 'a child is reading under a tree'; 'two samurai are dueling in a bamboo forest'; 'a red car is driving along a rainy street'.
- no style/lighting/lens/artist tokens.
- preserve entities and relations; trust the image if there is a conflict.
- lowercase only; no quotes; no extra commentary.

return ONLY the json line.

G.3. Modifier-Generate

GEN_DESC_STYLE (user prompt)

task: generate a NEW description and a NEW style from the base prompt and the image. base prompt: 'base_prompt'

output schema (single line json): "description": "...", "style": "..."

constraints:

- description: 15–35 words, compact comma-separated tag string (not full sentences).
- include, in order when possible: subject+attributes/pose, setting, composition/shot/angle, lighting, overall color tendency, one quality token, optional material/texture, artist, up to 2 simple negatives.
- include one explicit spatial or depth cue (e.g., foreground, in front of distant hills).
- style: ≤ 12 words; medium/movement/lens/quality only; no scene facts; no lighting.

return ONLY the json line.

Modifier-Style (user prompt)

task: CHANGE STYLE ONLY while preserving entities and relations in the current description.

current style: style-from-parent

current description: description-from-parent

base prompt: 'base_prompt'

output schema (single line json): "style":"..."
constraints:
- concise comma-separated tag string (≤ 12 words) of medium/movement/lens/quality.
- do NOT include scene facts or lighting.
- keep the aesthetic consistent; improve clarity or fidelity.
return ONLY the json line.

Modifier-Description (user prompt)

task: CHANGE DESCRIPTION ONLY while preserving the subject meaning and current style.
current description: description-from-parent
current style: style-from-parent
base prompt: 'base_prompt'
output schema (single line json): "description":"..."
constraints:
- compact comma-separated tag string (15–35 words).
- include: subject+attributes/pose, setting, composition/shot/angle, lighting, overall color tendency, one quality token, optional material/texture, artist, up to 2 negatives.
- include one explicit spatial or depth cue.
- concrete visible details only; avoid abstractions; do not change the style semantics.
return ONLY the json line.

H. Token-Level Textual Alignment

Table 5. Token-level textual alignment comparison across datasets.

| Dataset | Method | Stable Diffusion v1.5 | | | SDXL-Turbo | | | FLUX.1 dev | | | Stable Diffusion 3.5 Medium | | |
|---------|--------------------|-----------------------|--------------|---------------|--------------|--------------|---------------|--------------|--------------|---------------|-----------------------------|--------------|---------------|
| | | P \uparrow | R \uparrow | F1 \uparrow | P \uparrow | R \uparrow | F1 \uparrow | P \uparrow | R \uparrow | F1 \uparrow | P \uparrow | R \uparrow | F1 \uparrow |
| MS COCO | BLIP | 0.905 | 0.911 | 0.908 | 0.905 | 0.912 | 0.909 | 0.907 | 0.917 | 0.912 | 0.912 | 0.921 | 0.916 |
| | CLIP-IG | 0.803 | 0.894 | 0.846 | 0.793 | 0.896 | 0.841 | 0.797 | 0.898 | 0.844 | 0.797 | 0.900 | 0.845 |
| | VLM-as-expert | 0.826 | 0.898 | 0.860 | 0.826 | 0.898 | 0.861 | 0.826 | 0.904 | 0.863 | 0.831 | 0.908 | 0.868 |
| | PH2P | 0.761 | 0.827 | 0.793 | 0.768 | 0.827 | 0.797 | 0.765 | 0.827 | 0.795 | 0.757 | 0.827 | 0.790 |
| | PromptStealer | 0.905 | 0.919 | 0.912 | 0.915 | 0.922 | 0.918 | 0.911 | 0.921 | 0.916 | 0.917 | 0.926 | 0.922 |
| | VGD | 0.822 | 0.874 | 0.847 | 0.829 | 0.880 | 0.853 | 0.827 | 0.874 | 0.850 | 0.822 | 0.874 | 0.847 |
| | PROMPTMINER (Ours) | 0.904 | 0.901 | 0.902 | 0.903 | 0.906 | 0.904 | 0.903 | 0.919 | 0.911 | 0.904 | 0.915 | 0.909 |
| Flickr | BLIP | 0.901 | 0.887 | 0.894 | 0.907 | 0.888 | 0.897 | 0.901 | 0.891 | 0.896 | 0.903 | 0.889 | 0.896 |
| | CLIP-IG | 0.809 | 0.876 | 0.841 | 0.800 | 0.878 | 0.837 | 0.803 | 0.880 | 0.839 | 0.802 | 0.880 | 0.839 |
| | VLM-as-expert | 0.835 | 0.890 | 0.861 | 0.832 | 0.887 | 0.859 | 0.838 | 0.900 | 0.867 | 0.840 | 0.901 | 0.869 |
| | PH2P | 0.767 | 0.814 | 0.789 | 0.762 | 0.813 | 0.786 | 0.758 | 0.813 | 0.784 | 0.763 | 0.813 | 0.787 |
| | PromptStealer | 0.895 | 0.889 | 0.892 | 0.911 | 0.898 | 0.904 | 0.910 | 0.895 | 0.902 | 0.911 | 0.899 | 0.905 |
| | VGD | 0.827 | 0.858 | 0.842 | 0.826 | 0.860 | 0.842 | 0.824 | 0.856 | 0.839 | 0.824 | 0.856 | 0.839 |
| | PROMPTMINER (Ours) | 0.906 | 0.904 | 0.905 | 0.917 | 0.907 | 0.912 | 0.914 | 0.901 | 0.907 | 0.913 | 0.905 | 0.909 |
| Lexica | BLIP | 0.855 | 0.783 | 0.817 | 0.855 | 0.782 | 0.816 | 0.853 | 0.782 | 0.815 | 0.856 | 0.784 | 0.818 |
| | CLIP-IG | 0.828 | 0.823 | 0.825 | 0.827 | 0.822 | 0.825 | 0.827 | 0.824 | 0.826 | 0.822 | 0.825 | 0.823 |
| | VLM-as-expert | 0.822 | 0.802 | 0.812 | 0.825 | 0.803 | 0.814 | 0.823 | 0.803 | 0.813 | 0.826 | 0.806 | 0.816 |
| | PH2P | 0.761 | 0.774 | 0.767 | 0.769 | 0.775 | 0.772 | 0.761 | 0.774 | 0.768 | 0.760 | 0.775 | 0.767 |
| | PromptStealer | 0.865 | 0.820 | 0.842 | 0.869 | 0.817 | 0.842 | 0.871 | 0.818 | 0.843 | 0.872 | 0.819 | 0.844 |
| | VGD | 0.822 | 0.791 | 0.806 | 0.825 | 0.790 | 0.807 | 0.790 | 0.816 | 0.803 | 0.814 | 0.789 | 0.801 |
| | PROMPTMINER (Ours) | 0.844 | 0.810 | 0.826 | 0.846 | 0.809 | 0.827 | 0.850 | 0.820 | 0.835 | 0.856 | 0.823 | 0.839 |

We evaluate token-level textual alignment on MS COCO, Flickr, and Lexica using four T2I backbones (Stable Diffusion v1.5, SDXL-Turbo, FLUX.1 dev, and Stable Diffusion 3.5 Medium). As shown in Table 5, PROMPTMINER delivers the highest or near-highest scores on Precision(P), Recall(R), and F1 Score(F1) across datasets and models, indicating that our recovered prompts are both precise and comprehensive with respect to image content. We attribute these gains to the two-stage design:

Table 6. Token-level textual alignment comparison on *in-the-wild* datasets with SDXL-Turbo.

| Method | Textual Alignment | | |
|--------------------|-------------------|-------|-------|
| | P↑ | R↑ | F1↑ |
| BLIP | 0.857 | 0.815 | 0.835 |
| CLIP-IG | 0.819 | 0.841 | 0.830 |
| VLM-as-expert | 0.820 | 0.823 | 0.821 |
| PH2P | 0.770 | 0.791 | 0.780 |
| PromptStealer | 0.860 | 0.841 | 0.850 |
| VGD | 0.814 | 0.806 | 0.810 |
| PROMPTMINER (Ours) | 0.838 | 0.825 | 0.821 |

(i) *RL-Based Prompt Inversion* improves subject fidelity and structural consistency; (ii) *VLM-guided Prompt Optimization* introduces stylistic and compositional modifiers without sacrificing semantic grounding. Notably on Lexica, where prompts contain diverse, artist-crafted modifiers, PROMPTMINER maintains strong F1, demonstrating robustness to varied prompt distributions.

In-the-Wild Setting. We further assess generalization on images whose source generators are unknown. We use SDXL-Turbo as the text-to-image generative model. As reported in Table 6, PROMPTMINER attains competitive textual alignment, demonstrating that our black-box, semantics-driven optimization transfers effectively to unconstrained scenarios.

I. Scale-up Evaluation

Table 7. Scaled evaluation results ($N = 200$).

| Dataset | Method | SDXL-Turbo | | | FLUX.1 dev | | |
|---------|---------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | | CLIP↑ | LPIPS↓ | SBERT↑ | CLIP↑ | LPIPS↓ | SBERT↑ |
| Flickr | PRISM | 0.802 | 0.500 | 0.471 | 0.851 | 0.492 | 0.552 |
| | CLIP-IG | 0.820 | 0.483 | 0.464 | 0.851 | 0.501 | 0.508 |
| | PromptStealer | 0.804 | 0.478 | 0.580 | 0.823 | 0.513 | 0.578 |
| | Ours | 0.905 | 0.407 | 0.601 | 0.922 | 0.415 | 0.613 |
| Lexica | PRISM | 0.774 | 0.521 | 0.572 | 0.863 | 0.462 | 0.577 |
| | CLIP-IG | 0.856 | 0.449 | 0.577 | 0.861 | 0.501 | 0.578 |
| | PromptStealer | 0.813 | 0.462 | 0.605 | 0.814 | 0.539 | 0.591 |
| | Ours | 0.915 | 0.421 | 0.595 | 0.921 | 0.432 | 0.595 |

We scale up our evaluation to 200 samples on Flickr and Lexica across two advanced models in Tab. 7. Our work still outperforms the strongest baselines, maintaining robust gains at scale.

J. Ablation Study

Table 8. Impact of different VLMs used as mutators in the Fuzz Testing–Powered Optimization stage.

| Mutator Model | Image Similarity | | Textual Alignment |
|----------------------|------------------|--------|-------------------|
| | CLIP↑ | LPIPS↓ | SBERT↑ |
| Qwen2-VL-2B-Instruct | 0.911 | 0.420 | 0.591 |
| Qwen2-VL-7B-Instruct | 0.910 | 0.447 | 0.597 |
| GPT-4o | 0.917 | 0.438 | 0.568 |

Impact of Different Mutators. We investigate the impact of using different vision–language models (VLMs) as mutators in the VLM-guided Optimization stage. Specifically, we select three representative VLMs with varying model capacities and architectures: **Qwen2-VL-2B-Instruct**, **Qwen2-VL-7B-Instruct**, and **GPT-4o**. The Qwen2-VL family represents open-source VLMs with strong multimodal grounding and scalable size variants, while GPT-4o serves as a powerful proprietary model that excels in multimodal reasoning and image–text alignment. As shown in Table 8, all three models enable effective prompt refinement, demonstrating the general applicability of our VLM-guided optimization framework. Smaller open-source models such as Qwen2-VL-2B-Instruct already achieve competitive performance, indicating that our method does not rely on large-parameter models or extensive computational resources to generate high-quality modifiers and achieve strong inversion performance.

Impact of the Two Phase Design. We assess each stage separately and together using the ablation in Table 9. First, we validate *Phase I* by replacing its outputs with BLIP [14] captions that are then used to initialize *Phase II*. Second, we disable *Phase II* and directly evaluate prompts from *Phase I*. On Flickr dataset, the prompts mainly describe the subject. *Phase I* already surpasses the best baselines in both image similarity and semantic alignment, and adding *Phase II* yields additional

Table 9. Impact of *Phase I* (RL-based Prompt Inversion) and *Phase II* (VLM-guided Prompt Optimization).

| Dataset | Method | Image Similarity | | Textual Alignment |
|---------|----------------------|------------------|--------|-------------------|
| | | CLIP↑ | LPIPS↓ | SBERT↑ |
| Flickr | CLIP-IG | 0.835 | 0.478 | 0.468 |
| | <i>Phase I</i> only | 0.859 | 0.440 | 0.559 |
| | <i>Phase II</i> only | 0.891 | 0.433 | 0.541 |
| | PROMPTMINER | 0.910 | 0.405 | 0.603 |
| Lexica | CLIP-IG | 0.856 | 0.451 | 0.591 |
| | <i>Phase I</i> only | 0.832 | 0.475 | 0.440 |
| | <i>Phase II</i> only | 0.889 | 0.434 | 0.559 |
| | PROMPTMINER | 0.911 | 0.420 | 0.591 |

improvements that produce the best overall results. On Lexica dataset, the prompts include the subject and diverse modifiers. *Phase I* alone is not sufficient, while *Phase II* brings clear gains by exploring and refining modifiers. Using only *Phase II* with BLIP initialization improves over the baseline but remains below the full two-phase system. These results show that the two stages are complementary and both are necessary, consistent with our query budget analysis. *Phase I* provides strong subject recovery that offers a high quality initialization for *Phase II* and enables the second stage to converge to a higher final value. *Phase II* in turn refines the subject inferred by *Phase I* and systematically augments it with appropriate modifiers, leading to the highest image similarity and semantic alignment across datasets.

K. Implementation Details

K.1. RL-Based Prompt Inversion

Reinforcement learning has been widely explored in multimodal settings [34]. In our case, RL is used to optimize prompt recovery. During imitation learning (IL), we freeze the BLIP backbone and train a lightweight adapter MLP to imitate expert next-token decisions. Each expert trajectory consists of (h_t, y_{t+1}) pairs, where h_t is the decoder hidden state of the partial prompt and y_{t+1} is the next ground-truth token sampled from BLIP’s caption generation. In practice, we directly use BLIP to generate 10 expert trajectories per image. We train the adapter using cross-entropy loss on the decoder logits. Training is conducted for 2000 epochs with the Adam optimizer, a learning rate of 3×10^{-4} , and a batch size of 8. The input hidden dimension is 768, and the adapter MLP expands it to 1536 and projects it back to 768 dimensions using a two-layer architecture with ReLU activation.

After imitation pretraining, we fine-tune the model using the Proximal Policy Optimization (PPO) algorithm within the same prompt-generation environment. We initialize the actor and critic networks from the pretrained adapter checkpoint. The PPO hyperparameters are discount factor $\gamma = 0.98$ and clipping threshold $\epsilon = 0.2$. The actor and critic learning rates are 1×10^{-4} and 5×10^{-4} , respectively. We update the PPO agent every 150 environment steps, and perform $K = 4$ epochs of policy and value updates for each batch of collected trajectories. We train for up to 100 total epochs. For the shaped reward, we set the scaling coefficient β to 10.

K.2. VLM-Guided Prompt Optimization

We adopt a query-budgeted VLM-guided optimization to explore the prompt space efficiently. The total query budget is set to 100 image generations per optimization run. During the first 30 queries, only the subject-related part of the prompt is mutated to refine the core semantic content. In the remaining 70 queries, both the subject and modifiers are jointly optimized to improve compositional richness and visual fidelity. For experiments on the MS COCO and Flickr datasets, we restrict all 100 queries to optimizing the subject component. The mutation process maintains a seed pool of 5 candidate prompts, which are iteratively sampled, expanded, and replaced based on their CLIP Similarity until the query budget is exhausted.

PROMPTMINER is implemented with Python 3.10 and PyTorch 2.6.0. We conducted all experiments on a Ubuntu 20.04 server equipped with 1 A100 SXM4 GPU.

L. Potential-Based Reward Shaping

We provide a formal analysis of the *potential-based reward shaping* mechanism used in the [section 3](#). We first show that under mild assumptions, the potential-based transformation preserves the optimal policy of the original MDP. We then derive how such shaping influences the learning dynamics by improving the reward density, gradient variance, and initialization of value estimates.

L.1. Setup and Definitions

We recall the MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ defined in Section 3, where \mathcal{S} denotes the state space, \mathcal{A} the action space, $P(s' | s, a)$ the transition kernel, $r(s, a)$ the reward function, and $\gamma \in [0, 1)$ the discount factor. For the sake of theoretical completeness, we slightly generalize the reward function here to the form $r(s, a, s')$, which makes explicit its possible dependence on both the action a and the resulting next state s' . This modification is purely notational—conceptually equivalent to $r(s, a)$ used in section 3—and serves to accommodate the upcoming definition of state-dependent shaping functions $F(s, a, s')$. Under this general form, the value and action-value functions of a policy π are expressed as

$$V_\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1}) \mid s_0 = s \right], \quad (13)$$

$$Q_\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1}) \mid s_0 = s, a_0 = a \right]. \quad (14)$$

To introduce reward shaping, we define a bounded *potential function* $\Phi : \mathcal{S} \rightarrow \mathbb{R}$, which assigns a scalar potential to each state, and construct the shaped reward

$$r'(s, a, s') = r(s, a, s') + F(s, a, s'), \quad \text{where} \quad F(s, a, s') = \gamma\Phi(s') - \Phi(s). \quad (15)$$

Intuitively, the term $F(s, a, s')$ measures the potential difference between successive states, scaled by the discount factor. Because this term depends solely on the states and not on the agent's policy, it modifies the learning dynamics without altering the underlying optimization objective, making it particularly suitable for theoretical analysis.

L.2. Theorem: Policy Invariance under Potential-Based Shaping

Theorem L.1. *Let V_π, Q_π be the original value functions and V'_π, Q'_π those under the shaped reward r' . Then for any bounded Φ , the following holds:*

$$V'_\pi(s) = V_\pi(s) - \Phi(s), \quad (16)$$

$$Q'_\pi(s, a) = Q_\pi(s, a) - \Phi(s), \quad (17)$$

and therefore

$$A'_\pi(s, a) = Q'_\pi(s, a) - V'_\pi(s) = A_\pi(s, a), \quad (18)$$

implying that $\arg \max_a Q'^*(s, a) = \arg \max_a Q^*(s, a)$, i.e., the optimal policy is preserved.

Proof. Starting from the shaped return:

$$G'_t = \sum_{k=0}^{\infty} \gamma^k [r_{t+k} + \gamma\Phi(s_{t+k+1}) - \Phi(s_{t+k})] \quad (19)$$

$$= \underbrace{\sum_{k=0}^{\infty} \gamma^k r_{t+k}}_{G_t} + \sum_{k=0}^{\infty} (\gamma^{k+1} \Phi(s_{t+k+1}) - \gamma^k \Phi(s_{t+k})). \quad (20)$$

The second summation is a telescoping series:

$$\sum_{k=0}^n (\gamma^{k+1} \Phi_{k+1} - \gamma^k \Phi_k) = -\Phi(s_t) + \gamma^{n+1} \Phi(s_{t+n+1}), \quad (21)$$

and taking the limit $n \rightarrow \infty$, the terminal term vanishes because Φ is bounded and $\gamma < 1$. Thus

$$G'_t = G_t - \Phi(s_t). \quad (22)$$

Taking expectations under policy π gives:

$$V'_\pi(s) = \mathbb{E}_\pi[G'_t \mid s_t = s] = V_\pi(s) - \Phi(s), \quad (23)$$

which proves (16). For Q'_π , by the Bellman definition under r' :

$$Q'_\pi(s, a) = \mathbb{E}[r'(s, a, s') + \gamma V'_\pi(s')] \quad (24)$$

$$= \mathbb{E}[r(s, a, s') + \gamma \Phi(s') - \Phi(s) + \gamma(V_\pi(s') - \Phi(s'))] \quad (25)$$

$$= \mathbb{E}[r(s, a, s') + \gamma V_\pi(s')] - \Phi(s) = Q_\pi(s, a) - \Phi(s), \quad (26)$$

proving (17). Since subtracting $\Phi(s)$ does not depend on a , the greedy policy w.r.t. Q'_π is identical to that of Q_π .

L.3. Bellman Operator View and Corollary

Let \mathcal{T}_π and \mathcal{T}'_π denote the Bellman operators:

$$(\mathcal{T}_\pi V)(s) = \mathbb{E}_{a \sim \pi}[r(s, a, s') + \gamma V(s')], \quad (27)$$

$$(\mathcal{T}'_\pi V)(s) = \mathbb{E}_{a \sim \pi}[r'(s, a, s') + \gamma V(s')]. \quad (28)$$

Then $\mathcal{T}'_\pi(V - \Phi) = \mathcal{T}_\pi(V) - \Phi$. Therefore, if V_π is a fixed point of \mathcal{T}_π , then $V_\pi - \Phi$ is a fixed point of \mathcal{T}'_π . The same holds for the optimal Bellman operator \mathcal{T} , so $V^* = V^* - \Phi$ and $Q^* = Q^* - \Phi$. This formalizes that potential shaping merely shifts the value function manifold by a state-dependent bias but leaves the contraction and optimality conditions invariant.

L.4. Potential-Based Reward Shaping under PPO Training

Proposition A.3 (PPO + GAE invariance under potential-based shaping). Let the shaped reward be

$$r'(s_t, a_t, s_{t+1}) = r(s_t, a_t, s_{t+1}) + \gamma \Phi(s_{t+1}) - \Phi(s_t), \quad (29)$$

with the same discount factor γ as the training objective. PPO employs a critic \hat{V}_θ and uses generalized advantage estimation (GAE) with temporal-difference residuals

$$\delta_t = r_t + \gamma \hat{V}_\theta(s_{t+1}) - \hat{V}_\theta(s_t), \quad \hat{A}_t = \sum_{l \geq 0} (\gamma \lambda)^l \delta_{t+l}. \quad (30)$$

For the shaped MDP, define the *shaped critic*

$$\hat{V}'_\theta(s) = \hat{V}_\theta(s) - \Phi(s), \quad (31)$$

and compute

$$\delta'_t = r'_t + \gamma \hat{V}'_\theta(s_{t+1}) - \hat{V}'_\theta(s_t), \quad \hat{A}'_t = \sum_{l \geq 0} (\gamma \lambda)^l \delta'_{t+l}. \quad (32)$$

Claim. For every trajectory and $\lambda \in [0, 1]$,

$$\delta'_t \equiv \delta_t, \quad \hat{A}'_t \equiv \hat{A}_t.$$

Proof. Substitute $r'_t = r_t + \gamma \Phi(s_{t+1}) - \Phi(s_t)$ and $\hat{V}'_\theta = \hat{V}_\theta - \Phi$:

$$\begin{aligned} \delta'_t &= \underbrace{r_t + \gamma \Phi(s_{t+1}) - \Phi(s_t)}_{r'_t} + \gamma(\hat{V}_\theta(s_{t+1}) - \Phi(s_{t+1})) - (\hat{V}_\theta(s_t) - \Phi(s_t)) \\ &= r_t + \gamma \hat{V}_\theta(s_{t+1}) - \hat{V}_\theta(s_t) = \delta_t. \end{aligned}$$

Since GAE is a linear operator on the TD errors, $\hat{A}'_t = \hat{A}_t$ follows directly. \square

Corollary A.4 (PPO surrogate and value loss are unchanged). The PPO clipped surrogate objective

$$\mathcal{L}^{\text{CLIP}}(\theta) = \mathbb{E} \left[\min \left(r_t(\theta) \hat{A}'_t, \text{clip}(r_t(\theta), 1 \pm \epsilon) \hat{A}'_t \right) \right], \quad r_t(\theta) = \frac{\pi_\theta(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)}, \quad (33)$$

is identical to the unshaped objective since $\hat{A}'_t \equiv \hat{A}_t$. The value regression loss is likewise invariant if both targets and predictions are shifted consistently:

$$(R'_t - \hat{V}'_\theta(s_t))^2 = ((R_t - \Phi(s_t)) - (\hat{V}_\theta(s_t) - \Phi(s_t)))^2 = (R_t - \hat{V}_\theta(s_t))^2. \quad (34)$$

Entropy regularization is unaffected. Hence, PPO updates on shaped data are mathematically identical to unshaped updates when the critic is shifted by $-\Phi$.

Remark. This proposition is the PPO/GAE analogue of the classical policy-invariance theorem of Ng, Harada, and Russell [23]. Potential-based shaping leaves the true advantages $A'_\pi = A_\pi$ unchanged, and with the critic shift $\hat{V}' = \hat{V} - \Phi$, it also leaves the *estimated* advantages \hat{A}'_t identical sample-by-sample.

L.5. How Shaping Can Accelerate PPO Training Without Changing What Is Optimal

If shaping is implemented exactly as above, the PPO actor and critic updates are algebraically identical to those on the unshaped MDP. Any speed-up therefore arises from improved learning dynamics rather than a change in the underlying objective. Below we summarize the PPO-specific mechanisms through which shaping can still help.

(1) Variance reduction via informed baselines. In actor–critic methods, any state-dependent baseline can be subtracted from returns without biasing the gradient. Choosing Φ correlated with returns provides an additional control-variate that reduces the variance of \hat{A}_t , especially during early training when \hat{V}_θ is inaccurate. This can be realized by using the shaped critic $\hat{V}' = \hat{V} - \Phi$, or equivalently by adding $\Phi(s)$ to the baseline term in advantage computation.

(2) Finite-horizon credit assignment. PPO computes advantages over finite rollout segments. When partial-episode bootstrapping is used, the Φ terms cancel exactly (Proposition A.3). If truncated episodes are instead treated as terminal, shaping introduces a boundary term, providing denser feedback near segment boundaries and potentially improving early credit assignment—while still leaving the optimal policy unchanged.

(3) Critic warm-start and architectural priors. Setting the initial critic to $\hat{V}_0(s) \approx \Phi(s)$ or embedding $\Phi(s)$ as a fixed residual in the value head supplies a useful prior. Although the PPO gradients remain unbiased and unchanged in theory, better initialization of the critic often stabilizes optimization and accelerates the convergence of the actor–critic loop.

(4) Intentional partial shaping. Some implementations simply replace r with r' while keeping the critic unshifted. Then $\hat{A}'_t \neq \hat{A}_t$ early on, producing progress-aligned advantages that accelerate early learning. While this breaks the exact gradient equivalence, the optimal policy is still preserved by the theoretical invariance of potential-based shaping.

M. More Qualitative Results


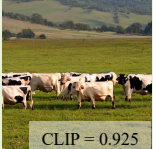



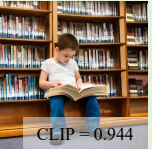
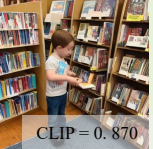



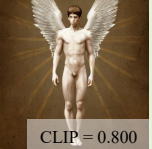
| Target Image | PROMPTMINER (Ours) | BLIP | CLIP-IG | VLM-as-expert | PH2P | PromptStealer | VGD |
|--|--|--|--|--|---|--|--|
|  MS COCO |  CLIP = 0.986 |  CLIP = 0.925 |  CLIP = 0.921 |  CLIP = 0.928 |  CLIP = 0.703 |  CLIP = 0.955 |  CLIP = 0.954 |
|  Flickr |  CLIP = 0.981 |  CLIP = 0.923 |  CLIP = 0.904 |  CLIP = 0.890 |  CLIP = 0.668 |  CLIP = 0.944 |  CLIP = 0.870 |
|  Lexica |  CLIP = 0.948 |  CLIP = 0.749 |  CLIP = 0.933 |  CLIP = 0.895 |  CLIP = 0.678 |  CLIP = 0.907 |  CLIP = 0.760 |
|  Lexica |  CLIP = 0.941 |  CLIP = 0.664 |  CLIP = 0.862 |  CLIP = 0.800 |  CLIP = 0.655 |  CLIP = 0.800 |  CLIP = 0.793 |

Figure 9. Visualization of images generated by Stable Diffusion v1.5 compared with target image.

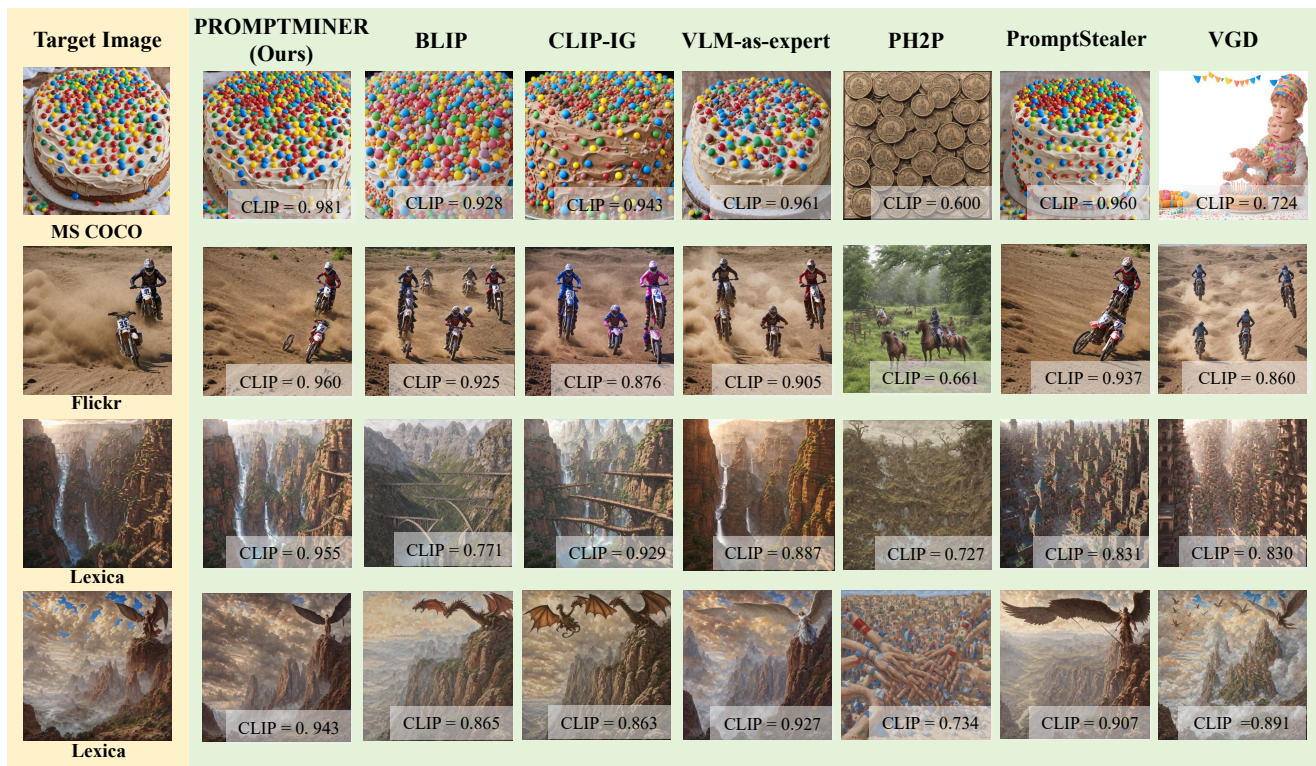


Figure 10. Visualization of images generated by SDXL Turbo compared with target image.



Figure 11. Visualization of images generated by Stable Diffusion 3.5 Medium compared with target image.

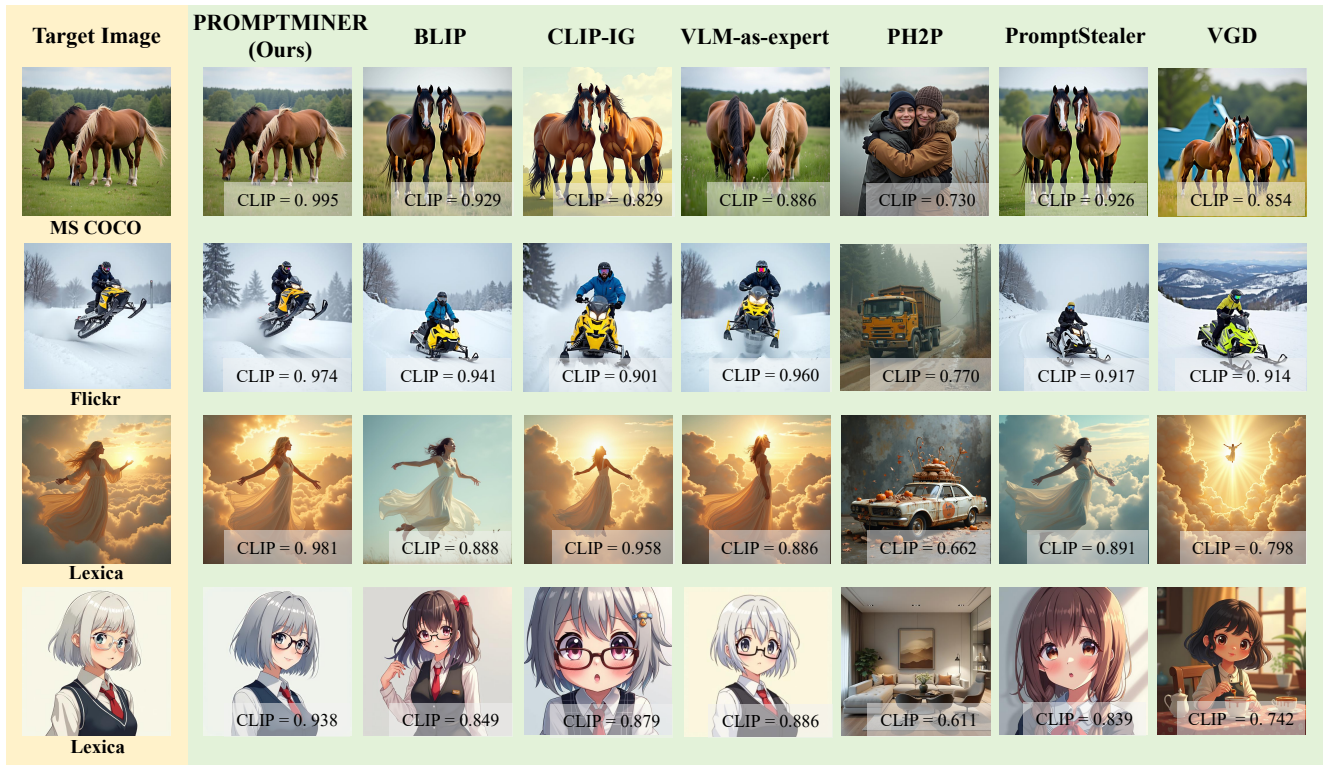


Figure 12. Visualization of mages generated by FLUX.1 dev compared with target image.

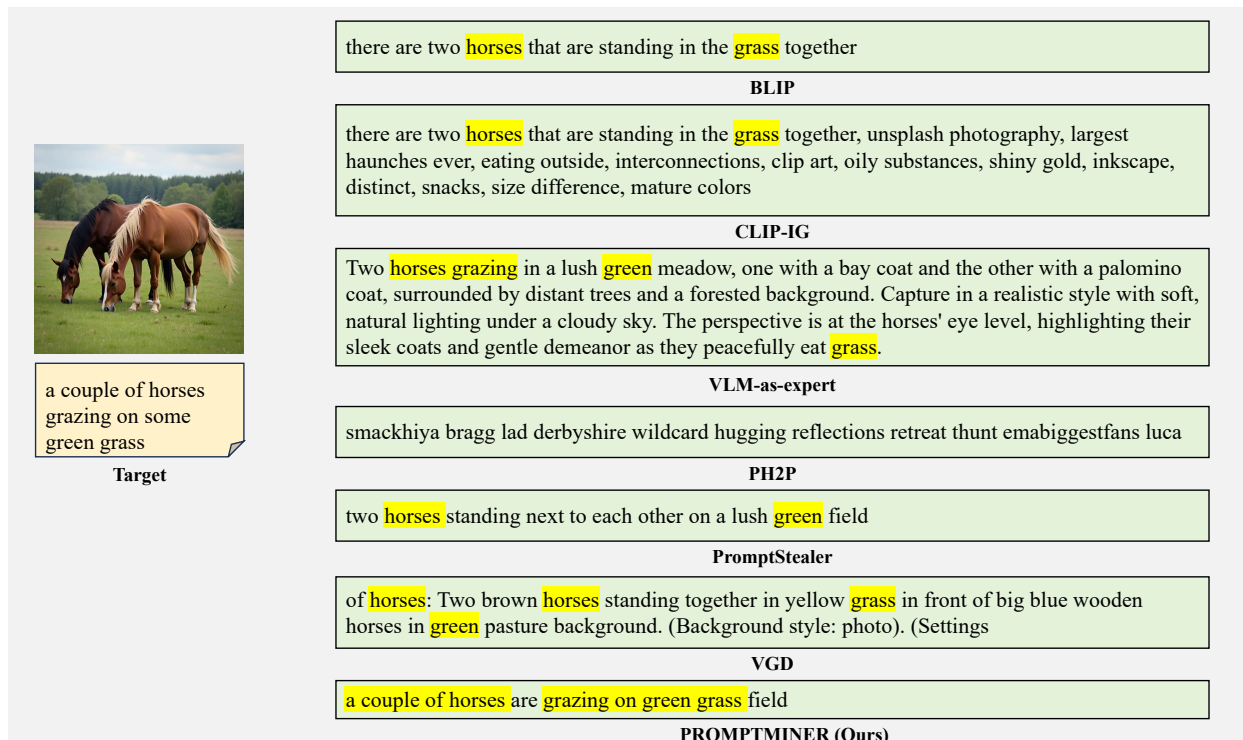


Figure 13. Qualitative Results of stolen prompts compared with target prompt on MS COCO.

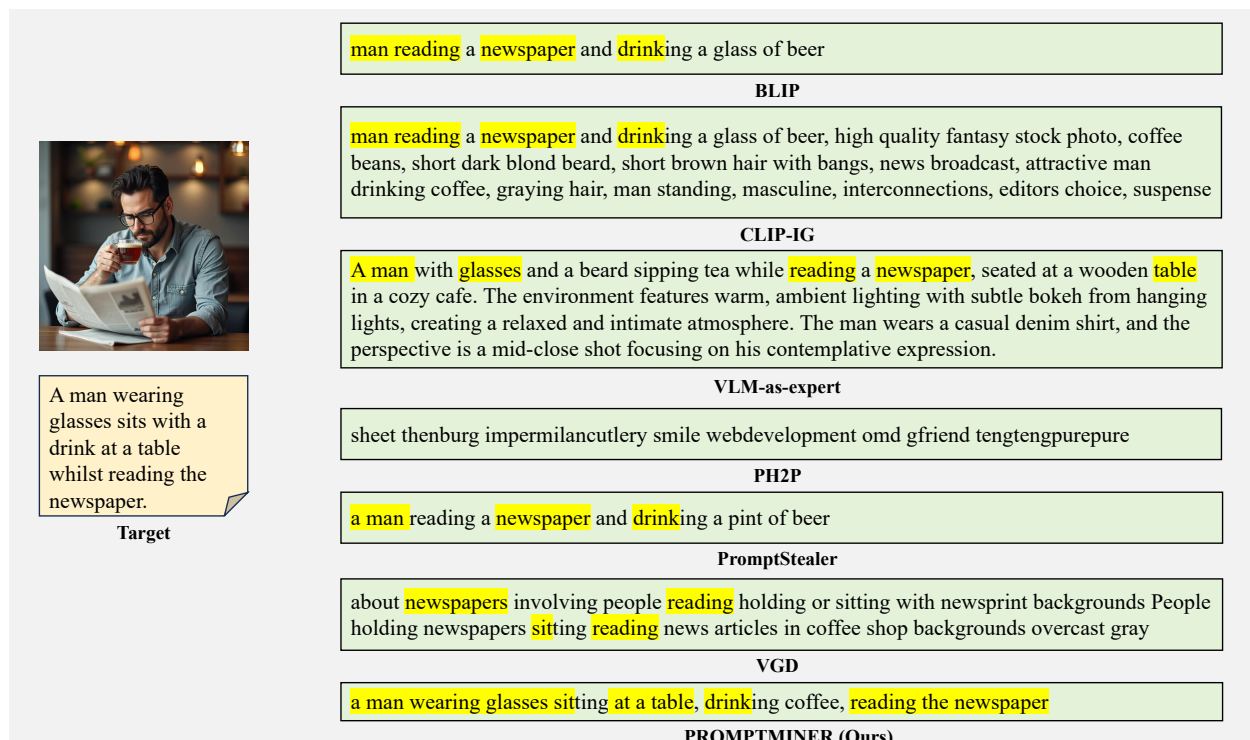


Figure 14. Qualitative Results of stolen prompts compared with target prompt on Flickr.

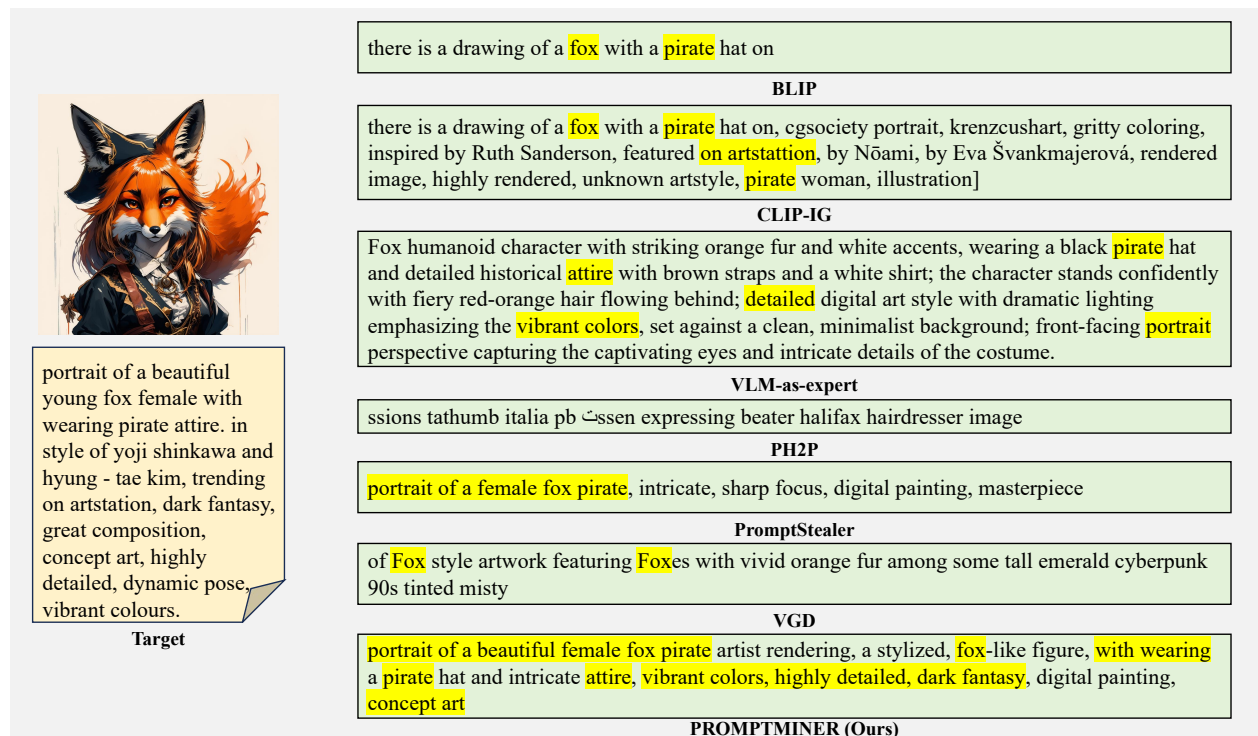


Figure 15. Qualitative Results of stolen prompts compared with target prompt on Lexica.