The Prompt Stealing Fallacy: Rethinking Metrics, Attacks, and Defenses

Zehang Deng[1,2], *Haoyang Li*[3], *Wanlun Ma*[1], *Ruoxi Sun*[2], *Derui Wang*[2],
Minhui Xue[2,4], *Haibo Hu*[3], *Sheng Wen*[1,*], *Yang Xiang*[1]

[1]*Swinburne University of Technology, Australia*
[2]*CSIRO's Data61, Australia*
[3]*The Hong Kong Polytechnic University, China*
[4]*Responsible AI Research (RAIR) Centre, Adelaide University, Australia*

Abstract

Text-to-image (T2I) models are increasingly embedded in creative workflows, where well-crafted prompts function as valuable forms of intellectual property (IP). However, these models are susceptible to prompt stealing attacks (PSAs), where adversaries aim to reconstruct the original prompts used to generate images. In this paper, 1) we identify key shortcomings in current evaluation practices and propose two improved metrics: *Style Similarity* (SS) and a novel *Prompt Significance* (PS) score, which together provide a more faithful assessment of PSA effectiveness. Rather than existing metrics that rely solely on semantic similarity between original and stolen information across text or image modalities, the new metrics PS and SS assess attack effectiveness with a more practical focus by explicitly accounting for the importance of modifiers and the style replication of images generated from stolen prompts. 2) Through extensive evaluation using these metrics, we find that existing PSA methods, ranging from soft prompt stealing in white-box settings to hard prompt stealing in black-box settings, are not as effective as reported, especially in recovering high-contribution prompt components. We attribute this to fundamental constraints: white-box methods suffer from mismatched optimization objectives that poorly align with token-level visual semantics, while black-box approaches experience severe information loss due to their decoupling from the target T2I model's generation process. 3) We further introduce PromptThief, a black-box PSA framework that addresses the information loss in prior methods by leveraging reinforcement learning with Semantic Text-Text Similarity (STS) and SS to guide high token-level contribution recovery. PromptThief significantly outperforms existing baselines across multiple metrics and real-world scenarios. 4) We propose and evaluate two defense mechanisms: an adversarial-example-based active approach and a passive scheme through feature-level prompt watermarking. Our evaluation reveals that the active defense offers only limited robustness against adaptive PSAs, highlighting the need for further exploration in this direction. In contrast, the passive watermarking scheme demonstrates strong and consistent detection performance, even under various image transformations, offering a practical and reliable path forward for prompt IP protection.

*Corresponding author.

1 Introduction

Text-to-image (T2I) generation models such as Stable Diffusion [29], DALL-E [22], and Midjourney [1] have significantly reshaped creative industries, enabling users to synthesize visually rich content from natural language prompts. At the core of this process lies prompt engineering [15, 20], a meticulous and often time-consuming task of crafting high-quality prompts that specify both subjects and stylistic modifiers to guide generation outcomes [6, 12, 23]. These prompts, increasingly treated as intellectual property (IP), are now traded on commercial marketplaces such as PromptBase [2], PromptHero [3], and Promptrr [4], incentivizing professional prompt creators and establishing an emergent "Prompt-as-a-Service" ecosystem. The global market for prompt engineering, valued at approximately US$ 222 million in 2023, is expected to expand at a compound annual growth rate (CAGR) of 32.8% through 2030 [27].

This commodification, however, has simultaneously surfaced a critical security threat: *prompt stealing attacks* (PSAs), in which adversaries attempt to reverse-engineer prompts from the generated images. Prior research has demonstrated both the feasibility and severity of such attacks. For example, PSteal [33] proposed a dual-module framework to reconstruct both the subject and stylistic modifiers, showing that even offline, black-box approaches can yield highly similar prompts. As shown in Table 1, we categorize PSAs into two primary types: *soft prompt stealing* is conducted in a white-box setting, where latent semantic representations are optimized and mapped to human-readable tokens (*e.g.,* PEZ [41], PH2P [19], and Textual Inversion [10]). In contrast, *hard prompt stealing* is carried out in a black-box setting, directly optimizing human-readable prompts based solely on original images M_{ori} (*e.g.,* PSteal [33], and DI-FT [43]).

Table 1: The overview of existing Prompt Stealing Attacks.

Methods	Original Task	Functionality O1	Reusability O2	Efficiency O3	Existing Metrics SIIS	Existing Metrics STS	Existing Metrics SITS	New Metrics SS	New Metrics PS	Adversary Knowledge
BLIP [17]	Image Captioning	◑	◑	●	○	●	○	/	/	Black Box
CLIP-IG [25]	Image Captioning	◑	◑	●	○	○	●	/	/	Black Box
GPT-4o [13]	Image Understanding	◑	◑	●	○	○	○	/	/	Black Box
PEZ [41]	Soft prompt Stealing	◑	○	○	●	○	○	/	/	White Box
PH2P [19]	Soft prompt Stealing	◑	○	○	●	●	○	/	/	White Box
Textual Inversion [10]	Soft prompt Stealing	◑	○	○	●	●	○	/	/	White Box
PSteal [33]	Hard prompt Stealing	◑	●	●	●	●	●	/	/	Black Box
DI-FT [43]	Hard prompt Stealing	◑	◑	●	●	●	○	/	/	Black Box
Ours	Hard prompt Stealing	●	●	●	●	●	○	●	●	Black Box

○: the item is not considered; ◑: the item is partially considered; ●: the item is considered; / : the item is not applicable at that time.
: indicates that evaluating this metric (SITS) exposes issues in the PSA method (Figure 3).

While recent studies have established the feasibility of PSAs, we highlight the need for a fundamental re-examination of existing approaches, particularly with respect to their evaluation metrics and reported performance. First, metrics widely adopted to assess PSA effectiveness, such as Semantic Image-Text Similarity (SITS), which measure prompt-image alignment, do not track attack performance and overlook token contributions in image generation. Our analysis reveals that such coarse-grained metrics often overlook subtle but high-impact modifiers, leading to misleading evaluations. Second, despite various proposed PSAs, current methods underperform in realistic black-box settings. Many rely on oversimplified prompt models or limited modifier sets, resulting in low recovery fidelity, especially for high-value tokens. These limitations highlight the need to revisit both the evaluation framework and the methodological design of PSAs to more accurately characterize their threat potential in the real world and guide the development of more robust defenses.

Contribution 1. In this paper, we challenge the prevailing assumptions in evaluating PSAs by rethinking the metrics commonly used to assess them (§4). We begin with a critical analysis of existing evaluation protocols, revealing that widely-used metrics such as SITS shows poor correlation with the actual visual and semantic fidelity of reconstructed prompts (See Figure 3). To address this gap, we adopt *Style Similarity* (SS), a previously proposed metric for measuring stylistic alignment between images [37], and further introduce a new metric, *Prompt Significance* (PS), which quantifies the contribution of stolen prompts to the stolen image. These two metrics provide a more fine-grained and faithful evaluation framework, enabling a deeper understanding of the true effectiveness of PSAs.

Contribution 2. Next, we revisit and systematically evaluate a range of existing PSAs (§5), including both white-box (*e.g.,* PEZ, PH2P, Textual Inversion) and black-box (*e.g.,* BLIP, CLIP-IG, GPT-4o, PSteal, DI-FT) PSAs, as summarized in Table 1. Through extensive experiments, we demonstrate that these methods, even under relaxed settings (white-box), fail to recover high-value, high-contribution tokens essential for guiding image generation. Our analysis reveals two key failure modes: white-box PSAs suffer from *mismatched discrete optimization objectives*, while black-box PSAs encounter significant information loss *because the prompt stealing process is decoupled from the target T2I model's generation process*, leading to suboptimal prompt reconstruction.

Contribution 3. To this end, we introduce PromptThief, a novel black-box PSA framework that addresses the information loss inherent in prior black-box methods (§6). PromptThief mitigates this loss through two key mechanisms: (1) it incorporates a reinforcement learning-based reward function that encourages the generation of high-contribution tokens, using STS to guide token-level relevance, and (2) it strategically couples with the target T2I model within a limited query budget, leveraging Style Similarity (SS) to directly evaluate the contribution of candidate tokens to the final generated image. The design is grounded in our proposed token contribution analysis, allowing PromptThief to prioritize high-impact prompt components and achieve significantly higher alignment with original prompts under strict query budgets. Our method achieves superior performance both in in-distribution and real-world prompt marketplace scenarios, with an average improvement of over 14% in SIIS compared to existing baselines. Furthermore, PromptThief yields over 16% gains on our proposed SS and PS metrics, highlighting its effectiveness in recovering both stylistic and semantically significant prompt components.

Contribution 4. We propose two practical and complementary defenses against prompt stealing attacks (PSAs). First, we design an *adversarial-example-based active defense* grounded in a token contribution perspective, which selectively perturbs high-contribution tokens, identified using impact and rarity, to mislead prompt reconstruction models while preserving generation quality. Second, we introduce a *feature-level prompt watermarking scheme* that embeds imperceptible and verifiable signatures into the latent features of generated images. The prompt watermarking scheme provides a guaran-

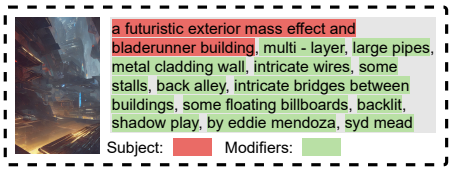


Figure 1: An example of prompt structure in T2I models.

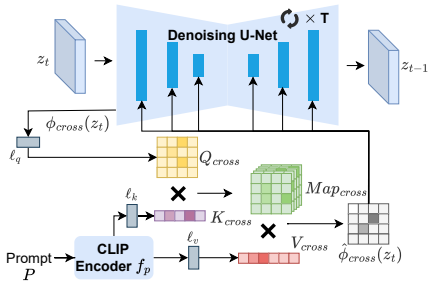


Figure 2: Illustration of cross attention in diffusion models.

tee on the false-positive rate for the watermark verification, while also providing robustness against evasive attacks that may cause false negatives. We evaluate both defenses against state-of-the-art attacks and demonstrate their effectiveness in mitigating prompt stealing attacks with minimal visual degradation. However, our evaluation reveals that current PSAs remain robust against the adversarial-example-based defense, indicating their limited effectiveness in practical scenarios. In contrast, the feature-level prompt watermarking scheme demonstrates strong detection performance, providing a reliable and resilient mechanism for tracing stolen prompts even under adaptive attack settings.

2 Preliminaries and Related Work

2.1 Prompt Structure in T2I Models

Unlike prompts for large language models (LLMs), text-to-image (T2I) prompts (See Figure 1) typically follow a two-part structure [33, 43]: a subject describing the main visual content, and a set of modifiers specifying stylistic or compositional attributes. The subject usually refers to tangible elements (e.g., buildings, characters), while modifiers describe abstract properties such as layout, lighting, and artistic style. Modifier phrases are commonly comma-separated.

2.2 Cross Attention in T2I Models

We illustrate how cross attention integrates text and image features in conditioned diffusion models [24, 26] in Figure 2. Given a noisy latent image z_t, the U-Net extracts spatial features $\phi_{\text{cross}}(z_t)$, which are linearly projected into queries $Q_{\text{cross}} = \ell_q(\phi_{\text{cross}}(z_t))$. Meanwhile, the text prompt P is encoded by a frozen CLIP encoder $f_p(P)$, and projected into keys and values: $K_{\text{cross}} = \ell_k(f_p(P))$, $V_{\text{cross}} = \ell_v(f_p(P))$. The cross attention map is computed via scaled dot-product attention: $Map_{\text{cross}} = \text{Softmax}\left(\frac{Q_{\text{cross}}K_{\text{cross}}^T}{\sqrt{d_{\text{cross}}}}\right)$, where d_{cross} are the dimensions of the keys and queries. This equation captures how each spatial location in the image attends to tokens in the prompt. The fused representation $\hat{\phi}_{cross}(z_t) = Map_{\text{cross}}V_{\text{cross}}$ injects textual semantics into the image features and is fed back into the denoising U-Net. Intuitively, the cross attention map Map_i gauges how much the i-th prompt token influences the image's spatial features as a whole. The fused representation is obtained by aggregating the value vectors with their corresponding attention weights: $\hat{\phi}_{\text{cross}}(z_t) = \sum_{i=1}^{N} \text{Map}_{cross}^{(i)} V_{\text{cross}}^{(i)}$, where N is the number of tokens in the prompt. This cross-attention mechanism enables the diffusion model to align specific prompt tokens with corresponding regions in the generated image.

2.3 Prompt Stealing Attack

Prompt stealing attacks against T2I models by inferring the original prompt P_{ori} from a generated image M_{ori}, threatening the intellectual property of creators who rely on carefully crafted prompts for artistic or commercial purposes [8, 30]. These attacks can lead to unauthorized replication, resale, and loss of exclusivity. Existing PSAs fall into two categories: *soft* and *hard* stealing attacks. Soft PSAs (under white-box settings), such as PH2P [19], PEZ [41] and Textual Inversion (TexInver) [10], aim to generate prompts with similar internal latent representations to the original but face challenges in exact replication due to the discreteness of tokens and high computational cost [16]. For hard PSAs (under black-box settings), PSteal [33] formulate the task as a multi-label classification problem over a large vocabulary. However, they achieve limited success, for example, recovering only 9.88% of prompt modifiers, highlighting the inherent difficulty of faithfully reconstructing original prompts. Similarly, DI-FT [43] employs BLEU as a reward function for prompt inversion across modalities (e.g., T2T, T2I, T2V). However, BLEU primarily captures surface-level n-gram overlaps and word order, rather than the high-contribution tokens encoded by the target T2I model's prompt-to-image mapping $f_p(P)$, resulting in degraded performance in the T2I setting.

3 Threat Model

In this section, we define the threat model of both the adversary and the defender.

Adversary's Objectives. Given a valuable image M_{ori} that was originally generated from a proprietary prompt P_{ori} via a text-to-image (T2I) model $\mathcal{G}$, the adversary's goal is to reconstruct a prompt P_{stolen} such that 1) the stolen prompt P_{stolen} should accurately maintain the meaning of the original prompt P_{ori} and the stolen image $M_{stolen} = \mathcal{G}(P_{\text{stolen}})$ is

visually similar to M_{ori} (O1: Functionality); 2) the stolen prompt P_{stolen} should ensure high reusability by enabling the generation of visually similar images, even when the main subjects are replaced (O2: Reusability); and 3) for commercial T2I models, the attacker aims to minimize the number of queries needed to recover the stolen prompt P_{stolen} (O3: Efficiency). These three objectives (O1/O2/O3) jointly reflect the economic reality of modern prompt marketplaces, where commercially purchased prompts are explicitly marketed and valued as reusable templates that can be applied across many generations and use cases, rather than as one-off instructions [14, 21]. Consequently, an attack that only satisfies O1 but fails O2 would not truly recover the economic value of the original proprietary prompt. At the same time, O3 (Efficiency) enforces economic rationality: if the total query cost of stealing a prompt exceeds, or even approaches, the market price of purchasing a high-quality reusable prompt, a rational adversary would simply buy the prompt instead of mounting the attack.

Adversary's Capabilities. The attacker operates in a practical black-box setting without access to the internal parameters or architecture of the T2I model $\mathcal{G}$. The adversary can only submit text prompts and observe the generated images. The model name is publicly available, as disclosed in prompt marketplaces such as PromptBase [2] and PromptHero [3]. Additionally, the ground-truth image M_{ori} is often displayed to attract potential buyers on these platforms. Importantly, the attacker's query budget is constrained by economic factors, including the average prompt price and the per-query cost. The average prompt price is listed at \$3.99[1], and under the Midjourney \$30/month standard plan[2], yielding a per-query cost of approximately \$0.006 (as detailed in Appendix A). Consequently, the attacker can afford at most 665 queries per target prompt within the budget.

Defender's Objectives. The defender, represented by individual prompt engineers, aims to safeguard their intellectual property within the prompt markets by preventing unauthorized extraction and misuse of their prompts. 1) For active (prevention-based) defense, such as adversarial examples, the defender seeks to thwart attackers from accurately reconstructing the original prompt, ensuring that images generated from any stolen prompt do not closely resemble those produced by the original. 2) For passive (detection-based) defenses, such as watermarking, the defender enables the tracing of unauthorized prompt usage and to attribute generated images back to their original prompts.

Defender's Capabilities. The defender operates under practical constraints typical of individual prompt engineers participating in a benign prompt market platform. They possess complete knowledge of their original prompt P_{ori} and the ground-truth image M_{ori} it generates. The defender is also able to modify the prompt P_{ori} and image M_{ori} before submitting it to the platform, as long as the intended image output remains unchanged on human perception. In our study, we consider both white-box and black-box defenses. In the white-box setting, the defender has full knowledge of the attacker, including the attack method and model. In contrast, the black-box setting assumes no prior knowledge of the attacker's strategies, but allows the defender to leverage publicly available prompt datasets (e.g., Lexica) to estimate the frequency distribution of modifiers.

[1] http://promptbase.com/marketplace?sortBy=score&time=month

[2] https://docs.midjourney.com/hc/en-us/articles/27870484040333-Comparing-Midjourney-Plans

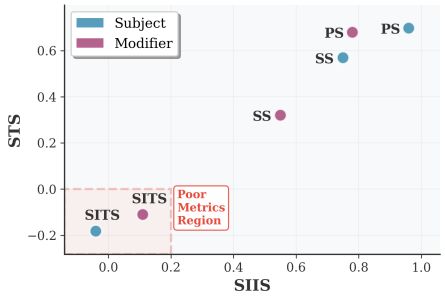


Figure 3: Correlation comparison between STS/SIIS and Other metrics.

4 Evaluation Metrics for PSAs

In this section, we first review and assess several widely used evaluation metrics in PSAs, and then propose two novel metrics: style similarity and prompt significance.

Evaluation Metrics of Functionality (O1). To satisfy *O1*, existing works [10, 19, 33, 41] employ three metrics:

1) Semantic Text-Text Similarity (STS) measures the similarity between two text descriptions using CLIP's text encoder:

$$\text{STS}(P_{\text{ori}}, P_{\text{stolen}}) = \cos(f_p(P_{\text{ori}}), f_p(P_{\text{stolen}})). \quad (1)$$

2) Semantic Image-Image Similarity (SIIS) is computed based on the image embeddings' similarity between two images:

$$\text{SIIS}(M_{\text{ori}}, M_{\text{stolen}}) = \cos(f_m(M_{\text{ori}}), f_m(M_{\text{stolen}})). \quad (2)$$

3) Semantic Image-Text Similarity (SITS) quantifies the alignment between a text and an image embedding:

$$\text{SITS}(P_{\text{stolen}}, M_{\text{stolen}}) = \cos(f_p(P_{\text{stolen}}), f_m(M_{\text{stolen}})), \quad (3)$$

We argue that SITS is uncorrelated with both SIIS and STS, exhibiting the lowest alignment between text and image embeddings due to the suboptimal performance of CLIP. Although CLIP's contrastive loss derives from both text and image data, the text predominantly describes *tangible objects* (i.e., subject) rather than abstract concepts, such as style, which substantially reduces the effectiveness of this metric in evaluating PSAs, particularly those containing numerous abstract modifiers. As shown in Figure 3, we conducted correlation experiments on Lexica dataset [33] between STS/SIIS

and other metrics. SITS lies in the "Poor Metrics Region" with near-zero or negative correlations with both SIIS and STS, indicating its limited effectiveness.

In addition, we introduce two novel metrics, Style Similarity (SS) and Prompt Significance (PS), to provide a more comprehensive evaluation of both the final and intermediate performances of prompt stealing attacks.

1) Style Similarity (SS) is evaluated using Contrastive Style Descriptors (CSD) models:

$$\mathrm{CSD}(M_{\mathrm{ori}}, M_{\mathrm{stolen}}) = \cos(f_{csd}(M_{\mathrm{ori}}), f_{csd}(M_{\mathrm{stolen}})) \quad (4)$$

CSD models are employed for style similarity measurement because traditional CLIP-based models often struggle to capture fine-grained stylistic details [37]. In contrast, CSD is specifically designed to emphasize stylistic features, such as color, textures, and artistic styles, by applying augmentations that retain style while minimizing content-related cues.

2) Prompt Significance (PS) measures how much a prompt phrase (*i.e.,* subject or modifier) contributes to the stolen image. In a successful attack scenario, we ideally expect both high semantic similarity (STS) between prompts and high visual similarity (SIIS) between images. However, as shown in Figure 4, high textual similarity (STS) does not always correlate with high image similarity (SIIS). This discrepancy makes it difficult to judge the effectiveness of an attack using either metric alone.

To address this limitation, we introduce a comprehensive metric, Prompt Significance (PS), which jointly considers both textual and visual alignment. Initially, we define it as a simple multiplication of the two metrics: $PS = STS \cdot SIIS$. However, we observed that SIIS exhibits a narrow sensitive range (see Figure 4(a), often saturating and failing to reflect subtle yet meaningful variations. To mitigate this, we propose a normalized version of PS:

$$\begin{aligned}\mathrm{PS}(P_{\mathrm{ori}}, P_{\mathrm{stolen}}, M_{\mathrm{ori}}, M_{\mathrm{stolen}}) = \; & STS(P_{\mathrm{ori}}, P_{\mathrm{stolen}}) \cdot \\ & \left(\frac{SIIS(M_{\mathrm{ori}}, M_{stolen}) - SIIS(M_{\mathrm{ori}}, M_{\mathrm{base}})}{SIIS(M_{\mathrm{ori}}, M'_{\mathrm{ori}}) - SIIS(M_{\mathrm{ori}}, M_{\mathrm{base}})} \right), \end{aligned} \quad (5)$$

where M_{base} is the image sampled from other unrelated images [3], serving as a baseline for similarity comparison, and M'_{ori} is the image reproduced from the original prompt. As validated in Figure 4(d), our proposed metric PS provides a sharper contrast across different PSA methods. The distribution of PS scores spans a wider range compared to SIIS and SS, allowing for better discrimination between PSAs. More importantly, PS captures both the final performance: i) how well the reconstructed prompt guides image generation (attack performance), and the intermediate semantic alignment, and ii) the textual similarity between the original and reconstructed prompts. This dual consideration makes PS a more comprehensive and sensitive metric, effectively highlighting the performance gap between vanilla, hard, and soft PSAs. Therefore, we employ four metrics of PSAs in our evaluation, SIIS, STS, SS, and PS.

Evaluation Metrics of Reusability (O2). To satisfy *Reusability (O2)*, we evaluate it using the CSD model [34], which measures Style Similarity (SS). Specifically, we assess whether altering the subject noun in the prompt affects the generated image's style. If the style remains consistent despite changes in the subject, it indicates higher reusability of the prompt.

Evaluation Metrics of Efficiency (O3). To satisfy *Efficiency (O3)*, we evaluate Query Efficiency (QE) by measuring the number of queries an attacker makes to the target T2I models.

5 Limitation of Existing PSAs

We revisit two types of representative prompt stealing attacks: soft prompt stealing (*e.g.,* PH2P [19], PEZ [41], and Textual Inversion [10]) and hard prompt stealing (PSteal [33] and DI-FT [43]). To conduct the evaluation, we randomly sampled 1000 prompt-image pairs from the Lexica dataset [33] and assessed them using five metrics: SIIS, STS, SS, and PS. We selected all methods listed in Table 1. Additionally, we refer to off-the-shelf models such as BLIP, CLIP-IG, and GPT-4o as vanilla methods.

5.1 Evaluation of Existing PSAs Success

The first question we want to investigate is *whether a successful PSA was executed.* (See Figure 4). We can make the following two key observations:

1) On SIIS and SS metrics (Figures 4(a) & 4(c)), hard PSAs show no significant difference from vanilla methods, *i.e.,* BLIP, CLIP-IG, and GPT-4o. Even though soft PSAs have access to the parameters of the target diffusion models, Only TexInver shows a statistically significant performance increase, surpassing the best-performing vanilla baseline. However, it requires over 2000 queries per prompt, exceeding the cost of simply purchasing the target model.

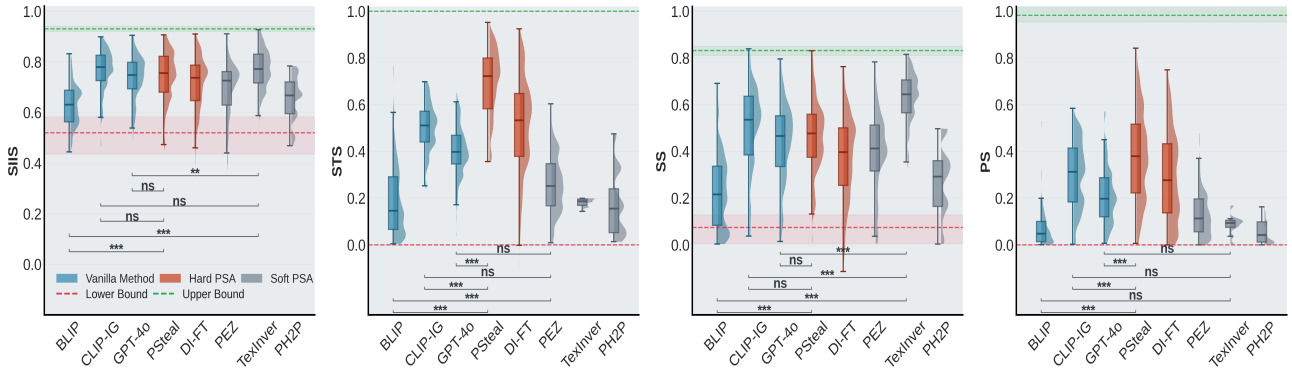
2) In contrast, for STS metrics (Figure 4(b)), black-box PSAs (PSteal and DI-FT) outperform vanilla methods, indicating they can extract more tokens (either exact matches or semantically similar ones). However, because these gains do not translate into improvements on SIIS and SS (Figures 4(a) & 4(c)) , we conclude that most of the stolen tokens are low-value or irrelevant to actual image generation quality.

5.2 Why Existing PSAs Fail

The second question is *why these PSAs failed*. We investigate this question from the following two aspects (See Figure 4):

Why white-box PSAs fail. As illustrated in Figure 4(c), only TexInver achieves the highest SS performance, demonstrating statistically significant improvements over CLIP-IG. In contrast, both PH2P and PEZ exhibit limited performance.

[3] $\mathrm{SIIS}(M_{\mathrm{ori}}, M_{\mathrm{base}})$ is computed on the Lexica dataset as the average score between M_{ori} and 100 randomly sampled unrelated prompts.



(a) (b) (c) (d)

Figure 4: Performance comparison of existing PSAs across four metrics (O1). Upper bounds reflect intra-prompt variation (averaged across multiple same prompt generations), while lower bounds reflect inter-prompt baselines (averaged over unrelated images/prompt). The null hypothesis (H_0): Vanilla PSA performs no better than Hard/Soft PSA ($\leq$). Statistical significance is indicated as follows: *ns* (not significant, $p > 0.05$), * ($p \leq 0.05$), ** ($p \leq 0.01$), *** ($p \leq 0.001$), and **** ($p \leq 0.0001$).

This discrepancy stems from the fact that, PH2P and PEZ indirectly aim to recover hard prompts (*i.e.,* human-readable words). This introduces a fundamental mismatch between the prompt recovery process and the underlying optimization objective. Concretely, these methods should attempt to find an embedding $\hat{v}$ that minimizes the following loss: $\min_{\hat{v}} \|\mathcal{G}(\hat{v}|z_t') - \mathcal{G}(f_p(P_{ori}|z_t))\|_2^2$, where z_t and $z_t' \sim \mathcal{N}(0,I)$ are the Gaussian noise sampled at each denoising step in the diffusion process. Then, the actual goal in PEZ and PH2P is to recover the discrete prompt P_{stolen}, which satisfies:

$$\min_{P_{stolen}} \|f_p(P_{stolen}) - \hat{v}\|_2^2. \tag{6}$$

This two-stage recovery process inevitably introduces an optimization gap, minimizing the embedding distance does not ensure semantic consistency or accurate reconstruction of the discrete prompt. Furthermore, since most prompt-based models are deployed within diffusion frameworks, the reverse generation process inherently involves sampling a random noise vector z_t' from a Gaussian distribution at step T. Crucially, this sampled z_t' differs from the original ground-truth noise z_t used during the forward diffusion process. Such stochasticity exacerbates the reconstruction error, making it inherently difficult to reliably recover the original discrete prompt from its embedding, even when $\hat{v}$ closely approximates $f_p(P_{ori})$.

Why Black-box PSAs Fail. Vanilla methods (BLIP, CLIP-IG, GPT-4o) and Black-box PSAs (PSteal, DI-FT) generally follow a two-stage pipeline: an image encoder f_m projects the target image M into a joint vision-language space, and a language generator $g(\cdot)$ produces a candidate prompt $P_{\text{stolen}} = g(f_m(M)), \quad P_{\text{stolen}} \in \mathcal{P}$, where $\mathcal{P}$ denotes the space of plausible natural language prompts. The objective is to approximate the original prompt P_{ori} and enable reconstruction of M_{stolen}. However, this indirect "image $\rightarrow$ embedding $\rightarrow$ text" mapping is inherently lossy: the projection discards fine-grained semantics, while the cross-modal generation introduces additional ambiguity. As a result, these methods show limited effectiveness in alignment-sensitive tasks (e.g., SIIS and SS).

PSteal and DI-FT attempt to mitigate this shortcoming but remain limited. PSteal appends modifiers predicted from an auxiliary classifier, yielding semantically richer captions and higher STS scores. DI-FT fine-tunes the generator on high-BLEU samples to produce prompts closer in surface form to P_{ori}. Yet both primarily optimize for text-level similarity rather than alignment with the diffusion model's internal representations. Consequently, despite apparent gains in STS (Figure 4(b)), they yield little benefit for SIIS and SS (Figures 4(a) & 4(c)).

Given that information loss is inevitable in both white-box and black-box PSAs, *our core idea is to avoid relying on exhaustive prompt reconstruction and instead concentrate on identifying and recovering high-contribution tokens, those that most strongly influence the T2I model behaviours.*

5.3 Token Contribution

Given conditioning tokens $P_{ori} = (X_1, X_2, ..., X_n)$ used by diffusion models to generate images, the contribution of token X_i is defined as:

$$C(X_i) = H(M) - H(M_{-i}), \tag{7}$$

where $M_{-i} = \mathcal{G}(P_{-X_i})$ represents the image generated by the T2I model G when token X_i is excluded. $M = \mathcal{G}(P_{ori})$ is the image generated with prompt P_{ori}. $H(\cdot)$ denotes the image quality evaluation function. A larger value of $C(X_i)$ indicates that token X_i plays a more significant role in determining the final quality of the image.

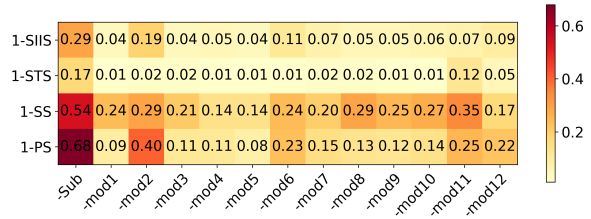


Figure 5: Token contribution effects are evaluated using SIIS, STS, SS, and PS on prompts sampled from Lexica, where perturbations are created by removing either the subject (*–Sub*) or a modifier (*–modN*).

Theorem 1 (*Token Contribution in diffusion models*)**.** *For any token X_i in the conditioning prompt, the token contribution $C(X_i)$ satisfies:*

$$C(X_i) \leq \Bigg(\underbrace{\alpha \|Map^{(i)}_{cross}\|_2 \, \|V^{(i)}_{cross}\|_2}_{\text{the direct contribution of the removed token}} \Bigg) + \underbrace{\beta \sum_{j \neq i} \|Map^{(j)}_{cross} - Map^{(j,-i)}_{cross}\|_2 \, \|V^{(j)}_{cross}\|_2}_{\text{the effect on remaining tokens due to attention re-normalization}}, \tag{8}$$

where $\alpha, \beta > 0$ are model-dependent constants and $Map^{(j,-i)}_{cross}$ is the re-normalized attention map for token X_j after removing token X_i. Theorem 1 characterizes the influence of a single token in the text prompt on the cross-modal feature fusion during generation. It shows that token contribution is determined both by the direct contribution of the removed token and the effect on remaining tokens (detailed proof in Appendix E).

By quantifying how each token shapes image generation through cross-attention, and bounding its direct and re-normalized effects, token contribution identifies the true IP-critical parts of a prompt for both attack and defense. Due to the black-box nature of our setting, the exact token contributions defined in Equations 8 cannot be directly computed, since both Map_{cross} and V_{cross} are inaccessible without knowledge of the target model $\mathcal{G}$. To approximate token contribution, we instead rely on metrics of functionalities (O1). As shown in Figure 5, the sensitivity of different metrics to token removal varies: STS responds sharply to the removal of a few critical word chunks (e.g., mod_{11}), whereas SS exhibits broader sensitivity across diverse modifiers. Motivated by these results, we employ STS as a proxy for token contribution during the reinforcement learning stage (§6.2), which reduces query overhead and avoids direct dependence on T2I models. In the subsequent selected search phase (§6.3), where querying T2I models is feasible, we switch to the SS metric, leveraging its stronger sensitivity to modifiers for identifying the final set of significant tokens.

6 A New Method of Prompt Stealing Attack

In this section, we introduce a novel prompt stealing attack, named PromptThief, which is designed based on the observations and insights discussed in §5. The attack is structured into three parts (See Figure 6):

Part 1: Technical Warm-up. LMM is warmed up using supervised fine-tuning on the Lexica dataset [33].

Part 2: Reinforcement Learning. We apply the Group Relative Policy Optimization (GRPO) [32], leveraging our estimated token contribution (STS as rewards to encourage the high-contribution token). Theoretical justifications for our choice of GRPO are presented in Appendix F.

Part 3: Selected Search. To compensate for the shortcomings in token contribution estimation using STS (as discussed in §5.3), we perform a limited number of searches and select the final stolen prompts based on SS scores.

6.1 Part 1: Technical Warm-up

In this stage, we conduct supervised fine-tuning (SFT) on the policy model π_θ using a dataset of *(image, instruction, original prompt)* pairs $(M_{\text{ori}}, I, P_{\text{ori}})$. The model learns to autoregressively generate the stolen prompt P_{stolen} conditioned on (M_{ori}, I), modeling the generation as a next-token prediction task until the special end token $\langle eos \rangle$ is emitted on Lexica dataset [33] $\mathcal{D}$. The SFT objective is:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{P_{\text{stolen}} \sim \mathcal{D}} \left[\sum_{t=1}^{L} \log \pi_\theta(a_t \mid s_t) \right], \tag{9}$$

where a_t is the action taken at time step t, and s_t is the corresponding state. After three training epochs, we observe stable convergence of $\mathcal{L}_{\text{SFT}}(\theta)$, indicating the model has acquired basic prompt reconstruction capabilities. The trained policy π^0_θ is then used to initialize the reinforcement learning stage.

6.2 Part 2: Reinforcement Learning

In this stage, we aim to enhance the policy model's performance through online self-learning using a dataset of (M, I, P_{ori}) with customized reward functions. Specifically, the initial policy model from the warm-up part samples stolen prompts, evaluates their correctness using GRPO Objective $\mathcal{J}_{GRPO}$, and updates its parameters in real time via GRPO [32].

Formulating Objective $\mathcal{J}_{\text{GRPO}}$. To further optimize the policy model beyond supervised learning, we adopt a goal-conditioned reinforcement learning framework. Specifically, we define a reward function that balances semantic quality and lexical overlap between the generated prompt and the original prompt. Our objective is inspired by Group Relative Policy Optimization (GRPO), where the reward at time step t is defined as:

$$r(s_t, a_t, s_{t+1}) = \alpha \cdot STS(s_{t+1}) + (1-\alpha) \cdot \text{Jac}(s_{t+1}), \alpha \in [0,1], \tag{10}$$

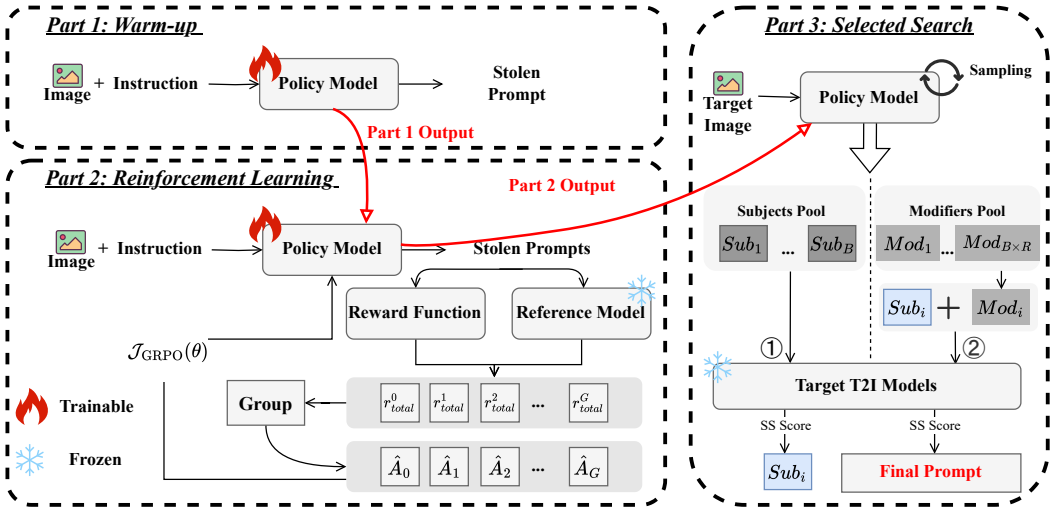


Figure 6: The overview of the proposed Prompt Stealing Attack, termed PromptThief.

where $STS(s_{t+1})$ estimates token contribution using a semantic similarity score ($f_p(\cdot)$), while $\text{Jac}(s_{t+1})$ denotes the Jaccard similarity between the generated tokens and ground truth tokens at step t.

Although STS effectively captures semantic alignment and was initially considered a standalone metric (see §5.3), we found through empirical analysis that relying solely on STS can cause training instability. This stems from the continuous nature of semantic similarity metrics, which do not always correlate well with discrete token predictions, leading to volatile gradients during learning (reflected in Equation 6). To mitigate this issue, we introduce the Jaccard score as a stabilizing component. It explicitly captures token-level overlap, helping regularize the learning signal when semantic similarity fluctuates. *Our hybrid reward design (STS + Jaccard) resolves the gradient misalignment that undermined prior reinforcement learning approaches, enabling stable and efficient learning under strict query budgets.*

In our experiments, we conducted hyperparameter tuning over $\alpha \in \{0.3, 0.5, 0.7\}$ and observed that setting $\alpha = 0.5$ achieved the best performance in terms of both convergence speed and generation quality. Therefore, we fix $\alpha = 0.5$ for all subsequent training and evaluation phases.

To control the extent of policy updates and prevent drastic changes, the total reward can be augmented with a regularization term as follows: $r_{\text{total}}(s_t, a_t, s_{t+1}) = r(s_t, a_t, s_{t+1}) - \beta \text{KL}\left(\pi_\theta(\cdot \mid s_t), \pi_\theta^{(0)}(\cdot \mid s_t)\right)$. For each image M and instruction I, a set of outputs $\{o_1, o_2, \ldots, o_G\}$ is sampled from the old policy model $\pi_\theta^{(0)}$. For each sampled output, the corresponding reward is computed using the total reward function, resulting in a reward set $\mathbf{r}_{\text{total}} = \{r^1_{\text{total}}, r^2_{\text{total}}, \ldots, r^G_{\text{total}}\}$. Based on this reward set, the estimated advantage for each sample is computed as:

$$\hat{A}_i = \tilde{r}_i = \frac{r_i - \text{mean}(\mathbf{r}_{\text{total}})}{\text{std}(\mathbf{r}_{\text{total}})}.$$

The Objective $\mathcal{J}_{GRPO}$ can be defined as follows:

$$\begin{aligned} \mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{M,I\sim P(M,I),\,\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(O|M,I)} \Bigg[\\ \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\Bigg\{ \min\left[\frac{\pi_\theta(o_{i,t}\mid M,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}\mid M,o_{i,<t})}\hat{A}_{i,t},\right. \\ \left.\text{clip}\left(\frac{\pi_\theta(o_{i,t}\mid M,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}\mid M,o_{i,<t})},\,1-\varepsilon,\,1+\varepsilon\right)\hat{A}_{i,t}\right] \\ -\beta\,\mathbb{D}_{\text{KL}}\left(\pi_\theta \,\|\, \pi_\theta^{(0)}\right)\Bigg\}\Bigg] \end{aligned} \tag{11}$$

After reinforcement learning, we can take the new trained policy model π_θ into next part.

6.3 Part 3: Selected Search

The trained policy model serves as a sampler to extract candidate subjects and modifiers from the target images. We conduct $B = 10$ [4]sampling rounds, where in each round, the model generates one subject and R modifiers. After B rounds, we obtain a subject pool $\{\text{Sub}_1, \ldots, \text{Sub}_B\}$ and a modifier pool $\{\text{Mod}_1, \ldots, \text{Mod}_{B\times R}\}$.

Subject selection is performed first, since subject tokens have a dominant influence on prompt semantics. For each candidate in the subject pool, we append the subject to a base prompt, generate an image with the target T2I model, and compute the SS score. We then select the subject that achieves the highest SS score.

[4]We found $B = 10$ is enough to extract the 95% word chunks.

For modifiers, we first apply frequency-based filtering, modifiers appearing in over half of the samples are directly selected. The remaining modifiers are ranked by frequency and added one by one in a greedy manner, evaluating each addition with the SS score. This process continues until the composed prompt reaches 77 tokens. This strategy typically requires 30-67 queries to construct a high-quality prompt while maintaining efficiency in the search process.

7 New Defenses for Prompt Stealing Attacks

As shown in Figure 5, different word chunks have varying impacts on image generation. Modifiers often play a critical role in shaping the style, quality, or uniqueness of the output, and thus exhibit distinct IP value compared to other parts of the prompt. Unlike prior defenses [33], we argue that selectively identifying valuable modifiers and applying targeted protection to them is key to effectively safeguarding the core intellectual property (IP) embedded in a prompt, especially since subjects can often be easily replaced without significantly affecting the underlying creative intent. Motivated by this insight, we propose *TokenGuard*, a comprehensive defensive framework focused on the selective protection of critical modifiers. *TokenGuard* integrates both adversarial-example-based defenses and prompt-level watermarking, providing robust IP protection in the evolving prompt marketplace.

7.1 Valuable Token Selection for Selective Protection

Before devising any defensive mechanism, we first identify the subset of *valuable modifiers* whose protection yields the greatest intellectual-property (IP) benefit. Drawing on practical prompt–engineering workflows [9, 11], a modifier's value is governed by two complementary criteria:

C1 (Impact). The modifier contributes substantially to image generation, as quantified by the token contribution score $C(X_m)$ defined in Eq. 7.

C2 (Rarity). The modifier is *not* among the most prevalent terms in the marketplace, so an attacker cannot trivially guess.

Notation. Let $\mathcal{D}$ be a corpus of N proprietary prompts and $\mathcal{M}$ the multiset of all modifiers extracted from $\mathcal{D}$. For a modifier $m \in \mathcal{M}$, let $f(m) := \frac{\#\text{occurrences of } m}{|\mathcal{M}|}$ denote its empirical frequency, and let $C(X_m)$ be its token-contribution score.

Step 1: Frequency-based pruning (satisfying C2). We remove ubiquitous modifiers whose market presence suggests low IP value. Specifically, sort $\mathcal{M}$ in descending order of $f(m)$ and discard the *head* portion

$$\mathcal{M}_{\text{head}} := \left\{ m \in \mathcal{M} \;\middle|\; \text{rank}_f(m) \le \tau \right\}, \quad \tau = \left\lceil 0.10 \cdot |\mathcal{M}| \right\rceil,$$

i.e. the top 10 % most-frequent modifiers.[5] The *tail* set $\widetilde{\mathcal{M}} := \mathcal{M} \setminus \mathcal{M}_{\text{head}}$ contains those modifiers whose rarity endows them with higher potential IP worth.

Step 2: Impact-weighted ranking (satisfying C1). Within $\widetilde{\mathcal{M}}$, we prioritise tokens according to their influence on the generated image. We normalise contributions to obtain a probability mass function $w(m) := \frac{C(X_m)}{\sum_{m' \in \widetilde{\mathcal{M}}} C(X_{m'})}$ for $m \in \widetilde{\mathcal{M}}$, which serves two purposes: (i) it yields a *continuous* importance weight, avoiding an arbitrary threshold on C, and (ii) it facilitates stochastic sampling in downstream defences (e.g. probabilistic watermark embedding). We use SS metrics to estimate the token contribution $C(X_m)$. The final high-impact set is obtained by selecting the top-K modifiers under $w(\cdot)$, $\mathcal{M}^\star := \arg\text{top}^K_{m \in \widetilde{\mathcal{M}}} w(m)$, where K is chosen to satisfy budgetary or perceptual constraints (e.g. total prompt length).

Discussion. By decoupling rarity and impact, the above two-stage procedure ensures that only those modifiers whose absence would noticeably degrade generation quality *and* whose appearance is insufficiently common to be guessed are earmarked for protection. Empirically, we find that filtering by $f(m)$ reduces the candidate pool by ≈85 %, while the subsequent $C(X_m)$-based ranking concentrates over 90 % of aggregate contribution mass into fewer than 20 tokens. These $\mathcal{M}^\star$ modifiers therefore represent the most cost-effective targets for the adversarial perturbations (§ 7.2) and watermark embedding (§ 7.3) that follow.

7.2 Adversarial-example-based Active Defense

Existing prompt stealing attacks, such as PSteal [33], construct a multi-label classifier [28] to build the mapping between images and the associated prompt modifiers [35]. Let the target multi-label classifier be denoted as $f : \mathcal{X} \to [0,1]^K$, where $\mathcal{X}$ is the input space (*e.g.,* the space of RGB images), and K is the number of possible labels. For an input image $x \in \mathcal{X}$, the classifier outputs a confidence vector $f(x) = (f_1(x), f_2(x), \ldots, f_K(x))$, where $f_k(x) \in [0,1]$ represents the confidence that label k applies to x.

White-box Defense. In the white-box setting, where full access to the target model f is available, we propose White-box Perturbation, which crafts an adversarial example x' to minimize the classifier's confidence over a target label set T, subject to an L_p-norm constraint $\|x' - x\|_p \le \varepsilon$. The objective function incorporates a token-weighted loss, defined as $\mathcal{L}^{\text{TC}}_{\text{adv}}(x') = \sum_{k \in T}(c_k[-\log(1 - f_k(x'))])$, where c_k represents the normalized contribution score of each label, encouraging stronger perturbation of semantically important tokens. This weighting scheme prioritizes high-impact features under a constrained perturbation budget, resulting in more targeted and efficient white-box adversarial defenses. More detailed in Appendix G.

Black-box Defense. In the black-box setting, where no access to the target model f's gradients, parameters, or architecture is available, we propose a two-stage defense framework. First,

[5] The 10 % cut-off follows empirical observations that the long-tail vocabulary begins almost immediately after this point.

a pre-trained conditional diffusion model generates a target image x_{target} conditioned on either a negative prompt (e.g., "without cats or dogs") or a prompt from the complement label set $S = \{1,\ldots,K\} \setminus T$. To improve semantic control, we introduce token-weighted classifier-free guidance, which scales null-conditioned deviations during the denoising process based on normalized token contributions. This mechanism emphasizes visually salient concepts while suppressing semantically weak ones, and maintains the $O(1)$ inference complexity of standard guidance. Second, we perform latent space optimization using a pre-trained VQ-VAE, aligning the adversarial image x' with x_{target} in latent space via an adversarial loss, while preserving perceptual similarity to the original image x through a regularization term. The total objective balances these two goals, resulting in effective black-box perturbations that are semantically targeted yet visually constrained. More detailed in Appendix H.

7.3 Feature-level Watermarking Passive Defense Scheme

We introduce a feature-level watermarking scheme that leverages the semantic discrepancy between original and adversarial images for detecting the presence or absence of target label features in the feature space of a multi-label classifier. The key idea is to treat the features corresponding to specific target labels as an implicit watermark embedded in the image's latent representation.

Let $E : \mathcal{X} \to \mathcal{Z}$ be a fixed pre-trained image encoder that maps an input image $x \in \mathcal{X}$ to a high-dimensional feature vector $z = E(x) \in \mathbb{R}^d$. A binary watermark extractor $W : \mathcal{Z} \to [0,1]$, typically implemented as a neural network, is trained to detect these target label features.

We define two datasets: $\mathcal{D}_{\text{pos}}$, consisting of images containing at least one target label (positive samples), and $\mathcal{D}_{\text{neg}}$, consisting of images with no target labels, including adversarial examples (negative samples). The watermark extractor W is trained to distinguish these using the following binary cross-entropy loss [44]:

$$\mathcal{L}_{\text{wm}}(W) = -\mathbb{E}_{x\sim\mathcal{D}_{\text{pos}}}[\log W(E(x))] - \mathbb{E}_{x\sim\mathcal{D}_{\text{neg}}}[\log(1 - W(E(x)))]. \tag{12}$$

During training, the parameters of the encoder E are frozen, and only the watermark extractor parameters are optimized, greatly reducing computational overhead.

Robustness analysis of watermarking. To ensure the trustworthiness of the watermark detection process, we formally analyze the false-positive (FP) guarantee and the robustness against changes in z in this part. Note that we have a limited test set of size $N = N_{\text{test, pos}} + N_{\text{test, neg}}$, to provide a confidence interval of the FP rate, we compute the Wilson interval [42] of the FP rate such that the highest FP rate can be bounded. Specifically, we have the upper bound of the FP rate as

$$\overline{FP} = \frac{1}{1+\rho^2/N}\left[FP + \frac{\rho^2}{2N} + \rho\sqrt{\frac{FP(1-FP)}{N} + \frac{\rho^2}{4N^2}}\right],$$

where $\rho = \Phi^{-1}(1-\alpha)$ is the inverse of the standard normal cumulative distribution function, given a confidence level $1-\alpha$. On the other hand, to analyze the robustness of detection towards changes in the extracted representation z, we have the following theorem:

Theorem 2 (*Robustness of watermark detection*)**.** *Given the extracted feature z and a one-layer neural network detector*[6] $\sigma(w^\top z + b)$*, let z_{pos} be the feature of a positive sample and z'_{pos} be that of a perturbed sample. The detection results will be guaranteed to change only if*

$$\|z'_{pos} - z_{pos}\| \geq \frac{2}{\|w\|}. \tag{13}$$

Proof. As we have a single-layer neural network with Sigmoid as the activation function, it is easy to know that $\sigma(w^\top z + b)$ is $\frac{1}{4}\|w\|$-Lipschitz. Therefore,

$$|W(z_{\text{pos}}) - W(z'_{\text{pos}})| \leq \frac{\|w\|}{4}\|z_{\text{pos}} - z'_{\text{pos}}\|. \tag{14}$$

There is a guaranteed FN only if

$$|W(z_{\text{pos}}) - W(z'_{\text{pos}})| \geq 0.5. \tag{15}$$

Plugging Equation 14 into Equation 15 concludes the proof. □

Theorem 2 provides a lower bound for improving the robustness of the detection. The detector will demonstrate a false negative (FN) under evasive attacks when the ℓ_2 distance between z'_{pos} and z_{pos} is no less than $\frac{2}{\|w\|}$. Based on the analysis, larger $\|w\|$ may compromise the robustness of the detector by reducing the minimal required distance between z'_{pos} and z_{pos} for fabricating an FN.

8 Evaluation of Attacks and Defenses

In this section, we present a comprehensive evaluation of the proposed attack and defense strategies. A detailed description of the experimental setup is provided in Appendix B.

8.1 Main Results on Attacks

8.1.1 Attack Functionality (O1)

As shown in Table 2, we present a detailed comparison between PromptThief and eight baseline methods, which include three white-box PSAs and four black-box PSAs. Here, our

[6] Our detector is precisely the single-layer network used in Eq. 12.

Table 2: In-distribution performance: Functionality (O1) results on PSAs. gray indicates white-box PSAs, while others correspond to black-box PSAs. OS denotes Open source.

Model	Test Set	Method	SIIS	STS	PS	SS
SD1.4 *(OS)*	Lexica *(In-distri)*	PEZ	0.69	0.09	0.03	0.39
		PH2P	0.63	0.14	0.06	0.27
		TexInver	0.76	0.20	0.11	0.63
		BLIP	0.63	0.18	0.07	0.22
		CLIP-IG	0.77	0.50	0.30	0.50
		GPT-4o	0.74	0.40	0.21	0.44
		PSteal	0.75	0.69	0.37	0.46
		DI-FT	0.74	0.54	0.28	0.40
		Ours	**0.89**	**0.69**	**0.53**	**0.63**

analysis primarily focuses on the quantitative results related to attack functionality (O1). Qualitative results are presented in Appendix C.

In-distribution performance. In the Table 2, where the T2I model is SD1.4 and the test set is Lexica, PromptThief achieves the best results across all four evaluation metrics: SIIS (0.89), STS (0.69), PS (0.53), and SS (0.63), outperforming even the white-box methods. For example, CLIP-IG, the best-performing black-box baseline in this setting, scores 0.77 on SIIS and 0.50 on STS, with lower scores on PS (0.30) and SS (0.50). Compared with PSteal, which is specifically designed for black-box prompt recovery, PromptThief achieves a substantially higher PS score (0.53 vs. 0.37), suggesting that our method recovers prompts that are much closer to the original ones in wording and style. These results demonstrate that PromptThief can conduct highly effective and semantically consistent attacks even without knowledge of the model's internal structure, while also delivering superior prompt fidelity relative to prior black-box baselines (see qualitative results (Figure 8, Appendix C)).

Out-of-distribution (OOD) performance. Illustrated in Table 3, we retain the same model (SD1.4) but evaluate on a test set (PromptBase/hero) collected from commercial platforms, representing an out-of-distribution scenario. PromptThief continues to demonstrate its superiority, achieving scores of 0.84 (SIIS), 0.63 (STS), 0.47 (PS), and 0.59 (SS). It consistently outperforms all baseline methods, including white-box approaches, highlighting its strong generalization capability across diverse data domains. Notably, the substantial improvement on SS (0.59 vs. 0.47 for DI-FT) indicates that our method recovers not only semantically faithful but also stylistically closer prompts in commercial settings, further underscoring its threat to proprietary prompt marketplaces (see qualitative results (Figure 9, Appendix C)).

OOD performance across T2I models. The third and most challenging setup considers a cross-model, out-of-distribution scenario by evaluating on DALL·E with the same PromptBase/hero test set (Table 4). In this proprietary setting, white-box methods are not applicable, so the comparison focuses exclusively on black-box baselines. Even under these constraints, PromptThief achieves the best results across all metrics (SIIS: 0.89, STS: 0.71, PS: 0.61, SS: 0.70), clearly surpassing the strongest alternatives. These findings demonstrate that PromptThief maintains high effectiveness without internal access, showcasing robust generalization across both model architectures and data domains (see qualitative results (Figure 10, Appendix C)).

Table 3: OOD performance: Functionality (O1) results on PSAs.

Model	Test Set	Method	SIIS	STS	PS	SS
SD1.4 *(OS)*	PromptBase/hero *(Commercial)*	PEZ	0.71	0.12	0.05	0.41
		PH2P	0.70	0.22	0.09	0.44
		TexInver	0.73	0.25	0.12	0.58
		BLIP	0.66	0.17	0.06	0.21
		CLIP-IG	0.74	0.41	0.21	0.39
		GPT-4o	0.77	0.45	0.25	0.40
		PSteal	0.73	0.55	0.27	0.39
		DI-FT	0.78	0.58	0.26	0.47
		Ours	**0.84**	**0.63**	**0.47**	**0.59**

Table 4: OOD performance across T2I models: Functionality (O1) results on PSAs. CS denotes closed source.

Model	Test Set	Method	SIIS	STS	PS	SS
DALLE *(CS)*	PromptBase/hero *(Commercial)*	BLIP	0.71	0.23	0.10	0.22
		CLIP-IG	0.81	0.56	0.38	0.53
		GPT-4o	0.78	0.41	0.24	0.50
		PSteal	0.77	0.63	0.37	0.51
		DI-FT	0.73	0.46	0.21	0.48
		Ours	**0.89**	**0.71**	**0.61**	**0.70**

8.1.2 Attack Reusability (O2)

Table 5 reports the results on the Reusability (O2) objective under different subject swap scenarios. Our method achieves the best performance across all settings. In the original subject setting, PromptThief reaches 0.63, compared to 0.50 for CLIP-IG, 0.46 for PSteal, and only 0.22 for BLIP. When identities are swapped, baseline methods exhibit sharp performance drops, for example, BLIP falls close to zero (0.02–0.03), and PSteal drops from 0.46 to as low as 0.22. Even the stronger baselines, such as CLIP-IG (0.41–0.43) and DI-FT (0.29–0.32), struggle to retain reusability. In contrast, PromptThief consistently maintains high scores (0.51–0.53), showing only a minor decrease compared to the original setting. These results highlight that our method is significantly more robust to identity variations, demonstrating its ability to generalize across diverse conditions. More qualitative examples are provided in Appendix D.

8.1.3 Computation Costs (O3)

We evaluated both training and evaluation costs, as summarized in Table 6. All GPU-h values are computed using A100 GPU rental rate in Lambda ($1.79/h)[7], and per query cost

[7] https://lambda.ai/pricing

Table 5: The effectiveness of Reusability (O2). YW = Young wife, YH = Young husband, EM = Elderly man.

Swap to	BLIP	CLIP-IG	GPT-4o	PSteal	DI-FT	Ours
Original	0.22	0.50	0.44	0.46	0.40	**0.63**
YW	0.02	0.41	0.39	0.27	0.32	**0.53**
YH	0.02	0.42	0.38	0.24	0.30	**0.51**
EM	0.03	0.43	0.40	0.22	0.29	**0.51**

Table 6: Computation cost breakdown for PSAs. Training costs only account for methods that require training. QE stands for query efficiency defined in § 4.

Training Costs (One for All)				
Method	GPU-h	Peak GPU Mem (GB)		Cost ($)
PSteal	53.2	30.6		95.2
DI-FT	312.2	113.6		558.8
PromptThief (Ours)	199.6	75.2		357.3
Inference Costs (Per Sample)				
Method	Run-time (mins)	Min QE	Max QE	Avg Cost ($)
PEZ	8.3	1000	1000	6.25
PH2P	73.3	>2000	>2000	>12
TexInver	18.2	>2000	>2000	>12
BLIP	0.01	/	/	<0.01
CLIP-IG	16.3	/	/	0.49
GPT-4o	0.04	/	/	0.04
PSteal	0.02	/	/	<0.01
DI-FT	0.2	/	/	<0.01
PromptThief (Ours)	3.6	30	67	0.27

is set to $ 0.006 (details in Appx. A). PromptThief offers up to 44× lower inference cost than soft PSAs, making it significantly more scalable in real-world settings. Although BLIP, GPT-4o, PSteal, and DI-FT exhibit very low inference costs, their reconstruction utility is substantially lower than PromptThief (See Table 2-4). Moreover, the training cost of PromptThief remains at a moderate level: higher than PSteal but still far below DI-FT, achieving a more favorable balance between computational overhead and reconstruction quality.

8.1.4 Ablation Study on Attacks

Attack performance on each part contribution. As shown in Table 7, we conduct an ablation study to analyze the individual and combined contributions of the key components in our attack framework: Warm-up, Reinforcement Learning (RL), and Selected Search. The baseline configuration with none of these components enabled (first row) yields relatively low performance, with SIIS/SS at 0.69/0.37 and STS at 0.34, indicating limited effectiveness in both visual alignment and prompt similarity. Introducing the Warm-up phase alone results in a notable performance increase (SIIS improves to 0.73, and SS rises to 0.42), suggesting its importance in providing a good initialization for subsequent optimization. When Warm-up is combined with RL, we observe further improvements across all metrics (e.g., SIIS: 0.79, SS: 0.51), demonstrating that RL helps to effectively explore the search space for more optimal prompts. Interestingly, even without reinforcement learning or warm-up, enabling Selected Search still yields strong performance (SIIS: 0.78, SS: 0.47). This indicates its effectiveness in steering the search process toward semantically richer candidates. The best performance is achieved when all three components are used together, resulting in the highest scores across the board (SIIS: 0.89, STS: 0.69, PS: 0.53, and SS: 0.63). These results clearly show that each component contributes meaningfully to the final performance, and their combination yields a highly effective and robust attack strategy.

Table 7: Ablation study on our attack method.

Warm-up	RL	Selected Search	SIIS	STS	PS	SS
○	○	○	0.69	0.34	0.13	0.37
●	○	○	0.73	0.39	0.19	0.42
●	●	○	0.79	0.50	0.31	0.51
○	○	●	0.78	0.49	0.30	0.47
●	●	●	**0.89**	**0.69**	**0.53**	**0.63**

Table 8: The Effectiveness of Transferability across different T2I models.

Swap to	SIIS	STS	PS	SS
SD1.4 → SD1.4	0.89	0.69	0.53	0.63
DALLE → DALLE	0.89	0.71	0.61	0.70
DALLE → SD1.4	0.80	0.59	0.38	0.53
SD1.4 → DALLE	0.79	0.53	0.33	0.48

Attack performance on unknown target model $\mathcal{G}$. Table 8 presents the evaluation of attack transferability across different text-to-image (T2I) models. When the attack is performed on the same model used for query generation (e.g., SD1.4→SD1.4), we observe strong performance across all metrics, indicating high attack effectiveness. However, when transferring the attack to a different target model (e.g., DALLE→SD1.4 or SD1.4→DALLE), performance decreases consistently across SIIS, STS, PS, and SS metrics. Despite this drop, our approach still achieves reasonable transferability, suggesting its potential applicability in black-box scenarios where the target model $\mathcal{G}$ is unknown.

8.2 Key Findings on Defenses

8.2.1 Adversarial-example-based Active Defense

Performance of Adversarial-example-based Active Defense. As shown in Table 9, the adversarial-example (AE)-based defense with $\varepsilon = 0.05$ yields only marginal performance degradation across both black-box and white-box settings. For instance, in the black-box setting, CLIP-IG drops slightly from 0.77 (SIIS) without defense (Table 2) to 0.76, and PSteal decreases from 0.75 to 0.74. Our PromptThief remains the most effective under this defense, achieving SIIS of 0.85, STS of 0.67, PS of 0.51, and SS of 0.61, values only slightly lower than its undefended performance (0.89, 0.69, 0.53, 0.63). A similar trend holds in the white-box setting, where PromptThief still attains the highest results (0.86/0.67/0.53/0.61), while baselines such as BLIP (0.61/0.18/0.05/0.21) and PSteal

Table 9: Experimental results on adversarial-example-based defense against various attacks. ε denotes the perturbation budget.

Model	Defenses	Method	SIIS	STS	PS	SS
Black-box	AE ($\varepsilon = 0.05$)	PEZ	0.68	0.08	0.03	0.39
		PH2P	0.62	0.13	0.03	0.27
		TexInver	0.75	0.19	0.10	0.62
		BLIP	0.61	0.17	0.04	0.20
		CLIP-IG	0.76	0.50	0.28	0.46
		GPT-4o	0.74	0.39	0.20	0.42
		PSteal	0.74	0.62	0.32	0.45
		DI-FT	0.73	0.50	0.23	0.39
		PromptThief	**0.85**	**0.67**	**0.51**	**0.61**
White-box	AE ($\varepsilon = 0.05$)	PEZ	0.67	0.08	0.03	0.38
		PH2P	0.62	0.12	0.03	0.26
		TexInver	0.73	0.18	0.09	0.60
		BLIP	0.61	0.18	0.05	0.21
		CLIP-IG	0.76	0.48	0.29	0.50
		GPT-4o	0.73	0.39	0.29	0.41
		PSteal	0.72	0.61	0.27	0.41
		DI-FT	0.73	0.50	0.27	0.39
		PromptThief	**0.86**	**0.67**	**0.53**	**0.61**

Table 10: Adaptive results on the adversarial-example-based active defense on the Lexica with SD1.4.

Method	SIIS	STS	PS	SS
PromptThief	0.89	0.69	0.53	0.63
PromptThief + AE	0.86	0.67	0.53	0.61
PromptThief_adapt^{AE} + AE	0.88	0.69	0.54	0.63

(0.72/0.61/0.27/0.41) remain far behind.

To better understand the limitations of our adversarial design method, we conduct additional experiments focusing specifically on valuable word chunks. Table 11 reports the STS scores across different variants, where $STS_{value}^{modifier}$ and $STS_{all}^{modifier}$ represent the semantic similarity of valuable and all modifiers, respectively, and STS_{value}^{Prompt} and STS_{all}^{Prompt} denote the corresponding values computed over the entire prompt. Our method demonstrates a clear advantage in suppressing the extraction of valuable modifiers, as indicated by the notably lower $STS_{value}^{modifier}$ scores. However, the defense becomes less effective when evaluated on all modifiers or full prompts. This is likely due to semantic overlap between valuable and non-valuable tokens, which allows attackers to partially recover the original semantics. These findings underscore the severity and subtlety of PSAs, particularly when key semantic components lack protection.

Adaptive attacks against adversarial-example-based active defense. To model a stronger adversary, we further consider an adaptive variant of PROMPTTHIEF, denoted as PromptThief_adapt^{AE}. This adaptive attacker is explicitly trained on samples generated by our adversarial-example-based active defense ($\varepsilon = 0.05$), by re-running both the warm-up stage (Part 1) and the reinforcement learning stage (Part 2) on the perturbed data distribution. At test time, the adaptive attacker only observes defended images and is never exposed to the corresponding clean counterparts.

Table 11: Experimental results on adversarial-example-based defense, focusing on the valuable word chunks.

Method	$STS_{value}^{modifier}$	$STS_{all}^{modifier}$	STS_{value}^{Prompt}	STS_{all}^{Prompt}
PSteal	0.16	0.46	0.48	0.62
Ours-wb	0.01	0.26	0.40	0.58
Ours-bb	0.13	0.48	0.44	0.62

The results in Table 10 show that when the attacker is explicitly trained on the adversarial-example distribution, PromptThief_adapt^{AE} slightly improves over the non-adaptive variant and even matches or marginally exceeds the performance achieved on clean samples. This indicates that knowledge of the AE-induced data distribution allows the attacker to partially adapt and recover its extraction capability.

8.2.2 Feature-level Watermarking Passive Defense Scheme

Table 12 summarizes the bit accuracy of watermark extraction under various image transformations, both with and without white-box or black-box adversarial-example-based defenses. Our passive watermarking scheme demonstrates high robustness when integrated into two prompt stealing methods, *i.e.,* PSteal and PromptThief, achieving consistently high bit accuracy across image transformations (*e.g.,* cropping, brightness adjustment, contrast changes, JPEG compression, and their combination). This indicates that our passive watermark can be reliably extracted even under significant visual distortions, highlighting its effectiveness.

In contrast, when adversarial-example-based defenses are applied (either white-box or black-box), the watermark accuracy drops significantly, especially for the PSteal variant (*e.g.,* from 0.95 to 0.39 under no attack). However, as shown in Table 9, these perturbations fail to meaningfully impair the functionality of prompt stealing itself. This indicates that while adversarial perturbations can obscure watermark detection, they do not mitigate the underlying threat, and thus fail to serve as effective defenses.

Adaptive attacks against feature-level watermarking passive defense. We also evaluate an adaptive attacker that explicitly optimizes to evade our feature-level watermark detector. Building on PromptThief, we modify the GRPO reward in Eq. 10 by adding a watermark-avoidance penalty based on the detector output $W(E(M_{\text{stolen}}))$, marked as PromptThief_adapt^{AE}. The adaptive reward is defined as

$$r_{\text{wm}}(s_t, a_t, s_{t+1}) = r(s_t, a_t, s_{t+1}) - \lambda \ell_{\text{wm}}(W(E(M_{\text{stolen}}))),$$

where $W(E(\cdot))$ is the watermark detector operating on feature embeddings, and ℓ_{wm} is calculated using eq. 12. This encourages the attacker to jointly (i) produce high-utility stolen prompts and (ii) adjust them to reduce the watermark detection probability.

Our experiments (Table 13) reveal that PromptThief_adapt^{AE} remains a strong attacker when

Table 12: Experimental results on watermark bit accuracy under various data transformations. Higher bit accuracy indicates better watermark retention. "WB" refers to the application of white-box adversarial-example-based defense, while "BB" denotes black-box adversarial-example-based defense.

Method	None	Crop	Brigh.	Cont.	JPEG	Comb.
PSteal	0.95	0.86	0.93	0.94	0.94	0.93
PromptThief (Ours)	0.89	0.77	0.87	0.88	0.89	0.87
PSteal-WB	0.39	0.39	0.41	0.41	0.40	0.41
PSteal-BB	0.39	0.39	0.41	0.42	0.40	0.41
PromptThief (Ours)-BB	0.57	0.53	0.58	0.59	0.59	0.59

Table 13: Adaptive results on the feature-level watermarking passive defense (watermark bit accuracy) under various data transformations.

Method	λ	None	Crop	Brigh.	Cont.	JPEG	Comb.
PromptThief (Ours)	/	0.89	0.77	0.87	0.88	0.89	0.87
PROMPTTHIEF_adapt^{WM}	0.1	0.71	0.61	0.76	0.69	0.72	0.71
PROMPTTHIEF_adapt^{WM}	0.3	0.61	0.66	0.63	0.60	0.61	0.63

optimized to evade feature-level watermarking, substantially reducing watermark bit accuracy across all transformations as λ increases. This shows that our attack remains effective even under watermark-aware adaptive attacks.

9 Discussion

Key takeaways. We have two main takeaways, on PSAs and on defenses. For PSA, PromptThief exhibits strong generalization across data distributions and target models. Experimental results show that contribution-aware PSA via GRPO training coupled with model-guided selected search at inference is more stable, robust to distribution shifts, and consistently converges toward high-contribution token subsets that preserve both semantic intent and visual fidelity. For defenses, our results indicate that adversarial-example-based active defenses that perturb only high-contribution tokens are fundamentally limited: attackers can "reward-hack" the contribution-aware objectives by replacing protected modifiers with semantically overlapping alternatives, thereby largely bypassing token-level perturbations. In contrast, our feature-level watermarking embeds a binary signature in the classifier feature space and trains a detector on positive/negative pairs with formal FP/FN guarantees, so the watermark remains stable under image transformations, allowing reliable ex-post detection of unauthorized prompt usage and offering a more practical path for prompt-IP protection.

Failure cases. Representative failure cases arise when the ground-truth prompt reflects a highly realistic photographic style with rich lighting descriptions (Figure 7). Although the original prompt specifies detailed illumination cues, PromptThief often recovers only coarse substitutes such as "cinematic lighting," resulting in mismatched photographic atmosphere. Moreover, realistic prompts frequently encode complex background structures and object interactions that our method only partially captures. For example, in case (a), fine-grained details of the bed and window lighting are missing, while in case (b), the recovered prompt omits key background elements such as the bookshelf and the associated light-environment interactions. While subject semantics are preserved, the stolen images lose the nuanced realism of the original scenes.

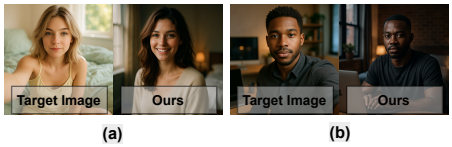


Figure 7: Failure Example for PromptThief.

Practical implications. Our results point to two concrete design shifts for the emerging "prompt-as-a-service" ecosystem. First, SS and PS should replace SITS metrics as the primary criteria for auditing both PSAs and active defenses: they explicitly expose when recovered prompts are missing high-contribution modifiers and thus more faithfully reflect the IP that is actually at risk. Second, our attack demonstrates strong effectiveness even against commercial prompt marketplaces, achieving high reconstruction quality at a substantially lower cost than the typical marketplace price. This highlights an urgent need for platforms to adopt stricter mechanisms to safeguard creators' prompt IP and prevent unauthorized reuse (e.g., Feature-level Watermarking Passive Defense).

10 Conclusion

In this work, we examined eight methodologies for evaluating PSAs in text-to-image T2I models, identifying shortcomings in metrics such as SITS. To address these, we proposed two metrics: SS and PS, providing a more comprehensive view of PSA effectiveness. We introduced PromptThief, a black-box PSA framework that reduces information loss and better targets high-impact prompt components via SS and SIIS. Extensive evaluations show that PromptThief outperforms prior approaches under both white- and black-box settings. We further proposed two defenses: an adversarial-example-based (active) and a feature-level prompt watermarking (passive) scheme. Our evaluation shows that while adversarial defenses offer limited robustness, feature-level watermarking ensures robust attribution even under image transformations.

Acknowledgments

Haoyang Li and Haibo Hu were supported by the National Natural Science Foundation of China (Grant No: 92270123), and the Research Grants Council (Grant No: 15208923 and 15207725), Hong Kong SAR, China.

Ethical Considerations

Data Sources and Compliance. All experiments were conducted using pre-trained large vision-language models (LLaVA-1.5) obtained from Hugging Face and trained on publicly available datasets. Our prompt-stealing evaluation primarily relied on test prompts derived from open-source datasets. In addition, a small subset of prompts was legally acquired from commercial marketplaces such as PromptBase and PromptHero under their respective terms of service. No private, user-generated, or personally identifiable data were collected or used. All datasets were stored locally on secure servers accessible only to the author team and were neither redistributed nor transmitted to third parties. All model usage and data acquisition complied with licenses/policies.

Stakeholders. The stakeholders potentially affected by this research include: (1) *prompt engineers, independent creators, and commercial prompt-marketplace providers*, whose intellectual property and platform reliability may be impacted by the disclosure of prompt-stealing vulnerabilities but who ultimately benefit from stronger protection mechanisms; (2) *end-users*, who rely on model services and expect their interactions and data to remain private; and (3) *the research community*, which benefits from a clearer understanding of the realistic capabilities and limitations of prompt-stealing attacks and corresponding defenses.

Ethical Principles. Our study follows the four principles of the Menlo Report. *Beneficence:* The primary goal of this work is to improve intellectual property protection by rigorously evaluating prompt-stealing threats and proposing effective defenses, thereby supporting the development of more realistic and robust security mechanisms. *Respect for Persons:* As described above, we avoided the use of any private, user-contributed, or personally identifiable data, relying exclusively on public or legitimately purchased datasets. *Justice:* We carefully considered the distribution of risks and benefits, particularly recognizing power asymmetries between researchers and independent prompt creators. To avoid amplifying harm, no proprietary prompt content is reproduced or redistributed, and access to sensitive attack implementations is restricted. *Respect for Law and Public Interest:* All datasets and models were obtained and used in compliance with licensing terms, and no scraping, circumvention of access controls, or interaction with private services was performed.

Potential Harms and Mitigations. We identified two primary categories of potential harm. First, *misuse of attack methods*: our prompt-stealing techniques could be misinterpreted as providing a blueprint for more effective attacks. To mitigate this risk, the full attack implementation was made available only to reviewers during the peer-review process for reproducibility. Post-publication access to the complete codebase is restricted and granted upon request to verified researchers conducting legitimate security analysis, following a responsible disclosure approach. Second, *data privacy and intellectual property risks*: these were mitigated by strictly avoiding private or user-generated data and by ensuring that all prompts were either publicly available or explicitly sold under commercial terms. No acquired data were redistributed.

We actively pursued harm reduction by designing and evaluating new defenses of PSAs. These defenses provide a practical mechanism for protecting prompt intellectual property and reduces the likelihood that the evaluated attacks can be successfully misused in real-world settings.

Decision Rationale. We weighed the potential risks of misuse against the benefits of advancing scientific understanding and improving defensive capabilities. By clarifying common misconceptions about prompt stealing, evaluating realistic threat models, and proposing concrete detection mechanisms, the benefits of this work substantially outweigh the residual risks. The study was designed to minimize harm, respect legal and ethical boundaries, and protect the interests of affected stakeholders. We therefore conclude that conducting and publishing this research is ethically justified.

Open Science

The source code of our work is publicly available at the repository. During the review stage, we provide all artifacts necessary for reproducing our results, including our proposed attack method, both defense mechanisms (active and passive), and all configuration files, hyperparameter settings, and environment specifications.

Code Organization. *1) Attack Method.* All components related to the attack pipeline are located under the `attack/` directory. A detailed `README` is provided within this folder, offering step-by-step instructions for downloading datasets, preprocessing them, obtaining the required base models, and executing the full attack procedure end-to-end. *2) Defense Methods.* The two defense mechanisms, Adversarial-example-based Active Defense and Feature-level Watermarking Passive Defense, are implemented under `defense/`. The corresponding `README` file in this directory describes their usage and provides instructions for running the defenses.

Accessibility Policy. *During the review stage.* We grant full access to both the attack and defense implementations through our public repository, ensuring that the review committees can comprehensively verify all experimental claims. *Upon publication.* To mitigate potential misuse, the attack code will be removed from the public repository after acceptance and subsequently shared only with verified researchers upon request. The defense code, by contrast, will remain permanently and openly accessible to support community-wide deployment and research on mitigation strategies.

Data Availability. All model training in our work uses publicly available datasets. A small portion of prompts used for evaluation was purchased from commercial prompt marketplaces such as PromptBase/Hero. Due to possible intellectual

property concerns regarding the sellers of these proprietary prompts, we do not release these specific examples.

References

[1] Midjourney. https://www.midjourney.com/home, 2025. Accessed: 2025-04-01.

[2] Promptbase. https://promptbase.com/, 2025. Accessed: 2025-04-01.

[3] PromptHero. https://prompthero.com/, 2025. Accessed: 2025-04-01.

[4] Promptrr. https://promptrr.io/, 2025. Accessed: 2025-04-01.

[5] Rajendra Bhatia and Chandler Davis. A cauchy-schwarz inequality for operators with applications. *Linear algebra and its applications*, 223:119–129, 1995.

[6] Stephen Brade, Bryan Wang, Mauricio Sousa, Sageev Oore, and Tovi Grossman. Promptify: Text-to-image generation through interactive prompt exploration with large language models. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–14, 2023.

[7] Qi Cai, Zhuoran Yang, Chi Jin, and Zhaoran Wang. Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*, pages 1283–1294. PMLR, 2020.

[8] Minsuk Chang, Stefania Druga, Alexander J Fiannaca, Pedro Vergani, Chinmay Kulkarni, Carrie J Cai, and Michael Terry. The prompt artists. In *Proceedings of the 15th Conference on Creativity and Cognition*, pages 75–87, 2023.

[9] Zijian Feng, Hanzhang Zhou, ZIXIAO ZHU, Junlang Qian, and Kezhi Mao. Unveiling and manipulating prompt influence in large language models. In *The Twelfth International Conference on Learning Representations*.

[10] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*.

[11] Jiyeon Han, Hwanil Choi, Yunjey Choi, Junho Kim, Jung-Woo Ha, and Jaesik Choi. Rarity score: A new metric to evaluate the uncommonness of synthesized images. In *The Eleventh International Conference on Learning Representations*.

[12] Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. Optimizing prompts for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:66923–66939, 2023.

[13] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[14] Md Raihanul Islam. *Prompt engineering in marketing analytics: A comprehensive blueprint for strategy and application*. Md Raihanul Islam, 2025.

[15] Ian Khan. *The quick guide to prompt engineering: Generative AI tips and tricks for ChatGPT, Bard, Dall-E, and Midjourney*. John Wiley & Sons, 2024.

[16] Hao Li, Soham De, Zheng Xu, Christoph Studer, Hanan Samet, and Tom Goldstein. Training quantized nets: A deeper understanding. *Advances in Neural Information Processing Systems*, 30, 2017.

[17] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[18] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[19] Shweta Mahajan, Tanzila Rahman, Kwang Moo Yi, and Leonid Sigal. Prompting hard or hardly prompting: Prompt inversion for text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6808–6817, 2024.

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

[21] Yuetian Mao, Junjie He, and Chunyang Chen. From prompts to templates: A systematic prompt template analysis for real-world llmapps. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, pages 75–86, 2025.

[22] OpenAI. Dall·e 2, 2024. Accessed: 2025-03-10.

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

[24] Geon Yeong Park, Jeongsol Kim, Beomsu Kim, Sang Wan Lee, and Jong Chul Ye. Energy-based cross attention for bayesian context update in text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:76382–76408, 2023.

[25] pharmapsychotic. Clip-interrogator. https://github.com/pharmapsychotic/clip-interrogator, 2024. Accessed: 2024-06-26.

[26] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[27] Grand View Research. Prompt engineering market size, share & trends analysis report by component (software, services), by technique (n-shot, generated knowledge), by application, by industry, by region, and segment forecasts, 2024 - 2030, 2024.

[28] Tal Ridnik, Gilad Sharir, Avi Ben-Cohen, Emanuel Ben-Baruch, and Asaf Noy. Ml-decoder: Scalable and versatile classification head. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 32–41, 2023.

[29] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[30] Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y Zhao. Glaze: Protecting artists from style mimicry by {Text-to-Image} models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2187–2204, 2023.

[31] Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y Zhao. Glaze: Protecting artists from style mimicry by {Text-to-Image} models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2187–2204, 2023.

[32] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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

[34] Gowthami Somepalli, Anubhav Gupta, Kamal Gupta, Shramay Palta, Micah Goldblum, Jonas Geiping, Abhinav Shrivastava, and Tom Goldstein. Measuring style similarity in diffusion models. *ECCV*, 2024.

[35] Adane Nega Tarekegn, Mario Giacobini, and Krzysztof Michalak. A review of methods for imbalanced multi-label classification. *Pattern Recognition*, 118:107965, 2021.

[36] Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. In *International Conference on Learning Representations*, 2020.

[37] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9568–9578, 2024.

[38] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6309–6318, 2017.

[39] Vladimir Vovk. The fundamental nature of the log loss function. *Fields of logic and computation II: Essays dedicated To Yuri Gurevich on the Occasion of His 75th Birthday*, pages 307–318, 2015.

[40] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.

[41] Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *Advances in Neural Information Processing Systems*, 36:51008–51025, 2023.

[42] Edwin B Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

[43] Dayong Ye, Tianqing Zhu, Feng He, Bo Liu, Minhui Xue, and Wanlei Zhou. Cross-modal prompt inversion: Unifying threats to text and image generative ai models. In *34rd USENIX Security Symposium (USENIX Security 24)*. USENIX Association, August 2025.

[44] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 657–672, 2018.

Appendix

A Per-query Cost

The per-query cost depends on the attacker's chosen subscription plan. Taking Midjourney as an example, the standard plan is priced at $30/month and includes **15 hours of fast GPU time** and **unlimited relax GPU time**[8]. According to official documentation, each fast GPU query takes approximately one minute, resulting in: $\frac{15 \text{ hours} \times 60 \text{ minutes/hour}}{1 \text{ minute/query}} = 900$ fast queries/month. For relax mode, which ranges from 0 to 10 minutes per image, we conservatively assume 10 minutes per query. If the system is used continuously (24 hours/-day), the monthly relax-mode query capacity is: $\frac{24 \times 60}{10} = 144$ queries/day, $144 \times 30 = 4320$ relax queries/month. Summing both modes yields a total query capacity of: $900\ (\text{fast}) + 4320\ (\text{relax}) = 5220$ queries/month. This leads to an effective cost of: $\frac{30}{5220} \approx 0.0057$ USD/query ≈ 0.006 USD. Hence, attackers can achieve large-scale querying at minimal cost under this subscription model.

B Experimental Setup

Metrics for Attack.
Functionality (O1). Measured using STS, SIIS, SS,and PS to assess attack performance
Reusability (O2). Evaluated with SS to check style consistency when changing the image subject.
Efficiency (O3). Assessed via Query Efficiency (QE), indicating the number of queries needed to obtain a usable prompt.
Attack Datasets. Following previous work [33], we use the test split of the Lexica dataset (containing 12,293 examples) as our in-distribution evaluation benchmark. To further assess the real-world applicability of our methods, we also evaluate them on 10 prompts sourced from commercial prompt marketplaces, such as PromptBase [2] and PromptHero [3], selected or purchased at random.
Attack Baselines. As summarized in Table 1, we select eight representative prompt-stealing attacks as baselines, encompassing both white-box methods (PEZ [41], PH2P [19], TexInver [10]) and black-box approaches (BLIP [17], CLIP-IG [25], GPT-4o [13], PSteal [33], and DI-FT [43]).
Attacked T2I Models. We consider two representative T2I models: Stable Diffusion v1.4 (SD1.4) and DALLE3. SD1.4 serves as a representative open-source model, while DALLE3 is a proprietary commercial model accessible via API. For each image generation process, we sample 50 steps with default settings [40].
Our Attack Setup. In our attack implementation, we utilize Llava-1.5-7b [18] as the policy model, which is trained on the Lexica dataset's training split, consisting of 61,467 examples [33]. The training procedure begins with a 3-epoch warm-up phase, followed by 30 epochs of reinforcement learning until convergence. Throughout the training, the model's maximum output length is constrained to 1024 tokens.

[8] https://docs.midjourney.com/hc/en-us/articles/32016412137741-GPU-Speed-Fast-Relax-Turbo

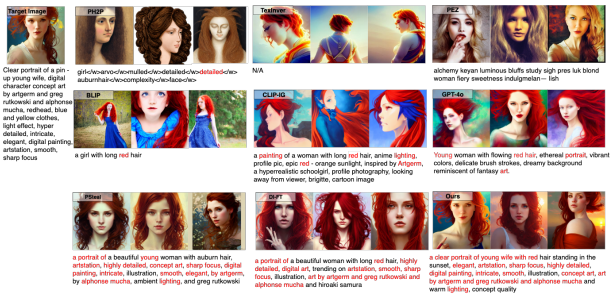


Figure 8: An Example of PSAs on the Lexica Dataset using SD1.4. The text below each image is the original/stolen prompt.

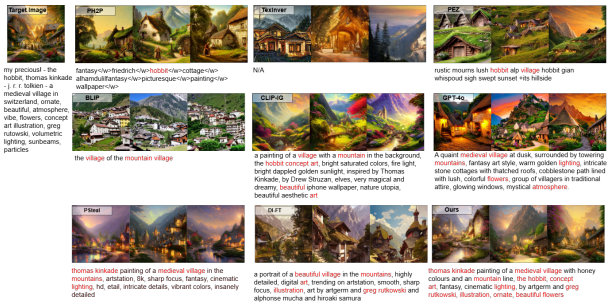


Figure 9: An Example of PSAs on the PromptHero using SD1.4.

Our Defense Setup. In our defense setup, we utilize real-world prompts from the test split of the Lexica dataset. For the text-to-image (T2I) generation, we employ Stable Diffusion 1.4 (SD 1.4). To simulate various deployment conditions and enhance robustness, we apply a range of image transformations, including cropping, brightness adjustment, contrast modification, JPEG compression, and a combined transformation set (Comb).

Metrics for Defense. Similar to PSAs, we apply SIIS/STS/PS/SS to evaluate the defense performance of the adversarial example. An effective defense should achieve low SIIS/STS/PS/SS. Additionally, we assess the effectiveness of the watermark on a held-out test set via bit accuracy:

$$\text{Accuracy}_{\text{pos}} = \frac{1}{N_{\text{test,pos}}} \sum_{x_i \in \mathcal{D}_{\text{test,pos}}} \mathbb{I}(W(E(x_i)) > 0.5),$$

$$\text{Accuracy}_{\text{neg}} = \frac{1}{N_{\text{test,neg}}} \sum_{x_j \in \mathcal{D}_{\text{test,neg}}} \mathbb{I}(W(E(x_j)) < 0.5),$$

where $\mathbb{I}$ is the indicator function. High values of both accuracies indicate robust separation between images with and without target label features.

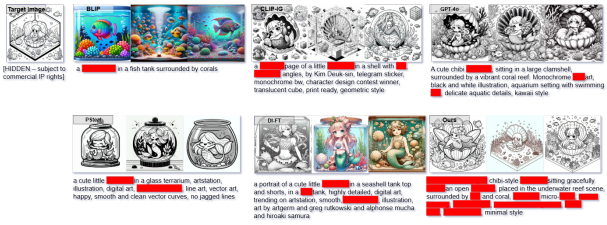


Figure 10: An Example of Attacks on the PromptBase using DELLE models. ■ indicates word chunks same as the original prompt.

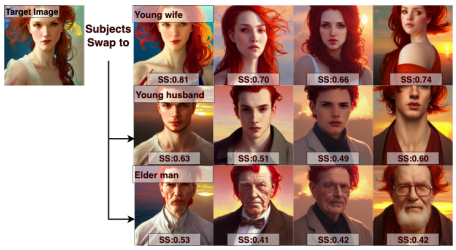


Figure 11: Reusability results of the example.

C Qualitative Attack Results of O1

Across Figures 8-10, which span in-distribution PSAs on the Lexica dataset using SD 1.4, out-distribution PSAs on PromptHero using SD 1.4, and cross-model PSAs on PromptBase using DALLE, we observe a consistent pattern: PromptThief reliably reconstructs both semantic and stylistic elements with high fidelity.

D Qualitative Attack Results of O2

Figure 11 shows that PromptThief's reconstructed prompts remain highly reusable: even after controlled subject swaps (e.g., "young husband," "elderly man"), the generations preserve strong stylistic coherence and maintain high SS scores.

E Proof of the Token Contribution (Theorem 1)

Proof. Recall the cross-attention output at diffusion step t: $\hat{\phi}_{\text{cross}}(z_t) = \sum_{j=1}^{n} Map_{\text{cross}}^{(j)} V_{\text{cross}}^{(j)}$, where for each token X_j and each spatial location ℓ, $Map_{\text{cross}}^{(j)}[\ell] = \frac{\exp\left(Q[\ell,:]\cdot K_{\text{cross}}^{(j)}/\sqrt{d_k}\right)}{\sum_{k=1}^{n}\exp\left(Q[\ell,:]\cdot K_{\text{cross}}^{(k)}/\sqrt{d_k}\right)}$.

1. Remove token X_i. When X_i is deleted, the renormalized maps for $j \neq i$ satisfy $Map_{\text{cross}}^{(j,-i)}[\ell] = \frac{\exp(Q[\ell,:]\cdot K_{\text{cross}}^{(j)}/\sqrt{d_k})}{\sum_{k\neq i}\exp(Q[\ell,:]\cdot K_{\text{cross}}^{(k)}/\sqrt{d_k})} = \frac{Map_{\text{cross}}^{(j)}[\ell]}{1-Map_{\text{cross}}^{(i)}[\ell]}$. Hence for each ℓ, $\left|Map_{\text{cross}}^{(j)}[\ell] - Map_{\text{cross}}^{(j,-i)}[\ell]\right| = \frac{Map_{\text{cross}}^{(j)}[\ell]\, Map_{\text{cross}}^{(i)}[\ell]}{1-Map_{\text{cross}}^{(i)}[\ell]}$.

Vectorizing over ℓ and using $\|a \odot b\|_2 \leq \|a\|_2 \|b\|_\infty$, we can get the following bound that characterizes the effects of removing token i: $\|Map_{\text{cross}}^{(j)} - Map_{\text{cross}}^{(j,-i)}\|_2 \leq \frac{\|Map_{\text{cross}}^{(j)} \odot Map_{\text{cross}}^{(i)}\|_2}{1-\|Map_{\text{cross}}^{(i)}\|_\infty} \leq \frac{\|Map_{\text{cross}}^{(i)}\|_\infty}{1-\|Map_{\text{cross}}^{(i)}\|_\infty}\|Map_{\text{cross}}^{(j)}\|_2$.

2. Decompose the output difference. Let $\hat{\phi}_{\text{cross}}^{(-i)}(z_t) = \sum_{j\neq i} Map_{\text{cross}}^{(j,-i)} V_{\text{cross}}^{(j)}$, so $\Delta\hat{\phi} = Map_{\text{cross}}^{(i)} V_{\text{cross}}^{(i)} + \sum_{j\neq i}\left(Map_{\text{cross}}^{(j)} - Map_{\text{cross}}^{(j,-i)}\right) V_{\text{cross}}^{(j)}$.

By the triangle inequality and submultiplicativity, $\|\Delta\hat{\phi}\|_2 \leq \|Map_{\text{cross}}^{(i)}\|_2 \|V_{\text{cross}}^{(i)}\|_2 + \sum_{j\neq i}\|Map_{\text{cross}}^{(j)} - Map_{\text{cross}}^{(j,-i)}\|_2 \|V_{\text{cross}}^{(j)}\|_2$.

3. Relate to the quality metric. Assume the quality metric $H(\hat{\phi}_{\text{cross}})$ is differentiable (or Lipschitz) in its argument. A first-order expansion gives $\Delta H := H\left(\hat{\phi}_{\text{cross}}(z_t)\right) - H\left(\hat{\phi}_{\text{cross}}^{(-i)}(z_t)\right) \approx \nabla_{\hat{\phi}} H \cdot \Delta\hat{\phi}$, and by Cauchy–Schwarz [5], $|\Delta H| \leq \|\nabla_{\hat{\phi}} H\|_2 \|\Delta\hat{\phi}\|_2$. Set $\alpha = \|\nabla_{\hat{\phi}} H\|_2$ and $\beta = \|\nabla_{\hat{\phi}} H\|_2$. Combining the above bounds yields exactly the statement of Equation (8), finishing the proof. □

F Theoretical Analysis for RL Selection

This section justifies *why* we adopt GRPO rather than the more widely-used Proximal Policy Optimisation (PPO) to improve the policy model in Stage 6.2. The analysis proceeds in three steps: (i) a unified objective under a KL–trust-region, (ii) closed-form optimality of GRPO, and (iii) a performance-gap bound showing that GRPO's expected improvement is never worse, and is usually strictly better, than PPO's under the same update budget.

(i) Unified objective under a KL–trust-region. Let π denote the current policy and π' a candidate update. Denote the advantage under π by $A^{\pi}(s,a)$. Using the *Performance Difference Lemma* [7] we have $V^{\pi'}(s_1) - V^{\pi}(s_1) = \mathbb{E}_{\pi'}[A^{\pi}(s,a)]$. Imposing a KL budget $\text{KL}(\pi' \,\|\, \pi) \leq \delta$, the update is the solution of

$$\max_{\pi'} \mathbb{E}_{\pi'}[A^{\pi}] \quad \text{s.t. } \text{KL}(\pi' \,\|\, \pi) \leq \delta. \tag{16}$$

(ii) closed-form optimality of GRPO. Problem (16) is a linearly-constrained convex program whose Lagrangian admits the mirror-descent solution (See Eq. (15) §4 in [36])

$$\pi'_{\text{opt}}(a \mid s) = \frac{\pi(a \mid s)\exp\left(\eta A^{\pi}(s,a)\right)}{\sum_{a'}\pi(a' \mid s)\exp\left(\eta A^{\pi}(s,a')\right)},$$

with η chosen to satisfy the KL constraint. GRPO realises this update exactly: the group baseline yields a zero-mean, unit-variance, minimum-variance advantage estimator, and the exponentiated reweighting applies the mirror-descent softmax step. Hence GRPO matches the *closed-form optimum* of (16), while PPO only approximates it.

(iii) Performance-Gap Bound. Combining the facts above yields the following guarantee:

Theorem 3 (GRPO Dominates PPO). *Let π'_G and π'_P be the GRPO and PPO updates obtained from the same batch of trajectories under the same KL radius δ. Assume the PPO critic has bias ε_c s.t. $\mathbb{E}[A^{\pi}_{PPO}] = \mathbb{E}[A^{\pi}] - \varepsilon_c$. If $\varepsilon_c \geq 0$ or any clip event occurs, then*

$$V^{\pi'_G}(s_1) - V^{\pi}(s_1) \;\geq\; V^{\pi'_P}(s_1) - V^{\pi}(s_1),$$

with equality only if no *ratio is clipped* and $\varepsilon_c = 0$.

Empirically we almost always face critic bias and at least a few clipped ratios; Theorem 3 therefore predicts (and our experiments confirm) that GRPO achieves a *larger* per-update improvement as well as faster overall convergence.

Practical Implications. GRPO offers clear advantages in sample efficiency, computation, and stability. Its lower-variance, unbiased group baseline makes each trajectory more informative, while the removal of the value network cuts GPU memory usage by roughly one policy-model copy and reduces wall-time per step by about half. Moreover, eliminating value-function drift and gradient truncation yields markedly smoother and more predictable learning dynamics.

G White-box Adversarial Example-based Defense

Given an original image $x \in \mathcal{X}$ and a subset of target labels $T \subseteq \{1,2,\ldots,K\}$, the defender's goal is to construct an adversarial image $x' \in \mathcal{X}$ such that $f_k(x')$ is minimized for all $k \in T$, implying that the classifier assigns low confidence to the target labels for x'. Additionally, we enforce that the perturbation remains imperceptible by constraining $\|x' - x\|_p \leq \varepsilon$, where $\varepsilon > 0$ is the perturbation budget, and $\|\cdot\|_p$ is the L_p-norm (typically $p = 2$ or $p = \infty$). Thus, we define our TokenGuard-WP (White-box Perturbation) loss function as: $\min_{x'} \mathcal{L}^{\text{TC}}_{\text{adv}}(x') \quad \text{s.t. } \|x' - x\|_p \leq \varepsilon$, $\mathcal{L}^{\text{TC}}_{\text{adv}}(x') = \sum_{k \in T}\left(c_k\left[-\log(1 - f_k(x'))\right]\right)$, where $\mathcal{L}^{\text{TC}}_{\text{adv}}(x')$ leverages the logarithmic properties of the loss, providing numerically stable gradients [39]. Here, $c_k \in (0,1]$ represents the *normalized token contribution* of label k, which is approximated using the STS score to simulate Eq.8 in Section 5.3, since prompt engineers typically do not have direct access to the internal mechanisms of T2I models. Such weighted cross-entropy loss assigns greater emphasis to labels corresponding to high-impact modifiers, as identified by their token contribution scores. Consequently, adversarial examples are optimized to disrupt these critical features more strongly, while deprioritizing less meaningful tokens. This fine-grained adjustment allows for better resource allocation under a fixed perturbation budget ε and enhances the effectiveness of the white-box protection strategy.

H Black-box Adversarial Example-based Defense

In the black-box setting, we have no access to the internal workings of f, including its gradients, weights, or architecture. Since direct optimization of x' with respect to f is infeasible without gradient information, we adopt a two-stage approach:

1) Target Image Generation via Diffusion Model: A pre-trained conditional diffusion model is used to sample a target image x_{target} that semantically excludes the labels in T [29]. The model is conditioned on either a negative prompt (*e.g.,* "an image without cats or dogs") or a positive prompt from the complement label set $S = \{1,\ldots,K\} \setminus T$. Formally: $p_\theta(x_0|c) = \int p_\theta(x_0, x_1, \ldots, x_{T_{\text{diff}}}|c)\, dx_1 \ldots dx_{T_{\text{diff}}}$. To integrate token contributions, we adopt a *token-weighted classifier-free guidance* approach, inspired by the classifier-free guidance in conditional image generation. Let $\tilde{c}_i = c_i / \max_j c_j$ be the normalized contribution score (using STS score) for each modifier i. During each denoising step in the generative process, we apply a *cancellation factor* $\gamma_i = 1 - \tilde{c}_i$ to scale the null-conditioned deviation term. The updated denoising distribution is defined as: $\tilde{p}_\theta(x_{t-1}\,|\,x_t, \mathbf{c}) \propto p_\theta(x_{t-1}\,|\,x_t, \mathbf{c}) + w \sum_i \gamma_i \left(p_\theta(x_{t-1}\,|\,x_t) - p_\theta(x_{t-1}\,|\,x_t, c_i)\right)$.

When a modifier has low contribution ($\tilde{c}_i \to 0$), the cancellation factor $\gamma_i \to 1$, resulting in near-complete null-prompt scaling, thus minimizing its influence on the generated image. Conversely, highly contributive modifiers ($\tilde{c}_i \approx 1$) are preserved with minimal cancellation. This design ensures that visually critical modifiers are emphasized in the generation process, while semantically weak modifiers are automatically suppressed. Importantly, this technique requires no additional model queries and retains the $O(1)$ complexity of standard classifier-free guidance.

2) Latent Space Optimization via VQ-VAE: Inspired by [31], given x_{target}, we optimize x' such that it is semantically aligned in the latent space of a pre-trained VQ-VAE encoder [38]. The TokenGuard-BP (Black-box Perturbation) loss function is defined as: $\mathcal{L}_{\text{adv}}(x') = \|E(x') - E(x_{\text{target}})\|_2^2$. To maintain visual similarity to the original image x, we add a perceptual regularization term: $\mathcal{L}_{\text{perc}}(x') = \sum_l \|\phi_l(x') - \phi_l(x)\|_2^2$. The total loss function used to generate the adversarial example is: $\mathcal{L}_{\text{total}}(x') = \mathcal{L}_{\text{adv}}(x') + \lambda \mathcal{L}_{\text{perc}}(x')$, where $\lambda > 0$ balances the adversarial and perceptual objectives. This approach effectively fools the black-box classifier by aligning x' with x_{target} in semantic space, while keeping perturbations within a perceptual and norm-constrained range.