Prompt Stealing Attacks Against Text-to-Image Generation Models

Xinyue Shen   Yiting Qu   Michael Backes   Yang Zhang

CISPA Helmholtz Center for Information Security

Abstract

Text-to-Image generation models have revolutionized the artwork design process and enabled anyone to create high-quality images by entering text descriptions called *prompts*. Creating a high-quality prompt that consists of a subject and several modifiers can be time-consuming and costly. In consequence, a trend of trading high-quality prompts on specialized marketplaces has emerged. In this paper, we perform the first study on understanding the threat of a novel attack, namely *prompt stealing attack*, which aims to steal prompts from generated images by text-to-image generation models. Successful prompt stealing attacks directly violate the intellectual property of prompt engineers and jeopardize the business model of prompt marketplaces. We first perform a systematic analysis on a dataset collected by ourselves and show that a successful prompt stealing attack should consider a prompt's subject as well as its modifiers. Based on this observation, we propose a simple yet effective prompt stealing attack, PromptStealer. It consists of two modules: a subject generator trained to infer the subject and a modifier detector for identifying the modifiers within the generated image. Experimental results demonstrate that PromptStealer is superior over three baseline methods, both quantitatively and qualitatively. We also make some initial attempts to defend PromptStealer. In general, our study uncovers a new attack vector within the ecosystem established by the popular text-to-image generation models. We hope our results can contribute to understanding and mitigating this emerging threat.[1]

1 Introduction

With the advent of Stable Diffusion [56], DALL·E 2 [53], and Midjourney [8], text-to-image generation models have revolutionized the artwork design process and sparked a surge of artwork creation in mainstream media. Rather than relying on professional artists, text-to-image generation models empower anyone to produce digital images, such as photorealistic images and commercial drawings, by entering text descriptions called *prompts*. According to [24, 34, 43], a high-quality prompt that leads to a high-quality image should consist of a *subject* and several *prompt modifiers*. The subject is a natural language description of the object or scenarios depicted in the image; the prompt modifiers are keywords or key phrases that are related to specific elements or styles of the image. Figure 1 shows an example of a typical prompt and the resulting generated image. In this example, "cozy enchanted treehouse in ancient forest" is the subject and the rest phrases, e.g., "diffuse lighting," "fantasy," etc., are prompt modifiers.

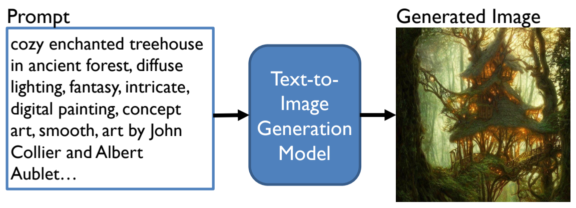

Prompt
cozy enchanted treehouse in ancient forest, diffuse lighting, fantasy, intricate, digital painting, concept art, smooth, art by John Collier and Albert Aublet…
Text-to-Image Generation Model
Generated Image

Figure 1: An image generated by Stable Diffusion and its corresponding prompt [56].

Although it seems that text-to-image generation models simplify the process of artwork design, crafting high-quality prompts is, in fact, complex and iterative. To create a desired and stable prompt, users need to constantly search for prompt modifiers and check the corresponding resulting images. Given the limited understanding of the impact of each prompt modifier, this process can be both time-consuming and costly. As a result, a new job type, namely *prompt engineer*, has emerged for people who specialize in producing high-quality prompts. Also, high-quality prompts become new and valuable commodities and are traded in specialized marketplaces, such as PromptBase [10], PromptSea [13], and Visualise AI [14]. The business model underlying these marketplaces is straightforward: customers browse sample images generated by text-to-image generation models. If they like a sample image, they can purchase the corresponding prompt. Then, customers can freely generate similar images

[1]Our code is available at https://github.com/verazuo/prompt-stealing-attack.

or modify the prompt's subject to generate other images in a similar style. Prompt marketplaces are gaining popularity. PromptBase has achieved 10K registered users until November 2022; approximately 45,000 prompts were sold by the top 50 prompt sellers (an estimated \$186,525 in total) over 9 months; new prompt trading platforms, e.g., Prompti AI, Prompt Attack, etc, continue to emerge [9, 12, 66].

In such a context, a natural research question has emerged: given an image generated by a text-to-image generation model, whether an adversary is able to infer its corresponding prompt? We name this novel attack as *prompt stealing attack*. A successful prompt stealing attack directly violates the intellectual property of prompt engineers. Moreover, it poses a significant threat to the business model of prompt marketplaces. So far, few tools in the open-source community can be tailored for stealing prompts, such as applying an image captioning model or greedily searching for the modifier combinations with the highest similarity of the image [3, 31, 39]. However, it is still unclear whether and to what extent an adversary can effectively steal prompts from generated images. In this paper, we aim to fill the gap.

Our Contributions. We perform the first large-scale study on understanding the prompt stealing attacks against images generated by text-to-image generation models. We start by collecting a large-scale dataset from Lexica [5], a well-known image gallery with 5M+ prompts and generated images from text-to-image generation models. Overall, we collect 250,000 pairs of prompts and images. After preprocessing and de-duplication on prompts, we are left with 61,467 prompt-image pairs (see Section 3.1). We name the dataset as *Lexica-Dataset*. Our quantitative and qualitative analysis on Lexica-Dataset shows that both a prompt's subject and its modifiers are important factors for the generated image's quality (see Section 3.2). This implies a successful prompt stealing attack should consider both subjects and modifiers.

Based on the findings above, we propose a simple yet effective prompt stealing attack, *PromptStealer*. PromptStealer comprises two modules: a subject generator and a modifier detector. Given a target image, the subject generator infers the stolen prompt's subject and the modifier detector predicts modifiers of the stolen prompt simultaneously. Then, the subject and the prompt modifiers are concatenated as the final stolen prompt. Experimental results on Lexica-Dataset show that PromptStealer outperforms the three baseline methods across semantic, modifier, image, and pixel similarities. For instance, PromptStealer achieves 0.70, 0.43, 0.80, and 0.90 in these metrics, respectively, while the corresponding results for the better baseline CLIP Interrogator [3] are 0.52, 0.01, 0.77, and 0.89. We also qualitatively evaluate PromptStealer and find that the images generated by stolen prompts are similar to the target images. Our further experiments on real-world prompts traded in marketplaces demonstrate similar results. Moreover, we show that the performance of PromptStealer can be further boosted by involving the adversary in the loop or providing multiple target images.

Additionally, we make attempts to mitigate prompt stealing attacks by proposing *PromptShield*. This method adds an optimized perturbation to an image relying on the technique of adversarial examples such that the adversary cannot infer the image's prompt appropriately. Experimental results show that PromptShield works well (see Section 5); however, it requires strong assumptions for the defender, and its performance is reduced by the adaptive attack.

Implications. Our work, for the first time, reveals the threat of prompt stealing in the ecosystem created by the popular text-to-image generation models. We believe that our findings can serve as a valuable resource for stakeholders to navigate and mitigate this emerging threat. Moreover, we hope to raise awareness of academia to work on the safety and security issues of the advanced text-to-image generation models. We commit to sharing our dataset and code with the research community to facilitate the research in this field.

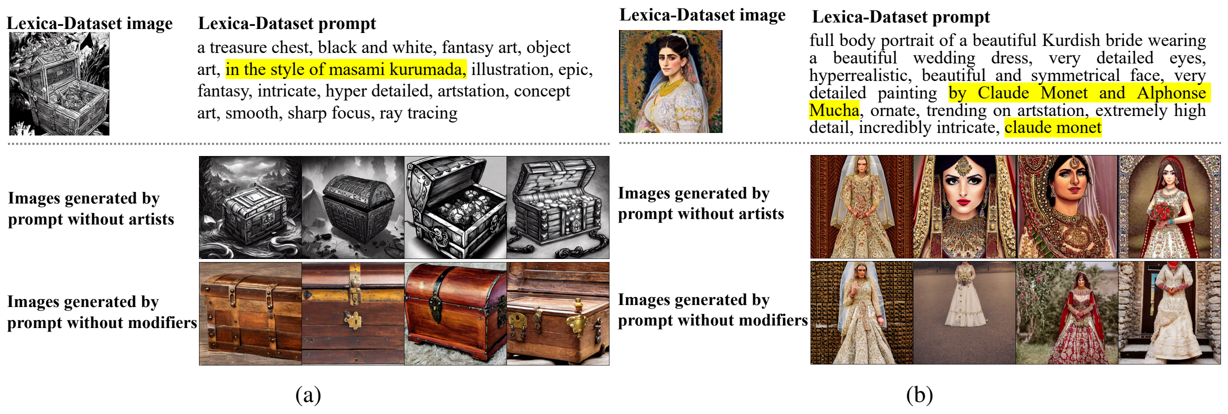
Ethics & Disclosure. According to the terms and conditions of Lexica [7], images on the website are available under the Creative Commons Noncommercial 4.0 Attribution International License. We strictly followed Lexica's Terms and Conditions, utilized only the official Lexica API for data retrieval, and disclosed our research to Lexica [6, 7]. We also responsibly disclosed our findings to prompt marketplaces such as PromptBase and PromptSea. PromptBase acknowledged our findings. They also launched a watermark editor for prompt engineers to defend against prompt stealing attacks. PromptSea explicitly discusses the risks of prompt stealing attacks in their white paper [50]. Our attack has also been recognized in Microsoft Vulnerability Severity Classification for AI Systems.[2]

2 Background

2.1 Text-to-Image Generation Models

Text-to-Image generation models aim to generate high-quality digital images with natural language descriptions, namely prompts. With the advancement of diffusion models, text-to-image generation has achieved a giant leap and gained significant attention [53, 56, 58]. Images generated by these models have been employed in various scenarios, such as children's books [48], magazine covers [22], and fashion imagery [1]. In this work, we focus on Stable Diffusion, the state-of-the-art and arguably the most popular text-to-image generation model. Besides, compared to other models like DALL·E 2 and Midjounry, Stable Diffusion is open-source and the community that promotes the usage of Stable Diffusion is very active. Since its release in March 2022, over 110K people have joined the Reddit community to share and discuss images generated by Stable Diffusion [54].

[2]https://www.microsoft.com/en-us/msrc/aibugbar.


Lexica-Dataset image
Lexica-Dataset prompt
Images generated by prompt without artists
Images generated by prompt without modifiers
(a)
(b)

a treasure chest, black and white, fantasy art, object art, in the style of masami kurumada, illustration, epic, fantasy, intricate, hyper detailed, artstation, concept art, smooth, sharp focus, ray tracing

full body portrait of a beautiful Kurdish bride wearing a beautiful wedding dress, very detailed eyes, hyperrealistic, beautiful and symmetrical face, very detailed painting by Claude Monet and Alphonse Mucha, ornate, trending on artstation, extremely high detail, incredibly intricate, claude monet

Figure 2: Two examples from Lexica-Dataset, and the corresponding generated images without artist modifiers and without any modifiers. The yellow highlight represents the artist modifiers. For the latter two cases, we generate four images with different random seeds to eliminate the potential biases.

2.2 The Prompt Marketplace

While Text-to-Image generation models have gained popularity, producing high-quality images remains a laborious and costly manual process. A user needs to constantly search for prompt modifiers and check the corresponding resulting images. One example is the logo of OctoSQL, an open-source CLI project. The designers use DALL·E 2 to create the logo and spend around \$30 interacting with the paid API [26]. As a result, prompt engineers, people who are skilled in producing high-quality prompts, start to sell their prompts in prompt marketplaces. In these marketplaces, a customer can explore sample images generated by different prompts; if they like a certain image, they can purchase the corresponding prompt through the marketplace.

Business Volume. To assess the potential impacts of prompt stealing attacks, we conducted a manual estimation of the business volume of a prompt marketplace, PromptBase. We chose PromptBase for two reasons: 1) PromptBase is the first and most widely recognized prompt marketplace, thus its business volume can represent the marketplaces to some extent; 2) PromptBase discloses the total sales of each prompt engineer. In our analysis, we calculate the sales volume of the top 50 prompt engineers over the past 9 months. We find that they collectively sold approximately 45,000 prompts, resulting in an estimated total revenue of \$186,525.

Legal Perspectives of Prompt Trading. The terms of service and white papers of prompt marketplaces [49, 50] explicitly state that all prompts on their platforms are the intellectual property of the prompt engineers and cannot be obtained through purchase. Engaging in prompt stealing attacks would, therefore, violate intellectual property rights and jeopardize the business model of the prompt marketplace.

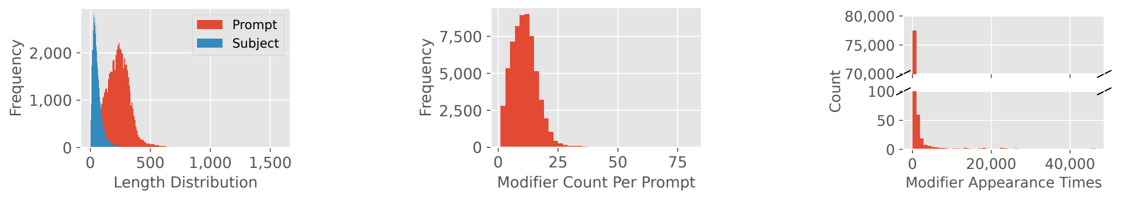
3 Preliminary Analysis

3.1 Data Collection

To the best of our knowledge, no large-scale prompt-image datasets are available at the time of our study. In order to assess prompt stealing attacks, we collect a dataset by ourselves via Lexica [5]. We opt for Lexica for two reasons. First, Lexica is a well-known image gallery of Stable Diffusion; it contains over 5M prompt-image pairs extracted from Stable Diffusion's discord server [21]. Lexica covers the art creations of many Stable Diffusion's active users, such as artists and prompt engineers. This allows us to better simulate the real-world attack scenario. Second, Lexica provides a query API [6] that given a specific prompt, returns the 50 most similar prompts and the corresponding images. Note that similar prompts are likely from the same user during their prompt engineering process. Therefore, the data from Lexica can closely reflect people's real-world usage of Stable Diffusion.

As Lexica only provides a query API and does not release its public dataset, we regard an open-source prompt dataset on Hugging Face[3] as a starting point for our data collection. This dataset contains 80K prompts crawled from Lexica, which are used to train a prompt generator. However, since this dataset does not contain any images and the image download link, we cannot directly use it. To address this, we randomly sample 5,000 prompts from the dataset and then query these prompts to Lexica's query API. As mentioned above, the API returns 50 results for each query, including images, prompts, grid, etc. In the end, we crawl 250,000 prompts and the corresponding generated images. We specifically exclude images generated by models other than Stable Diffusion, such as Aperture, a model designed to generate photorealistic images and can only be accessed on the Lexica website. We filter out the prompts and images that could not be parsed correctly. Also, images

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



(a) Subject/prompt length distribution. (b) Modifier count distribution. (c) Modifier appearance distribution.

Figure 3: General statistics of prompts in Lexica-Dataset.

Table 1: Top 10 prompt modifiers and their appearance times for each category.

No.	Trending	#	Artist	#	Medium	#	Movement	#	Flavor	#
1	artstation	46,357	greg rutkowski	11,552	concept art	24,095	fantasy art	1,567	highly detailed	26,818
2	cgsociety	3,822	artgerm	7,273	digital art	7,483	art nouveau	1,132	8k	22,987
3	deviantart	2,309	artgerm and greg rutkowski and alphonse mucha	6,237	vector art	127	hyperrealism	1,041	sharp focus	22,774
4	pixiv	1,018	wlop	5,989	an ultrafine detailed painting	83	photorealism	924	digital painting	18,211
5	pinterest	646	ilya kuvshinov	3,918	a detailed matte painting	63	street art	447	intricate	18,002
6	behance	321	alphonse mucha	3,716	poster art	58	surrealism	388	illustration	17,473
7	instagram	127	rossdraws	2,802	an oil painting	51	romanticism	384	octane render	14,394
8	zbrush central	57	craig mullins	2,341	pixel art	42	art deco	377	smooth	13,906
9	cg society	52	james jean	2,133	a detailed painting	40	realism	292	elegant	13,081
10	polycount	51	rhads	1,976	a character portrait	39	rococo	290	4k	10,325

belonging to the grid type (each of those images contains several generated images stitched together) are neglected as well. Finally, after the de-duplication of prompts, we get 61,467 prompt-image pairs. We name the dataset as *Lexica-Dataset*.

As mentioned before, each prompt consists of a subject and several prompt modifiers. Thus, we further decompose prompts in Lexica-Dataset into subjects and prompt modifiers. Concretely, we split each prompt by commas. The first part is regarded as the subject according to [34, 43], while the rest are treated as prompt modifiers. We standardize the format of some modifiers, e.g., “3 d” to “3d.” We also remove the style-evocation words, including “in the style of,” “inspired by,” “trending on,” etc. In the end, we obtain 77,616 prompt modifiers. Figure 2 depicts some samples from the dataset.

3.2 Data Analysis

Subjects and Prompt Modifiers. Figure 3a depicts the length distribution of subjects and prompts in Lexica-Dataset. On average, the length of a subject is 56 characters, while the length of a prompt is 237 characters. This means a subject takes only 23.63% of a prompt, and the rest is assembled with prompt modifiers. Figure 3b further depicts the distribution of modifier count per prompt. On average, each prompt contains 11 modifiers. This might suggest prompt modifiers play an essential role in the image generation process. To further investigate this, Figure 2 also shows some generated images without prompt modifiers. As we can see, modifiers indeed largely influence image quality in terms of style and details. Take the treasure chest as an example (see Figure 2a), if the modifiers are not considered, Stable Diffusion generates a plain treasure chest. Meanwhile, by considering various modifiers, the original image depicts a more stylized treasure chest with many fine-grained details. These results imply that a successful prompt stealing attack should not only consider the target prompt’s subject but also put efforts into recovering the modifiers. In other words, using an image captioning model to get the caption (subject) of a target image is insufficient for prompt stealing. Experiment results in Section 4 further confirm this. On the other hand, we emphasize that the subject also plays an essential role in a prompt as it describes the main content of the image.

For the modifiers, we further show the distribution of their appearance times in Figure 3c. We can observe a Pareto distribution. Specifically, Lexica-Dataset contains 61,467 prompts and 77,616 modifiers. Among them, only 7,672 (9.88%) modifiers are used more than ten times, and 1,109 (1.43%) modifiers are used more than 100 times. From the adversary's perspective, this Pareto distribution eases the requirements for prompt stealing attacks, as a relatively small modifier set can already cover most modifiers used in prompts.

Modifiers in Category-Level. We further attribute prompt modifiers into different categories. The open-source prompt engineering tool CLIP Interrogator [3] defines five categories for modifiers, i.e., trending, artist, medium, movement, and flavor, and offers a relatively complete modifier list for each category. In our analysis, we adopt the same set of categories and assign all modifiers in Lexica-Dataset to them. Concretely, a modifier in Lexica-Dataset is considered as part of a category if it belongs to the corresponding modifier list from CLIP Interrogator. Table 1 shows the top 10 prompt modifiers with respect to their appearance times in each category of Lexica-Dataset. We find that users are less likely to use movement modifiers. The highest one is "fantasy art" with appearance times being only 1,567. This is probably because the usage of movement modifiers requires background knowledge of art. Conversely, the top three modifiers in the flavor category are all well-known image quality boosters [43], such as "highly detailed" and "8k." On average, 14.32%, 6.11%, 3.25%, 0.98%, and 75.33% of the modifiers in a prompt belong to the artist, trending, medium, movement, and flavor categories, respectively. Excluding the flavor category which contains various kinds of modifiers, the artist category occupies the largest proportion of prompts compared to the rest. This reveals that users are more likely to use artist modifiers to lead the generated images to certain visual styles. Figure 2 further shows some examples of generated images without artist modifiers. In Figure 2b, the artist modifiers "Claude Monet" and "Alphonse Mucha" indeed heavily influence the final outlook and fine-grained details of the original image, compared to the images without any artist modifiers. A similar observation can be made in the example depicted in Figure 2a.

Next, we investigate the semantic relations among modifiers of different categories. Diffusion-based text-to-image generation models like Stable Diffusion rely on CLIP [52] to obtain text embeddings of prompts. We thereby utilize CLIP's text encoder to obtain modifiers' embeddings and visualize them via a T-SNE [67] plot (see Figure 4). For each category, we consider the top 20 modifiers with respect to their appearance times. We find that the modifiers in the artist category are tightly grouped and distant from others. Meanwhile, the other four categories are relatively mixed. This again shows the importance of the artist modifiers in the image generation process. Meanwhile, it is expected that other categories are close to each other. For instance, movement modifiers are often accompanied by medium modifiers. This has also been uncovered in a previous qualitative study [34] and recommended by DALL·E 2 prompt book [46].

Takeaways. To summarize, we have established a prompt-image dataset Lexica-Dataset that can closely reflect people's real-world usage of Stable Diffusion. Our data analysis reveals that besides its subject, a prompt's modifiers also play an essential role in determining its generated image's outlook. This suggests a successful prompt stealing attack should consider both the subject and modifiers of a prompt. Moreover, we attribute prompt modifiers into five different categories and discover that the artist modifiers heavily influence generated images' styles and details.

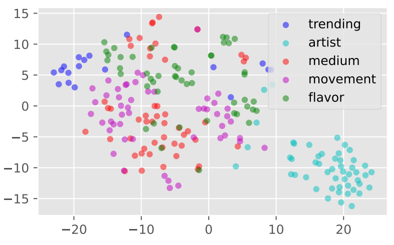


Figure 4: T-SNE visualization of modifier embeddings with respect to different categories.

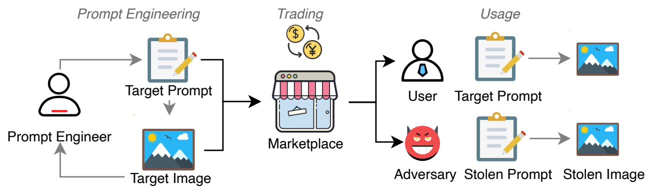


Figure 5: The scenario of prompt trading and stealing.

4 Prompt Stealing Attack

4.1 Threat Model

Scenario. Our attack is designed for the scenario shown in Figure 5. First, a prompt engineer performs an iterative exploring process to find an ideal prompt that can lead to a high-quality image using a text-to-image generation model. This prompt is referred to as the *target prompt*. Then, the prompt engineer can sell the prompt via marketplaces like PromptBase [10], by providing an example image generated by the target prompt, namely *target image*. If a regular user is interested in the style of the target image, the user can get the target prompt by trading with the prompt engineer through the marketplace. After getting the target prompt, one of the usage scenarios for the user is to modify the prompt's subject to generate other images with a similar style (see Section 2).

Conversely, an adversary aims to steal the target prompt from a target image. We refer to the prompt stolen by the adversary as *stolen prompt*. Moreover, the adversary can feed the stolen prompt to the text-to-image generation model again, and the generated image here is referred to as *stolen image*.

Adversary's Goal. Given a target image generated by a text-to-image generation model, the goal of the adversary is to steal the target prompt that leads to the target image. The

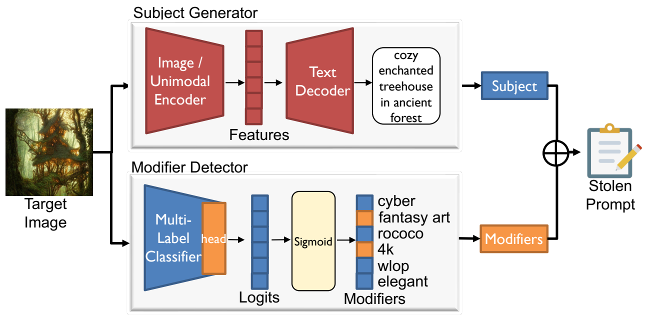


Figure 6: Overview of PromptStealer.

stolen prompt should ideally satisfy four quantitative goals.

- **Semantic Similarity.** The stolen prompt should share high semantic similarity with the target prompt, normally measured in the text embedding space. This is important because text-to-image generation models essentially condition on the text embeddings during the image generation. Here, the semantic similarity considers both subjects and prompt modifiers.
- **Modifier Similarity.** The stolen prompt should contain as many modifiers of the target prompt as possible. As the modifiers take an essential role in guaranteeing the quality of the generated image (see Section 3), higher modifier similarity results in better attack performance.
- **Image / Pixel Similarity.** By feeding the stolen prompt to the text-to-image generation model again, the adversary expects the model to generate a similar stolen image to the target image. Here, the similarity can be regarded as both image semantic and pixel-level similarity.

Besides, considering the usage of target prompt mentioned in Section 2, the stolen prompt should also lead to images depicting different subjects in style of the target image. Therefore, we also qualitatively assess PromptStealer from this angle.
Adversary's Capability. We assume the adversary's capabilities in a real-world setting. First, the adversary can collect public prompts and the corresponding images via online services like Lexica. Second, the adversary has black-box access to the target text-to-image generation model.

4.2 PromptStealer

In this work, we introduce PromptStealer, a simple yet effective prompt stealing attack. The design principle of PromptStealer is based on our observations in Section 3, i.e., a successful prompt stealing attack should focus on both the subject and modifiers of a target prompt. Specifically, PromptStealer consists of two modules: a subject generator and a modifier detector. Figure 6 shows the overview of PromptStealer.
Subject Generator. Given the target image, the subject generator aims to generate the subject of the stolen prompt. In this work, we adopt the image captioning framework for this purpose, which typically consists of an image/unimodal encoder and a text decoder [31, 68]. Concretely, the encoder first converts the target image to a fixed-length feature representation. This representation is then used in a text decoder to generate a caption/subject for the target image. Note, existing image captioning models are not directly suitable for subject generation for two main reasons. Firstly, they are trained on normalized data and are prone to use generic terms, such as referring to a specific celebrity as "a woman" or labeling a "terrier dog" simply as "a dog." In contrast, the subject of a high-quality prompt demands more specific and detailed information, such as race and landmarks. Secondly, image captioning models are trained to describe real-world pictures; however, the target images often showcase various art styles. To bridge this gap, we fine-tune the encoder as well as the decoder on the subjects and corresponding images in Lexica-Dataset. We evaluate two representative image captioning frameworks, i.e., BLIP and GIT, in our experiments (see Section 4.6).
Modifier Detector. Modifiers are keywords or key phrases related to specific elements or styles of the target image. A high-quality prompt typically includes numerous and varied modifiers that complexly interact with each other to dictate the style of the generated image. The goal of the modifier detector is thus to detect modifiers in a given target image. A similar task in the CV domain is object detection, which leverages a multi-label classifier to predict all objects in one given image simultaneously. Existing multi-label classifiers commonly consist of a backbone model and a multi-label classification head. The backbone model outputs an image representation, and the classification head transforms the representation into prediction logits. However, existing multi-label classifiers cannot be directly applied in the scenario of modifier detection. This is because they are primarily designed to output labels like "cat," "dog," and "airplanes," which are not modifiers such as artist names or painting styles. To adapt multi-label classifiers to our scenario, we reset the multi-label head layer of the multi-label classifier with Lexica-Dataset's modifier set and then fine-tune the whole model on the modifier sets and corresponding images in Lexica-Dataset. In addition, we apply a Sigmoid activation layer at the end of the multi-label classifiers to normalize the outputs to prediction posteriors. The labels/modifiers whose posteriors are higher than a predefined threshold are regarded as the stolen prompt's modifiers.

After applying the subject generator and the modifier detector to the target image, we concatenate the obtained subject and the prompt modifiers together as the final stolen prompt.

4.3 Experimental Settings

Text-to-Image Generation Model. We focus on Stable Diffusion as it is one of the most prevalent text-to-image generation models in the field. Besides, the open-source nature and the active community support (e.g., Lexica) of this model enable us to perform a large-scale evaluation. Most of our experiments are directly performed on Lexica-Dataset. In certain cases, we also need to use Stable Diffusion to generate images for evaluation. To this end, we adopt the official Stable Diffusion v1.4 model.[4] For each image generation process, we sample 50 steps with default settings [69]. The size of the generated image is 512×512. We also consider other representative text-to-image generation models, like DALL·E 2 and Midjounery. However, since they are not open-source and only provide non-free APIs, collecting large-scale datasets from them is financially infeasible for us. As an alternative, we directly apply PromptStealer trained on Lexica-Dataset (based on Stable Diffusion) to them as a case study to test our approach's generalizability (see Section 4.8).

PromptStealer. PromptStealer consists of two modules: a subject generator and a modifier detector. For the subject generator, we evaluate two representative image captioning frameworks, i.e., BLIP [31] and GIT [68]. Concretely, we first adopt the BLIP and GIT model pre-trained on MS-COCO dataset and then follow their default settings to fine-tune them on the subjects and corresponding images of Lexica-Dataset. Regarding the modifier detector, we assess two multi-label frameworks, ML-Decoder [55] and Query2Label [33]. Note, we have in total of 77,616 prompt modifiers in Lexica-Dataset, which can all be treated as labels for the multi-label classifier. However, many modifiers only appear a few times (see Figure 3c). Therefore, in our main experiments, we only consider modifiers that appear more than 10 times (7,672 modifiers) as labels. We choose 0.6 as the threshold to decide whether a label/modifier is in the modifier set of the stolen prompt. The impacts of different label numbers and thresholds are investigated in Section 4.6. PromptStealer is trained on 80% of the samples in Lexica-Dataset, and the rest samples are used for testing.

Baseline Methods. We consider three baseline attack methods in our experiments. The first baseline is a pre-trained image captioning model. Given an image, the model produces a caption for the image, which is directly regarded as the stolen prompt. Here, we regard BLIP pre-trained on MS-COCO dataset as the first baseline. We further fine-tune the first baseline model on Lexica-Dataset as our second baseline. Besides, we also include an open-source prompt engineering tool, CLIP Interrogator [3] in our evaluation. This method iteratively calculates the similarity between the combinations of modifiers and the target image. Once the similarity stops rising, it regards the current combination as the stolen prompt. Before comparison, we evaluate the performance of CLIP Interrogator with two modifier sets, i.e., the same modifier set as PromptStealer (7,637 modifiers) and the complete modifier set of Lexica-Dataset (77,616 modifiers). As shown in our technique report [62], CLIP Interrogator with larger prompt modifiers demonstrates slightly better performance, therefore we utilize CLIP Interrogator with the complete modifier set as the third baseline method.

Evaluation Metric. As discussed in Section 4.1, the adversary has four quantitative goals: semantic similarity, modifier similarity, image similarity, and pixel similarity. Thus, we adopt four quantitative metrics for these goals, respectively.

- **Semantic Similarity.** The semantic similarity is the cosine similarity between the embeddings of the target prompt and the stolen prompt. We use CLIP's text encoder to get the embeddings.
- **Modifier Similarity.** The modifier similarity is the Jaccard similarity between the modifiers of the target prompt and those of the stolen prompt.
- **Image Similarity.** The image similarity is the cosine similarity between the embeddings of the target and stolen images, which is a widely adopted metric to measure similarity between images, both at the semantic level and the visual/pixel level [60]. Following previous works [15, 60], we rely on CLIP's image encoder to obtain an image's embedding. For each stolen prompt, we generate four stolen images from Stable Diffusion (by varying the random seed) to eliminate the potential biases. We then calculate the similarity between each stolen image and the target image and, in the end, average the results.
- **Pixel Similarity.** The pixel similarity is a traditional metric to quantify the pixel differences between the target image and the stolen, which are commonly calculated by the complement of the mean squared error (MSE) [70].

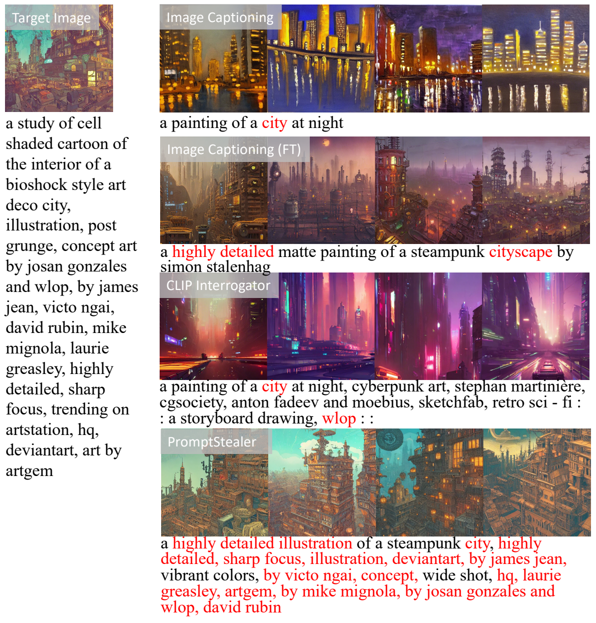
We acknowledge that any metric has limitations. To perform a comprehensive assessment, we also provide qualitative evaluations to assess the similarity perceived by the end-users.

- **Human-Rated Similarity.** The human-rated similarity refers to the perceived similarity between target and stolen images by end-users. Specifically, for each target image and its corresponding stolen images, two labelers are assigned to label it using a 5-level Like-scaler, ranging from "not similar at all" to "very similar." The detailed criteria for each level is stated in Table 11 in Appendix. We randomly sample 10 target images in our test set and report the mean value.

4.4 Quantitative Evaluation

Table 2 shows the performance of PromptStealer together with the three baselines. We first observe that the image captioning model itself is not sufficient to achieve a successful prompt stealing attack. Even fine-tuned on the Lexica-Dataset, it only achieves 0.45, 0.14, 0.74, and 0.89 in semantic, modifier, image and pixel similarities. To compare, PromptStealer

[4] https://huggingface.co/CompVis/stable-diffusion-v1-4.



a study of cell shaded cartoon of the interior of a bioshock style art deco city, illustration, post grunge, concept art by josan gonzales and wlop, by james jean, victo ngai, david rubin, mike mignola, laurie greasley, highly detailed, sharp focus, trending on artstation, hq, deviantart, art by artgem

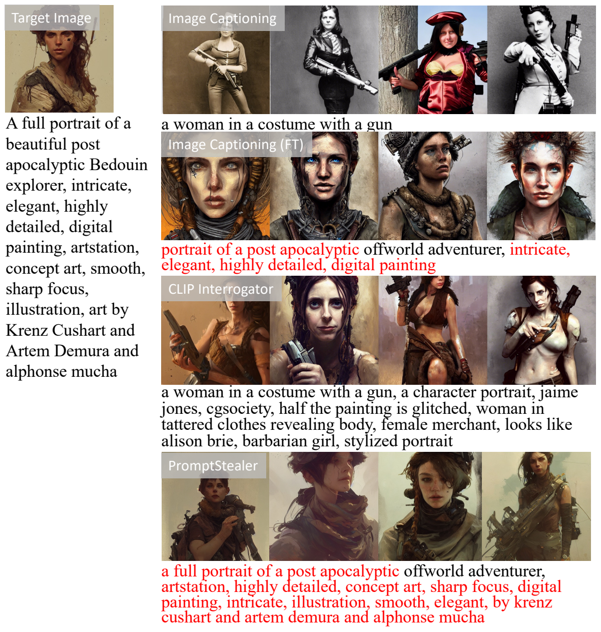
a painting of a city at night

a highly detailed matte painting of a steampunk cityscape by simon stalenhag

a painting of a city at night, cyberpunk art, stephan martinière, cgsociety, anton fadeev and moebius, sketchfab, retro sci - fi : : a storyboard drawing, wlop : :

a highly detailed illustration of a steampunk city, highly detailed, sharp focus, illustration, deviantart, by james jean, vibrant colors, by victo ngai, concept, wide shot, hq, laurie greasley, artgem, by mike mignola, by josan gonzales and wlop, david rubin

(a)



A full portrait of a beautiful post apocalyptic Bedouin explorer, intricate, elegant, highly detailed, digital painting, artstation, concept art, smooth, sharp focus, illustration, art by Krenz Cushart and Artem Demura and alphonse mucha

a woman in a costume with a gun

portrait of a post apocalyptic offworld adventurer, intricate, elegant, highly detailed, digital painting

a woman in a costume with a gun, a character portrait, jaime jones, cgsociety, half the painting is glitched, woman in tattered clothes revealing body, female merchant, looks like alison brie, barbarian girl, stylized portrait

a full portrait of a post apocalyptic offworld adventurer, artstation, highly detailed, concept art, sharp focus, digital painting, intricate, illustration, smooth, elegant, by krenz cushart and artem demura and alphonse mucha

(b)

Figure 7: Two attack examples of PromptStealer and the three baselines. The text below each image is the target/stolen prompt.

achieves 0.70, 0.43, 0.80, and 0.90 in these metrics, respectively, while the corresponding results for the better baseline CLIP Interrogator are 0.52, 0.01, 0.77, and 0.89. We also notice that the modifier similarity of PromptStealer is not as high as other metrics. This is primarily because the modifier similarity is based on exact word matching, which is typically not high in the NLP domain [25]. For example, modifiers like "digital art" and "digital painting" are not considered a match despite their high semantic similarity. To address this, we offer semantic similarity as an additional metric. We further observe that the pixel similarity is not very distinguishable in this scenario. For instance, PromptStealer performs slightly better than CLIP Interrogator (0.90 vs. 0.89 in pixel similarity). This might be due to the pixel similarity metric itself. Since it is sensitive to changes in pixel values, it does not necessarily consider the human perception of similarity. We have evaluated other pixel-level metrics such as SSIM [70] and pHash [63]. The performance is similar. Therefore, we introduce human-rated similarity in the qualitative evaluation.

4.5 Qualitative Evaluation

Table 2 shows the human-rated similarity of PromptStealer and three baselines. The Cronbach's Alpha among the labelers is 0.90 on average, demonstrating good inter-reliability. We observe that PromptStealer gets far better human-rated similarity compared with other methods. For instance, the human-rated similarity for ImgCap, ImgCap (FT), Clip Interrogator, and PromptStealer are 1.65, 3.20, 2.95, and 4.45.

Figure 7 further shows two attack examples of PromptStealer and the baselines. Here, for each stolen prompt, we let Stable Diffusion generate four images with different random seeds to eliminate the potential biases. In both cases, we can see that the stolen images by PromptStealer are more

Table 2: The performance of PromptStealer and the three baselines on Lexica-Dataset. ImgCap refers to the image captioning method; ImgCap (FT) represents the image captioning method fine-tuned on Lexica-Dataset. Human refers to human-rated similarity.

Method	Semantic	Modifier	Image	Pixel	Human
ImgCap	0.19	0.00	0.65	0.89	1.65
ImgCap (FT)	0.45	0.14	0.74	0.89	3.20
CLIP Interrogator	0.52	0.01	0.77	0.89	2.95
PromptStealer	**0.70**	**0.43**	**0.80**	**0.90**	**4.45**

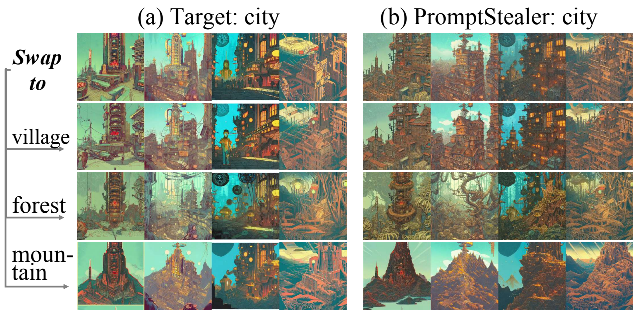


Figure 8: Transferability results of the example in Figure 7a.

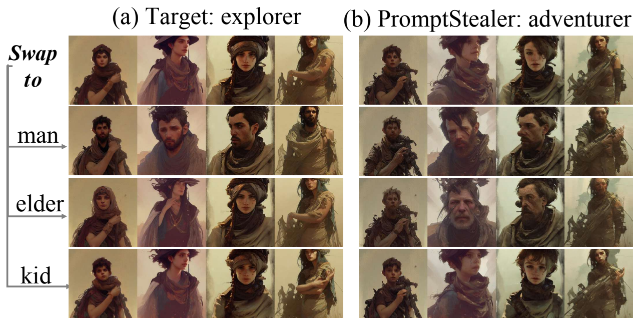


Figure 9: Transferability results of the example in Figure 7b.

similar to the target images compared to the baselines. Take Figure 7b as an example, the stolen images by the image captioning method are the least similar to the target image. Concretely, the first, second, and fourth stolen images by image captioning are in vintage style, while the third image has some modern features with respect to color and clothing fabric. However, none of the images share a similar style to the target image. With modifiers introduced, the qualities of the stolen images generated by CLIP Interrogator improved a lot. All four images show the same vintage yellow tint as the target image. However, the wrong modifiers steer the stolen images in the wrong direction, thus reducing the efficacy of CLIP Interrogator. For example, the modifier "looks like alison brie," which is not part of the target prompt, directs the results toward the celebrity Alison Brie (see the first, second, and fourth images). To compare, we find that PromptStealer is able to recover a large proportion of the modifiers of the target prompt. The large coverage of modifiers thus promises coherence between the target and stolen images.

As mentioned before, an important usage of the prompt traded through the marketplace is that the user can modify its subject to generate other images with a similar style. We refer to this usage as transferability. Ideally, the stolen prompt should have high transferability as well. To validate the transferability of the stolen prompts, we continue using two examples in Figure 7, and compare the generated images when the subject in the prompt is replaced with new ones. For example, we replace "city" in the subject of the stolen prompt with new ones such as "village," "forest," and "mountain." We also provide the generated images using the target prompts, which are replaced with the same new subjects. Figure 8 and Figure 9 showcase the transferability results. We find that with the stolen prompts, the adversary can generate images that adapt well to other subjects while maintaining similar features to those generated using target prompts, indicating that the stolen prompts by PromptStealer have high transferability.

Table 3: Performances of subject generators. (FT) denotes that the model is fine-tuned on the subjects and corresponding images of Lexica-Dataset.

	Semantic	Modifier	Image	Pixel	Human
GIT	0.62	**0.43**	0.72	0.89	3.20
GIT (FT)	0.68	**0.43**	0.79	**0.90**	4.25
BLIP	0.66	**0.43**	0.79	**0.90**	3.55
BLIP (FT)	**0.70**	**0.43**	**0.80**	**0.90**	**4.45**

Table 4: Performances of different modifier detectors.

	Semantic	Modifier	Image	Pixel	Human
Query2Label	0.63	0.34	0.75	0.89	4.00
ML-Decoder	**0.70**	**0.43**	**0.80**	**0.90**	**4.45**

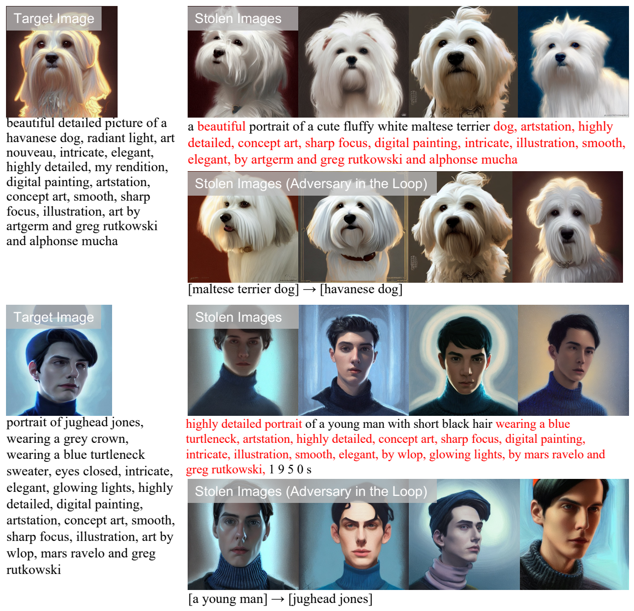
4.6 Ablation Study

We further evaluate whether PromptStealer is still effective with 1) different subject generators; 2) different modifier detectors; 3) label number; and 4) posterior threshold.

Subject Generator. Table 3 shows the performance of different subject generators. Regarding the model architectures, we find that BLIP performs slightly better than GIT in the prompt stealing scenario. Besides, it is clear to observe an increase for both of the two models across all metrics after fine-tuning on Lexica-Dataset. For instance, BLIP obtains an increase of 0.90 in human-rated similarity after fine-tuning and exhibits the best performance among all other models. Therefore, we utilize the fine-tuned BLIP as our subject generator in the following experiments.

Modifier Detector. We consider two modifier detector architectures, i.e., ML-Decoder and Query2Label, in our experiments. As illustrated in Table 4, we observe that ML-Decoder achieves better performance than Query2Label, as evidenced by the higher scores across all evaluation metrics. This can be attributed to the group-decoding scheme of ML-Decoder. Different from Query2Label which assigns a query per class, ML-Decoder uses a fixed number of queries and interpolates to reach the final number of classes using a group fully-connected block.

Label Number. In the main experimental setting, we only consider modifiers appearing more than 10 times in Lexica-Dataset as the labels (7,672) for the multi-label classifier. We are interested in whether changing the label number affects the attack performance. To this end, we consider two other variants, i.e., modifiers appear more than 50 times and 100 times which lead to 1,966 labels and 1,109 labels, respectively. Figure 17a in the Appendix shows the results. We find that for semantic, modifier, image, and pixel similarities, PromptStealer with 1,109 labels and 1,966 labels have slightly weaker performance than PromptStealer with 7,672 labels. For instance, the semantic similarities for 1,109, 1,966, and 7,672 labels are 0.67, 0.68, and 0.70, respectively. This is


Target Image
Stolen Images
Stolen Images (Adversary in the Loop)
Target Image
Stolen Images
Stolen Images (Adversary in the Loop)

beautiful detailed picture of a havanese dog, radiant light, art nouveau, intricate, elegant, highly detailed, my rendition, digital painting, artstation, concept art, smooth, sharp focus, illustration, art by artgerm and greg rutkowski and alphonse mucha

a beautiful portrait of a cute fluffy white maltese terrier dog, artstation, highly detailed, concept art, sharp focus, digital painting, intricate, illustration, smooth, elegant, by artgerm and greg rutkowski and alphonse mucha

[maltese terrier dog] → [havanese dog]

portrait of jughead jones, wearing a grey crown, wearing a blue turtleneck sweater, eyes closed, intricate, elegant, glowing lights, highly detailed, digital painting, artstation, concept art, smooth, sharp focus, illustration, art by wlop, mars ravelo and greg rutkowski

highly detailed portrait of a young man with short black hair wearing a blue turtleneck, artstation, highly detailed, concept art, sharp focus, digital painting, intricate, illustration, smooth, elegant, by wlop, glowing lights, by mars ravelo and greg rutkowski, 1 9 5 0 s

[a young man] → [jughead jones]

Figure 10: Two examples of improving PromptStealer by manually modifying subjects, i.e., adversary in the loop. The red color marks the correctly predicted modifiers.

reasonable as a relatively small modifier set can already cover most modifiers in target prompts, as shown in Figure 3c.

Threshold. Figure 17b in the Appendix shows the results regarding the impact of the posterior threshold for the multi-label classifier. We find that PromptStealer obtains the highest modifier, image, and pixel similarity when the threshold is set to 0.6. For semantic similarity, the highest result (0.71) is achieved when the threshold is 0.3, which is very close to the semantic similarity (0.70) when the threshold is 0.6. In conclusion, 0.6 is a suitable threshold for the multi-label classifier of PromptStealer.

4.7 Adversary in the Loop

We acknowledge that PromptStealer is not perfect. Figure 10 shows some failed cases of PromptStealer. We find that the main reason for these failed cases to happen is that PromptStealer cannot accurately capture the key subjects from the target images, such as celebrities and animals. However, the adversary can easily improve this by manually modifying the stolen prompt with their knowledge, e.g., replacing "a young man" with "jughead jones," or modifying the misidentified "maltese terrier dog" to "havanese dog." Examples in Figure 10 show the effects of involving the adversary in the loop. As we can see, the attack results improve significantly. Besides involving an adversary in the loop, more advanced machine learning models can be used to solve the issue as well, like adding a subject detector when generating subjects. We leave this as future work.

Table 5: The performance of PromptStealer and baseline methods on DiffusionDB.

Method	Semantic	Modifier	Image	Pixel	Human
ImgCap	0.28	0.00	0.69	0.88	1.65
ImgCap (FT)	0.50	0.22	0.79	0.88	3.25
CLIP Interrogator	0.51	0.01	0.79	0.88	2.15
PromptStealer	**0.64**	**0.30**	**0.82**	**0.89**	**3.95**

Table 6: The performance of PromptStealer and baseline methods on real-world traded prompts.

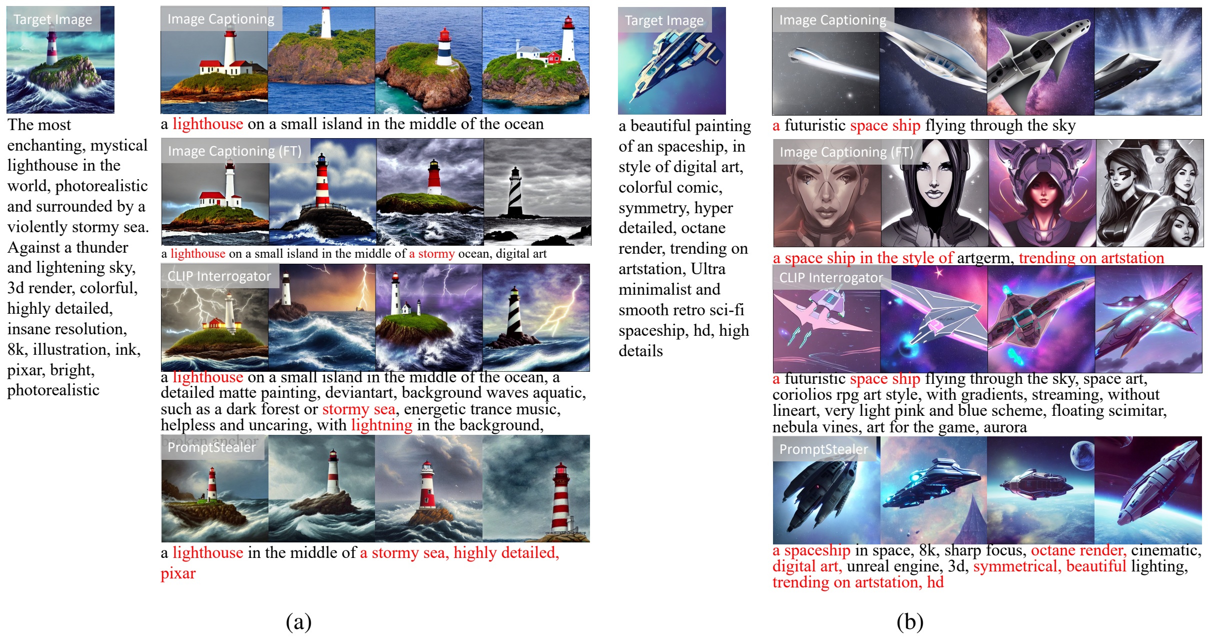
Method	Semantic	Modifier	Image	Pixel	Human
ImgCap	0.44	0.00	0.80	**0.89**	2.85
ImgCap (FT)	0.55	0.08	**0.83**	0.88	3.35
CLIP Interrogator	0.56	0.00	0.80	0.88	3.40
PromptStealer	**0.63**	**0.22**	**0.83**	**0.89**	**4.05**

4.8 Open-World Evaluation

Other Datasets. So far, all the testing samples for PromptStealer are from Lexica-Dataset. To simulate more realistic attack scenarios, we evaluate whether PromptStealer can be generalized to target images from out-of-distribution datasets. *DiffusionDB*. DiffusionDB [4] is a recent open-source prompt-image dataset and we randomly sample 1k prompts and corresponding images for evaluation. We find PromptStealer outperforms the three baselines both quantitatively and qualitatively. As illustrated in Table 5, PromptStealer achieves 0.64, 0.30, 0.82, and 0.89 on semantic, modifier, image, and pixel similarity, respectively. Regarding qualitative performance, PromptStealer also obtains the highest human-rated similarity with a score of 3.95. See our technical report [62] for the qualitative attack examples.

Real-world Traded Prompts. Additionally, we apply PromptStealer to 10 randomly selected/purchased real-world prompts traded on two prompt marketplaces, PromptBase [10] and PromptDB [11]. Table 6 shows that PromptStealer consistently outperforms baseline methods both quantitatively and qualitatively. As Figure 11 shows, PromptStealer successfully identifies the main subject and modifiers in the target prompts. For instance, for the first target image, PromptStealer can successfully infer "lighthouse" and "a stormy sea" in the subject and the modifiers "highly detailed" and "pixar." Moreover, the stolen images exhibit a strong resemblance to the target image, indicating the capability of PromptStealer in open images.

Other Text-to-Image Generation Models. Midjourney and DALL·E 2 are also mainstream text-to-image generation models. However, as mentioned before, the two models are still not open-source, so we could not conduct large-scale quantitative experiments on them. Instead, we perform a case study and hope it could shed light on the generalizability of PromptStealer on these unseen models. Concretely, we first use two target prompts from Lexica-Dataset to generate two target



(a) Target Image: The most enchanting, mystical lighthouse in the world, photorealistic and surrounded by a violently stormy sea. Against a thunder and lightening sky, 3d render, colorful, highly detailed, insane resolution, 8k, illustration, ink, pixar, bright, photorealistic

Image Captioning: a lighthouse on a small island in the middle of the ocean

Image Captioning (FT): a lighthouse on a small island in the middle of a stormy ocean, digital art

CLIP Interrogator: a lighthouse on a small island in the middle of the ocean, a detailed matte painting, deviantart, background waves aquatic, such as a dark forest or stormy sea, energetic trance music, helpless and uncaring, with lightning in the background, broken anchor

PromptStealer: a lighthouse in the middle of a stormy sea, highly detailed, pixar

(b) Target Image: a beautiful painting of an spaceship, in style of digital art, colorful comic, symmetry, hyper detailed, octane render, trending on artstation, Ultra minimalist and smooth retro sci-fi spaceship, hd, high details

Image Captioning: a futuristic space ship flying through the sky

Image Captioning (FT): a space ship in the style of artgerm, trending on artstation

CLIP Interrogator: a futuristic space ship flying through the sky, space art, coriolios rpg art style, with gradients, streaming, without lineart, very light pink and blue scheme, floating scimitar, nebula vines, art for the game, aurora

PromptStealer: a spaceship in space, 8k, sharp focus, octane render, cinematic, digital art, unreal engine, 3d, symmetrical, beautiful lighting, trending on artstation, hd

Figure 11: Two attack examples of PromptStealer and the three baselines on real-world traded prompts. The first one is from PromptDB, and the latter one is from PromptBase. The red color marks the correctly predicted modifiers.

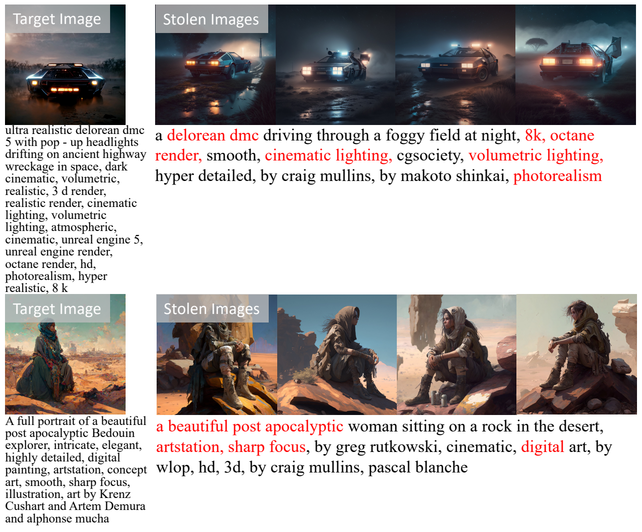
Table 7: Performances across Stable Diffusion versions.

Version	Semantic	Modifier	Image	Pixel	Human
v1.4	**0.63**	0.43	0.79	0.90	**4.5**
v1.5	**0.63**	**0.45**	0.79	**0.91**	4.3
v2.0	0.53	0.32	**0.82**	0.88	4.25

images from Midjourney and DALL·E 2. We then directly apply the trained PromptStealer to steal the prompts from the two target images and use the corresponding text-to-image generation models again to get the stolen images. The results are displayed in Figure 12 and Figure 13.

We find that even though PromptStealer has never seen images generated from Midjourney and DALL·E 2 before, it still manages to catch certain key prompt modifiers, such as “octane render” and “artstation.” This indicates that PromptStealer still works on these unseen text-to-image generation models to a certain extent. We also notice that PromptStealer does not perform as well as it does on Stable Diffusion. For example, it cannot accurately deduce the artist modifiers. This could be caused by the different architectures and weights of these text-to-image generation models. One possible solution is to train PromptStealer on multiple datasets related to different models like Midjourney and DALL·E 2. However, as no dataset is available by the time we perform this study, we leave this as future work.

Newer Versions of Stable Diffusion. With the advancement of Stable Diffusion, it is also interesting to observe whether PromptStealer works on different versions of Stable Diffusion.



Target Image: ultra realistic delorean dmc 5 with pop - up headlights drifting on ancient highway wreckage in space, dark cinematic, volumetric, realistic, 3 d render, realistic render, cinematic lighting, volumetric lighting, atmospheric, cinematic, unreal engine 5, unreal engine render, octane render, hd, photorealism, hyper realistic, 8 k

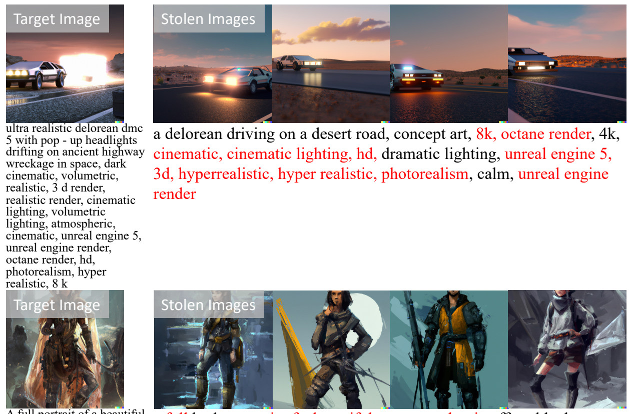
Stolen Images: a delorean dmc driving through a foggy field at night, 8k, octane render, smooth, cinematic lighting, cgsociety, volumetric lighting, hyper detailed, by craig mullins, by makoto shinkai, photorealism

Target Image: A full portrait of a beautiful post apocalyptic Bedouin explorer, intricate, elegant, highly detailed, digital painting, artstation, concept art, smooth, sharp focus, illustration, art by Krenz Cushart and Artem Demura and alphonse mucha

Stolen Images: a beautiful post apocalyptic woman sitting on a rock in the desert, artstation, sharp focus, by greg rutkowski, cinematic, digital art, by wlop, hd, 3d, by craig mullins, pascal blanche

Figure 12: Two attack examples of PromptStealer in open-world evaluation (other model Midjourney). The target and stolen images are generated by Midjourney. The red color marks the correctly predicted modifiers.

Note that the data we collected contains no model version information. Therefore, given a target prompt, we utilize a certain version of Stable Diffusion to generate the target image and then apply PromptStealer to steal its prompts. We test three versions of Stable Diffusion: v1.4, v1.5, and v2.0. Table 7 shows the results. Overall, PromptStealer performs



ultra realistic delorean dmc 5 with pop - up headlights drifting on ancient highway wreckage in space, dark cinematic, volumetric, realistic, 3 d render, realistic render, cinematic lighting, volumetric lighting, atmospheric, cinematic, unreal engine 5, unreal engine render, octane render, hd, photorealism, hyper realistic, 8 k

a delorean driving on a desert road, concept art, 8k, octane render, 4k, cinematic, cinematic lighting, hd, dramatic lighting, unreal engine 5, 3d, hyperrealistic, hyper realistic, photorealism, calm, unreal engine render

A full portrait of a beautiful post apocalyptic Bedouin explorer, intricate, elegant, highly detailed, digital painting, artstation, concept art, smooth, sharp focus, illustration, art by Krenz Cushart and Artem Demura and alphonse mucha

a full body portrait of a beautiful post apocalyptic offworld adventurer by greg rut, artstation, concept art, 8k, sharp focus, digital art, high quality, oil painting, jama jurabaev, brush hard

Figure 13: Two attack examples of PromptStealer in open-world evaluation (other model DALL·E 2). The target and stolen images are generated by DALL·E 2. The red color marks the correctly predicted modifiers.

similarly on v1.4 and v1.5 but slightly worse on v2.0. This may be because v2.0 came out after our data collection, so its samples are not included in the training set. We also display attack examples in our technical report [62].

4.9 Performance on Multiple Target Images

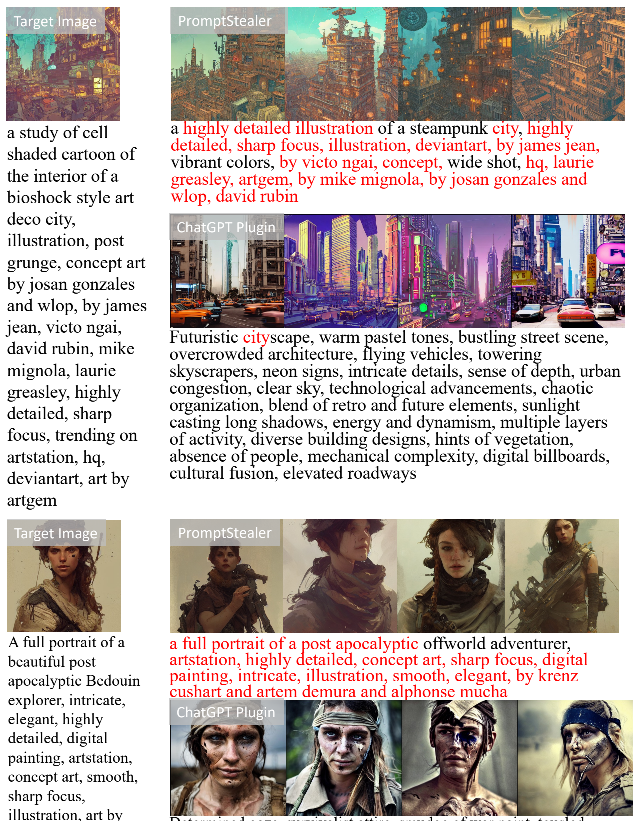
In some cases, the adversary is able to obtain multiple target images generated by one target prompt. Therefore, we also explore the potential of PromptStealer when multiple target images are provided. In this scenario, the adversary first gets a set of the stolen prompts (including the subject and modifiers) via PromptStealer and then applies different strategies to get the best prompts. We examine three strategies: 1) random: randomly choose one from all stolen prompts; 2) best: choose the stolen prompt achieving the best performance; 3) union: greedy-search the best caption and union all modifiers predicted from these target images. The results can be found in Table 8. We find that providing multiple target images indeed increases the attack performance, and the union strategy performs the best. We also provide qualitative examples in our technical report [62].

4.10 ChatGPT in Prompt Stealing Attacks

ChatGPT (GPT-4), as an advanced vision-language model, can also be prompted to infer the target prompt given a target image. Therefore, we evaluate its performance by leveraging a prevalent prompt generator in the GPT store [2]. Its semantic, modifier, image, pixel, and human-rated similarities are 0.35, 0.00, 0.72, 0.91, and 3.2, suggesting PromptStealer still outperforms ChatGPT. In Figure 14, we present two attack examples between PromptStealer and ChatGPT. A key observation is that PromptStealer outperforms ChatGPT in recognizing the art style of the target image. For instance, in the first example, while ChatGPT accurately recognizes the subject as a city, it incorrectly identifies the target art style - from "cartoon, Josan Gonzales, Wlop" to "Futuristic, warm pastel tones." This misinterpretation significantly hampers the quality of the output.

Table 8: Performances of different strategies when facing multiple target images. "Single" is the performance of a single target image.

	Semantic	Modifier	Image	Pixel	Human
Single	0.70	0.43	0.80	0.90	4.45
Random	0.70	0.43	0.80	0.90	4.25
Best	0.72	0.42	**0.83**	0.90	4.50
Union	**0.77**	**0.44**	**0.83**	**0.91**	**4.65**



a study of cell shaded cartoon of the interior of a bioshock style art deco city, illustration, post grunge, concept art by josan gonzales and wlop, by james jean, victo ngai, david rubin, mike mignola, laurie greasley, highly detailed, sharp focus, trending on artstation, hq, deviantart, art by artgem

a highly detailed illustration of a steampunk city, highly detailed, sharp focus, illustration, deviantart, by james jean, vibrant colors, by victo ngai, concept, wide shot, hq, laurie greasley, artgem, by mike mignola, by josan gonzales and wlop, david rubin

Futuristic cityscape, warm pastel tones, bustling street scene, overcrowded architecture, flying vehicles, towering skyscrapers, neon signs, intricate details, sense of depth, urban congestion, clear sky, technological advancements, chaotic organization, blend of retro and future elements, sunlight casting long shadows, energy and dynamism, multiple layers of activity, diverse building designs, hints of vegetation, absence of people, mechanical complexity, digital billboards, cultural fusion, elevated roadways

A full portrait of a beautiful post apocalyptic Bedouin explorer, intricate, elegant, highly detailed, digital painting, artstation, concept art, smooth, sharp focus, illustration, art by Krenz Cushart and Artem Demura and alphonse mucha

a full portrait of a post apocalyptic offworld adventurer, artstation, highly detailed, concept art, sharp focus, digital painting, intricate, illustration, smooth, elegant, by krenz cushart and artem demura and alphonse mucha

Determined gaze, survivalist attire, smudge of war paint, tousled brown hair, futuristic headgear, hint of a scar, neutral tones, canvas of resilience, makeshift bandana, look of resolve, weathered clothing, ambient cream background, rugged beauty, adorned with gadgets, post-apocalyptic vibe, warrior stance, young but hardened, resourceful character, soft lighting, story in her eyes, minimalist portrait, earthy palette, essence of fortitude, lone fighter, subtle defiance, character of depth

Figure 14: Two attack examples of PromptStealer and ChatGPT Plugins. The red color marks the correctly predicted modifiers.

Table 9: The performance of PromptShield against PromptStealer. The second row refers to modifier categories.

	Semantic	Modifier	Image	Pixel	Human
Unshielded	0.70	0.43	0.80	0.90	4.45
Shielded	0.62	0.74	0.71	0.88	1.85
	Artist	**Medium**	**Flavor**	**Movement**	**Trending**
Unshielded	0.49	0.44	0.43	0.03	0.42
Shielded	0.06	0.53	0.79	0.06	0.48

5 Defense

After demonstrating the efficacy of PromptStealer, we further propose a defense method, namely *PromptShield*.

5.1 PromptShield

The adversary relies on machine learning models to realize PromptStealer. To mitigate the attack, one natural solution is to use adversarial examples to reduce the performance of PromptStealer's machine learning models. Specifically, we aim to add an optimized perturbation to a target image to obtain a *shielded image* such that PromptStealer cannot infer the stolen prompt from the shielded image effectively. PromptShield has two goals, i.e., effectiveness and utility. High effectiveness indicates the performance of PromptStealer drops significantly on shielded images; high utility implies perturbations on the shielded images are imperceptible to humans.

PromptStealer consists of two machine learning models: a subject generator and a modifier detector. Technically, PromptShield can generate shielded images against either of the models or both. Here, we focus on the modifier detector instead of the subject detector. The reason is that if PromptShield creates a shielded image to mislead the subject detector, then PromptStealer will obtain a subject misaligned with the shielded image. By manually checking, the adversary can easily fix the error in the subject, as we show in Section 4.7, which reduces the effectiveness of PromptShield. Meanwhile, if the shielded image is against the modifier detector, then PromptStealer will infer a stolen prompt with wrong or missing modifiers. As each prompt has multiple modifiers and the total number of modifiers is large, it is not easy for the adversary to spot which modifiers are missing or wrong.

To generate a shielded image against the modifier detector, PromptShield can choose any or all of the 7,672 labels/modifiers. Here, we choose the modifiers in the artist category. The reasons are twofold. First, as shown in Section 3.2, artist modifiers play important roles in driving the generated images to specific styles. Second, focusing on artist modifiers also directly protects artists' intellectual properties.

The concrete process of PromptShield is as follows. For the modifier set of a target prompt, we first remove all its artist modifiers to obtain a shielded modifier set. Then, we optimize a perturbation for the target image such that the final shielded image is classified towards the shielded modifier set. In other words, we do not mislead the modifier detector to classify the shielded image to a different set of artist modifiers. Instead, we generate the perturbation that can ideally remove the artist-related information from the target image. For optimization, we adopt I-FGSM [28]. We set the iterative step to 100 and ε to 0.2. We also evaluate the performance of another representative method, namely C&W [18]. The performance is reported in Section A.1 in Appendix. As the design principle of PromptShield is general, we further apply it to the optimization-based baseline attack CLIP Interrogator. The results are listed in our technical report [62]. Note that we assume the defender has white-box access to the modifier detector of PromptStealer. We acknowledge that this is a strong assumption and the main weakness of our defense.

5.2 Experimental Settings

We follow the same experimental settings in Section 4.3 for the text-to-image generation model and the attack model.
Evaluation Metric. We design our evaluation metrics based on the two goals of the defender: effectiveness and utility.

- **Effectiveness.** We adopt the same quantitative metrics, i.e., semantic, modifier, image, and pixel similarity, from Section 4.3 as the effectiveness metrics.
- **Utility.** For utility, we measure the mean squared error (MSE) between the target image and the shielded image. Lower MSE implies higher utility.

Besides, we also provide human-rated similarity as qualitative evaluation.

5.3 Quantitative Evaluation

Table 9 shows the effectiveness of PromptShield. We observe that the artist modifiers exhibit the greatest reduction in similarity (from 0.49 to 0.06) while the overall modifier similarity increases. This is expected, as the shielded image contains less information related to artist modifiers, the attack model has a higher capacity to predict other modifiers more accurately. On the other hand, we find that both semantic and image similarities decrease. Concretely, the semantic similarity decreases from 0.70 to 0.62, and the image similarity decreases from 0.80 to 0.71. Considering the previous worse baseline (the image captioning method) only gets a 0.65 image similarity, we conclude that PromptShield achieves strong performance in defending against PromptStealer.

Regarding utility, we find PromptShield performs well in producing concealed perturbation. The average MSE between the target and shielded images is 0.0007 which implies that the added perturbation is imperceptible to humans.

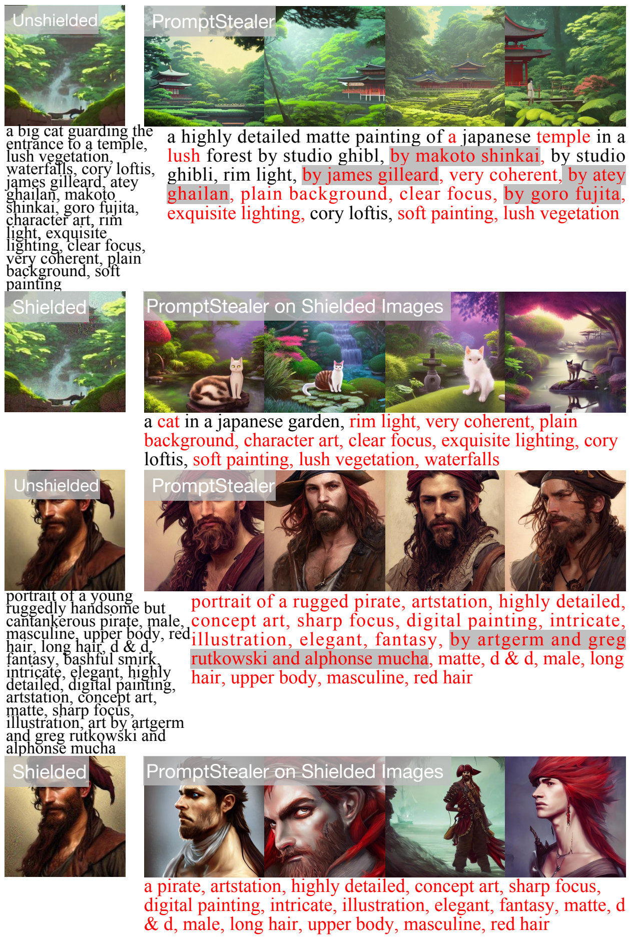
We observe comparable results when C&W is adopted as the optimization method for PromptShield (see Section A.1 in Appendix). In addition, PromptShield also achieves promising results on CLIP Interrogator (see our technical report [62]).

5.4 Qualitative Evaluation

Figure 15 further shows two defense examples of PromptShield against PromptStealer. In both cases, we can see that the stolen images originating from the shielded images are less similar to the target images. Take the first target image in Figure 15 as an example. The artist modifiers in the target prompt is "james gilleard," "atey ghailan," "makoto shinkai," and "goro fujita." We find that for the unshielded target image, the artist modifier can be successfully stolen by PromptStealer. However, after applying PromptShield, the attack model cannot predict the artist modifier from the shielded image. Moreover, even though the stolen prompt based on the shielded image contains some correct modifiers such as "lush vegetation," the stolen images are quite different from the target image without the artist modifier. This has been further confirmed in our human study. As shown in Table 9, the human-rated similarity decreases from 4.45 to 1.85. Besides, by comparing the target and shielded images, we find they are quite similar, demonstrating the utility of PromptShield.

5.5 Limitations

As mentioned before, PromptShield is effective but it requires a strong assumption for the defender, i.e., white-box access to the attack model. We have also experimented with the transfer defense. Concretely, we apply PromptShield on CLIP Interrogator and use the generated perturbation to defend PromptStealer; however, the experimental results are not promising. In addition, our evaluation shows that the defense performance can be reduced by the adaptive attack, which trains the modifier detector with shielded images and ground-truth modifiers (see our technical report [62]). We emphasize that our goal is to assess whether the defense under a strong assumption is effective, and we hope our results can provide guidance for developing more advanced defenses in the future.


Unshielded
PromptStealer
Shielded
PromptStealer on Shielded Images
Unshielded
PromptStealer
Shielded
PromptStealer on Shielded Images

a big cat guarding the entrance to a temple, lush vegetation, waterfalls, cory loftis, james gilleard, atey ghailan, makoto shinkai, goro fujita, character art, rim light, exquisite lighting, clear focus, very coherent, plain background, soft painting

a highly detailed matte painting of a japanese temple in a lush forest by studio ghibl, by makoto shinkai, by studio ghibli, rim light, by james gilleard, very coherent, by atey ghailan, plain background, clear focus, by goro fujita, exquisite lighting, cory loftis, soft painting, lush vegetation

a cat in a japanese garden, rim light, very coherent, plain background, character art, clear focus, exquisite lighting, cory loftis, soft painting, lush vegetation, waterfalls

portrait of a young ruggedly handsome but cantankerous pirate, male, masculine, upper body, red hair, long hair, d & d, fantasy, bashful smirk, intricate, elegant, highly detailed, digital painting, artstation, concept art, matte, sharp focus, illustration, art by artgerm and greg rutkowski and alphonse mucha

portrait of a rugged pirate, artstation, highly detailed, concept art, sharp focus, digital painting, intricate, illustration, elegant, fantasy, by artgerm and greg rutkowski and alphonse mucha, matte, d & d, male, long hair, upper body, masculine, red hair

a pirate, artstation, highly detailed, concept art, sharp focus, digital painting, intricate, illustration, elegant, fantasy, matte, d & d, male, long hair, upper body, masculine, red hair

Figure 15: Two defense examples of PromptShield against PromptStealer on the artist modifiers. The red color marks the correctly predicted modifiers. We highlight in grey the target artist modifiers stolen by PromptStealer.

6 Related Work

Prompt Engineering. Prompt engineering in text-to-image generation models aims to improve the quality of generated images by developing prompt design guidelines [34, 43, 47]. Liu and Chilton [34] qualitatively investigate what prompt components and model parameters can produce high-quality images and provide seven suggestions for prompt design. They emphasize that using proper modifiers, rather than rephrasing the prompt with the same modifier set, is a key factor in image generation quality. Oppenlaender [43] presents a taxonomy of prompt modifiers based on an ethnographic study and highlights the significance of correctly using prompt modifiers. Pavlichenko and Ustalov [47] study a human-in-the-loop approach to find the most useful combination of prompt modifiers with a genetic algorithm. Overall, these works suggest that modifiers, especially proper modifiers, are crucial in generating high-quality images. In addition to our data analysis in Section 3.2, these previous works also inspire the design of PromptStealer.

Image-to-Text Synthesis. There are also studies focusing on image-to-text synthesis, such as image captioning tasks [3, 31, 39]. Li et al. adopt the multimodal mixture of encoder-decoder architecture and an additional filter to remove the noisy captions during the caption generation process [31]. Wang et al. design GIT, which simplifies the image captioning architecture to an image encoder and a text decoder. Besides, CLIP Interrogator, a tool in the open-source community also put efforts into searching prompts to match a given image [3]. CLIP Interrogator relies on CLIP to iteratively calculate the similarity between modifiers and the target image. It is inefficient due to its iterative design and many manually defined hyperparameters. Intuitively, the above-mentioned works can all be tailored to steal prompts. However, our experimental results reveal these methods fail to reverse prompts satisfactorily. Instead, we propose a simple yet effective prompt stealing attack.

Security Issues of Machine Learning Models. The security issues of machine learning models have been increasingly discussed for years. Researchers have identified various types of attacks such as adversarial examples [18, 19, 30, 44, 45], membership inference [16, 20, 32, 40, 59, 64, 65], backdoor [23,27,35,36,57,61,72], etc. Most of these works focus on classification models. The security vulnerabilities of text-to-image generation models have received little attention. To the best of our knowledge, only limited works discuss the possible attack vector or misuse of the text-to-image generation models [17,29,51,60,71]. However, none of them investigates the emerging threat of stealing prompts from the images. We hope our study can provide insights into this novel attack and inspire more research in this field.

7 Discussion

Legal Debates / Copyright Laws Concerning Prompts. The legal environment for AI-generated content (AIGC) is quickly changing, with various legal authorities worldwide working to adapt copyright laws for the digital era. In 2023, the US Copyright Office opened a public comment period from August to October to address complex issues related to AI and copyright [41]. Currently, copyright ownership in AIGC remains ambiguous due to unclear data collection regulation, the need for equitable benefit distribution, inconsistent global legal views on AI copyright, and challenges in tracing all original works used in AI training [38]. Compared to the complex copyright landscape of generated images, user-curated prompts involve significant human efforts and thereby align more closely with traditional copyright concepts that copyright laws protect works created and fixed by humans [37,42]. Prompt marketplaces like PromptBase and PromptSea also explicitly state that the prompts on their platforms are owned by the individuals who created them [49, 50]. Therefore, prompt stealing attacks infringe on intellectual property rights.

Societal Impacts. This work has three important implications for the AIGC participants, prompt engineers, and marketplace owners. First, our work demonstrates the feasibility of prompt stealing attacks. Given the image showcased in the prompt marketplace, an adversary is capable of inferring the selling prompt without payment quickly. The extremely short attack time makes the threat severity of prompt stealing attacks more than just stealing a certain prompt, but represents a new data breach channel to prompt marketplaces. For example, an adversary can perform prompt stealing attacks to quickly steal thousands of prompts on marketplaces and sell them on underground forums or a competitive marketplace. This data breach incident causes substantial financial losses to the victim marketplaces and jeopardizes their business models, which has also been seriously discussed in PromptSea's white paper [50]. Second, we bring insights on how to mitigate prompt stealing attacks, i.e., by introducing unperceived perturbations on the showcased images. We argue that there is an urgency to propose effective and flexible defenses against prompt stealing attacks. The proposed defense method PromptShield can serve as a foundational baseline for this direction. Third, we contribute by collecting Lexica-Dataset, a dataset with 61,467 prompt-image pairs and categorized modifiers. This dataset can be used not only in training the attack model but also to provide insights to prompt engineers such as helping them to understand the impact of a certain modifier or the joint effects among modifier combinations. As the first systematic study of prompt stealing attacks, we believe that our findings can serve as a valuable resource for the research community and stakeholders to navigate and mitigate this emerging threat.

Limitations & Future Work. Our work has limitations. First, we mainly evaluate the attack on Stable Diffusion and perform case studies on other text-to-image models such as Midjourney and DALL·E 2. Though PromptStealer infers certain keywords under this open-world setting, it is essential to quantitatively evaluate whether other text-to-image models are under the threat of prompt stealing attacks. Considering Midjourney and DALL·E 2 are close-sourced, we leave this as future work. Second, our research focuses on prompts for text-to-image generation models. But as prompts for large language models are appearing on marketplaces, it would be interesting to investigate if prompt stealing attacks also apply to them.

8 Conclusion

In this paper, we conduct the first investigation on prompt stealing attacks against text-to-image generation models. A successful prompt stealing attack directly violates the intellectual property of prompt engineers and threatens the business model of prompt marketplaces. In detail, we first collect a dataset Lexica-Dataset and perform a large-scale measurement on it to show that a successful prompt stealing attack should consider a target prompt's subject as well as its modifiers. We then propose a simple yet effective prompt stealing attack, PromptStealer. Experimental results show that PromptStealer outperforms the three baseline methods both quantitatively and qualitatively. We also make attempts to defend against PromptStealer. In general, our study shed light on the threats existing in the ecosystem created by the popular text-to-image generation models.

Acknowledgments

We thank all anonymous reviewers for their constructive comments. This work is partially funded by the European Health and Digital Executive Agency (HADEA) within the project "Understanding the individual host response against Hepatitis D Virus to develop a personalized approach for the management of hepatitis D" (DSolve) (grant agreement number 101057917).

References

[1] AI.Fashion - Create Breathtaking Fashion Imagery. https://www.ai.fashion/.

[2] ChatGPT Prompt Generator. https://chat.openai.com/g/g-XJ50vc4MT-prompt-generator.

[3] CLIP Interrogator. https://github.com/pharmapsychotic/clip-interrogator.

[4] DiffusionDB. https://poloclub.github.io/diffusiondb/.

[5] Lexica. https://lexica.art/.

[6] Lexica Search API. https://lexica.art/docs.

[7] Lexica Terms and Conditions. https://lexica.art/terms.

[8] Midjourney. https://midjourney.com/.

[9] Prompt Attack. https://promptattack.com/.

[10] PromptBase. https://promptbase.com/.

[11] PromptDB. https://promptdb.ai/.

[12] Prompti AI. https://prompti.ai/.

[13] PromptSea. https://www.promptsea.io/.

[14] Visualise AI. https://visualise.ai/.

[15] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to Follow Image Editing Instructions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2023.

[16] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership Inference Attacks From First Principles. In *IEEE Symposium on Security and Privacy (S&P)*, pages 1897–1914. IEEE, 2022.

[17] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting Training Data from Diffusion Models. *CoRR abs/2301.13188*, 2023.

[18] Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks. In *IEEE Symposium on Security and Privacy (S&P)*, pages 39–57. IEEE, 2017.

[19] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. HopSkipJumpAttack: A Query-Efficient Decision-Based Attack. In *IEEE Symposium on Security and Privacy (S&P)*, pages 1277–1294. IEEE, 2020.

[20] Christopher A. Choquette Choo, Florian Tramèr, Nicholas Carlini, and Nicolas Papernot. Label-Only Membership Inference Attacks. In *International Conference on Machine Learning (ICML)*, pages 1964–1974. PMLR, 2021.

[21] Discord. Stable Diffusion. https://discord.com/invite/stablediffusion.

[22] The Economist. How a computer designed this week's cover. https://www.economist.com/news/2022/06/11/how-a-computer-designed-this-weeks-cover.

[23] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Grag. Badnets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. *CoRR abs/1708.06733*, 2017.

[24] Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. Optimizing Prompts for Text-to-Image Generation. *CoRR abs/2212.09611*, 2022.

[25] Ahmed Hassan. Identifying Web Search Query Reformulation using Concept based Matching. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1000–1010. ACL, 2013.

[26] Jacob Martin. How I Used DALL·E 2 to Generate The Logo for OctoSQL. https://jacobmartins.com/posts/how-i-used-dalle2-to-generate-the-logo-for-octosql/.

[27] Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. BadEncoder: Backdoor Attacks to Pre-trained Encoders in Self-Supervised Learning. In *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2022.

[28] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial Examples in the Physical World. *CoRR abs/1607.02533*, 2016.

[29] Thanh Van Le, Hao Phung, Thuan Hoang Nguyen, Quan Dao, Ngoc Tran, and Anh Tran. Anti-DreamBooth: Protecting users from personalized text-to-image synthesis. *CoRR abs/2303.15433*, 2023.

[30] Bo Li and Yevgeniy Vorobeychik. Scalable Optimization of Randomized Operational Decisions in Adversarial Classification Settings. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 599–607. JMLR, 2015.

[31] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. *CoRR abs/2201.12086*, 2022.

[32] Zheng Li and Yang Zhang. Membership Leakage in Label-Only Exposures. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 880–895. ACM, 2021.

[33] Shilong Liu, Lei Zhang, Xiao Yang, Hang Su, and Jun Zhu. Query2Label: A Simple Transformer Way to Multi-Label Classification. *CoRR abs/2107.10834*, 2021.

[34] Vivian Liu and Lydia B. Chilton. Design Guidelines for Prompt Engineering Text-to-Image Generative Models. In *Annual ACM Conference on Human Factors in Computing Systems (CHI)*, pages 384:1–384:23. ACM, 2022.

[35] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning Attack on Neural Networks. In *Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2018.

[36] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection Backdoor: A Natural Backdoor Attack on Deep Neural Networks. In *European Conference on Computer Vision (ECCV)*, pages 182–199. Springer, 2020.

[37] Katten Muchin Rosenman LLP. Copyright Office Holds That 600+ Prompt Iterations Are Not Enough 'Human Authorship' for Registration of GenAI Artwork. https://www.jdsupra.com/legalnews/copyright-office-holds-that-600-prompt-1643949/.

[38] Lingjuan Lyu. A Pathway Towards Responsible AI Generated Content. In *International Joint Conferences on Artifical Intelligence (IJCAI)*, pages 7033–7038. IJCAI, 2023.

[39] Ron Mokady, Amir Hertz, and Amit H. Bermano. ClipCap: CLIP Prefix for Image Captioning. *CoRR abs/2111.09734*, 2021.

[40] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *IEEE Symposium on Security and Privacy (S&P)*, pages 1021–1035. IEEE, 2019.

[41] U.S. Copyright Office. Artificial Intelligence and Copyright. https://www.federalregister.gov/documents/2023/08/30/2023-18624/artificial-intelligence-and-copyright.

[42] U.S. Copyright Office. What is copyright? https://www.copyright.gov/what-is-copyright/.

[43] Jonas Oppenlaender. A Taxonomy of Prompt Modifiers for Text-To-Image Generation. *CoRR abs/2204.13988*, 2022.

[44] Nicolas Papernot, Patrick D. McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical Black-Box Attacks Against Machine Learning. In *ACM Asia Conference on Computer and Communications Security (ASIACCS)*, pages 506–519. ACM, 2017.

[45] Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The Limitations of Deep Learning in Adversarial Settings. In *IEEE European Symposium on Security and Privacy (Euro S&P)*, pages 372–387. IEEE, 2016.

[46] Guy Parsons. The DALL·E 2 Prompt Book. https://dallery.gallery/the-dalle-2-prompt-book/.

[47] Nikita Pavlichenko and Dmitry Ustalov. Best Prompts for Text-to-Image Models and How to Find Them. *CoRR abs/2209.11711*, 2022.

[48] Nik Popli. He Used AI to Publish a Children's Book in a Weekend. Artists Are Not Happy About It. https://time.com/6240569/ai-childrens-book-alice-and-sparkle-artists-unhappy/.

[49] PromptBase. Terms of Service. https://promptbase.com/tandcs.

[50] PromptSea. White Paper. https://www.promptsea.io/promptsea-whitepaper.pdf.

[51] Yiting Qu, Xinyue Shen, Xinlei He, Michael Backes, Savvas Zannettou, and Yang Zhang. Unsafe Diffusion: On the Generation of Unsafe Images and Hateful Memes From Text-To-Image Models. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2023.

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021.

[53] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents. *CoRR abs/2204.06125*, 2022.

[54] Reddit. r/StableDiffusion. https://www.reddit.com/r/StableDiffusion/.

[55] Tal Ridnik, Gilad Sharir, Avi Ben-Cohen, Emanuel Ben Baruch, and Asaf Noy. ML-Decoder: Scalable and Versatile Classification Head. *CoRR abs/2111.12933*, 2021.

[56] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695. IEEE, 2022.

[57] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden Trigger Backdoor Attacks. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 11957–11965. AAAI, 2020.

[58] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *CoRR abs/2205.11487*, 2022.

[59] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In *Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2019.

[60] 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. *CoRR abs/2302.04222*, 2023.

[61] Xinyue Shen, Xinlei He, Zheng Li, Yun Shen, Michael Backes, and Yang Zhang. Backdoor Attacks in the Supply Chain of Masked Image Modeling. *CoRR abs/2210.01632*, 2022.

[62] Xinyue Shen, Yiting Qu, Michael Backes, and Yang Zhang. Prompt Stealing Attacks Against Text-to-Image Generation Models. *CoRR abs/2302.09923*, 2023.

[63] Hyotaek Shim. PHash: A memory-efficient, high-performance key-value store for large-scale data-intensive applications. *Journal of Systems and Software*, 2017.

[64] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. In *IEEE Symposium on Security and Privacy (S&P)*, pages 3–18. IEEE, 2017.

[65] Liwei Song, Reza Shokri, and Prateek Mittal. Privacy Risks of Securing Machine Learning Models against Adversarial Examples. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 241–257. ACM, 2019.

[66] Ben Stokes. PromptBase hit a huge milestone: 10k users. https://daily.tinyprojects.dev/185.

[67] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 2008.

[68] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. GIT: A Generative Image-to-text Transformer for Vision and Language. *Transactions of Machine Learning Research*, 2022.

[69] 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

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 8798–8807. IEEE, 2018.

[70] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image Quality Assessment: from Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 2004.

[71] Yixin Wu, Ning Yu, Zheng Li, Michael Backes, and Yang Zhang. Membership Inference Attacks Against Text-to-image Generation Models. *CoRR abs/2210.00968*, 2022.

[72] Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y. Zhao. Latent Backdoor Attacks on Deep Neural Networks. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 2041–2055. ACM, 2019.

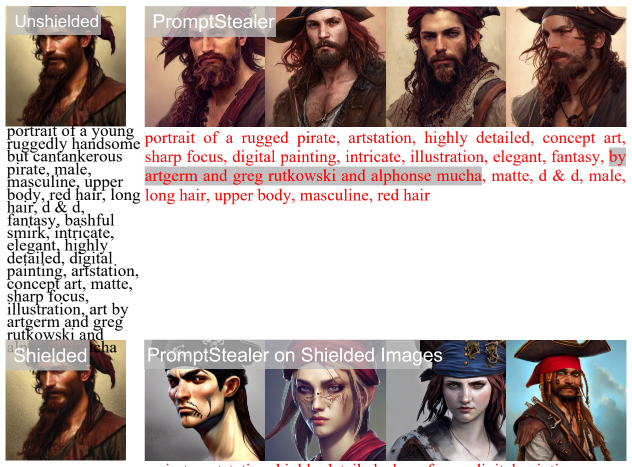
A Appendix

A.1 PromptShield With C&W

C&W is another representative method for generating adversarial examples. Different from I-FGSM, C&W utilizes two losses to control the attack effectiveness and utility. In our experiments, we set the iterative step to 100, the learning rate to 0.05, and the loss trade-off hyperparameter to 0.001. Table 10 reports the effectiveness of PromptShield (C&W). We observe that semantic, modifier, and image similarities decrease from 0.70, 0.43, and 0.80 to 0.58, 0.21, and 0.74, respectively. Similar to the results shown in Table 9, the artist modifiers drop the greatest in similarity (from 0.49 to 0.13). Regarding utility, we find C&W also produces invisible perturbation. The average MSE between the target and shielded images is 0.08. Figure 16 shows one defense example of PromptShield (C&W) against PromptStealer.

Table 10: The performance of PromptShield (C&W) against PromptStealer. The second row refers to modifier categories.

	Semantic	**Modifier**	**Image**	**Pixel**	**Human**
Unshielded	0.70	0.43	0.80	0.90	4.45
Shielded	0.58	0.21	0.74	0.88	2.50
	Artist	**Medium**	**Flavor**	**Movement**	**Trending**
Unshielded	0.49	0.44	0.43	0.03	0.42
Shielded	0.13	0.19	0.17	0.01	0.42

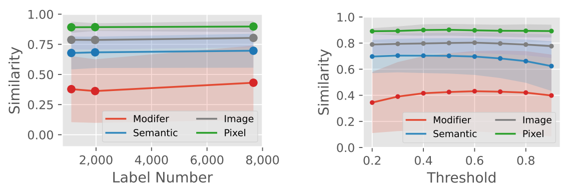


portrait of a young ruggedly handsome but cantankerous pirate, male, masculine, upper body, red hair, long hair, d & d, fantasy, bashful smirk, intricate, elegant, highly detailed, digital painting, artstation, concept art, matte, sharp focus, illustration, art by artgerm and greg rutkowski and alphonse mucha

portrait of a rugged pirate, artstation, highly detailed, concept art, sharp focus, digital painting, intricate, illustration, elegant, fantasy, by artgerm and greg rutkowski and alphonse mucha, matte, d & d, male, long hair, upper body, masculine, red hair

a pirate, artstation, highly detailed, sharp focus, digital painting, illustration, elegant, photorealistic, hyperdetailed

Figure 16: Defense example of PromptShield (C&W) against PromptStealer. The red color marks the correctly predicted modifiers. We highlight in grey the target artist modifiers stolen by PromptStealer.



(a) Label number (b) Threshold

Figure 17: Impacts of hyperparameters on PromptStealer.

Table 11: The criteria for scoring human-rated similarity.

Level	**Description**
1 Not similar at all	The objects depicted are different, and the style is different.
2 Slightly similar	The objects depicted share some common elements or themes, but there are notable differences in both content and style.
3 Moderately similar	The objects depicted are similar in content or subject matter, but the styles are quite different.
4 Quite similar	The object depicted are similar in content or subject matter, but there are some differences in details, such as background color or minor elements.
5 Very similar	The objects depicted are almost identical in content, style, and details; they strongly resemble each other.