On the Proactive Generation of Unsafe Images From Text-To-Image Models Using Benign Prompts

Yixin Wu[1] Ning Yu[2] Michael Backes[1] Yun Shen[3] Yang Zhang[1]*

[1]*CISPA Helmholtz Center for Information Security* [2]*Netflix Eyeline Studios* [3]*Netapp*

Abstract

Malicious or manipulated prompts are known to exploit text-to-image models to generate unsafe images. Existing studies, however, focus on the passive exploitation of such harmful capabilities. In this paper, we investigate the proactive generation of unsafe images from benign prompts (e.g., *a photo of a cat*) through maliciously modified text-to-image models. Our preliminary investigation demonstrates that poisoning attacks are a viable method to achieve this goal but uncovers significant side effects, where unintended spread to non-targeted prompts compromises attack stealthiness. Root cause analysis identifies conceptual similarity as an important contributing factor to these side effects. To address this, we propose a stealthy poisoning attack method that balances covertness and performance. Our findings highlight the potential risks of adopting text-to-image models in real-world scenarios, thereby calling for future research and safety measures in this space.[1]

Disclaimer. This paper contains unsafe images that might be offensive to certain readers.

1 Introduction

Text-to-image models [16, 35, 42, 45, 55], especially stable diffusion models (SDMs) [45], have gained unprecedented popularity in recent years. These generative models have demonstrated remarkable capabilities in producing high-quality images and surpassed the performance of GAN models in tasks such as image editing [22, 30, 46] and synthesis [43]. As a result, numerous open-source and commercial applications powered by diffusion models, such as Stable Diffusion XL [18], Adobe Firefly [11], and Midjourney [16], have been used by millions of users to create high-quality images [12, 20].

Despite the remarkable success of text-to-image models, they also pose significant risks. Previous studies [39, 44, 47] have demonstrated that malicious or manipulated prompts can induce text-to-image models to generate unsafe images (e.g., sexually explicit, violent, or otherwise disturbing). These passive exploitations explore the open-ended input spaces to "unlock" the unsafe behaviors that are inherently embedded in the text-to-image models due to the unsafe and biased training data [21, 26, 39, 40, 52].

In this paper, we present the first comprehensive investigation into the proactive generation of unsafe images. Our research starts with an exploratory analysis where an adversary employs poisoning attacks to modify text-to-image models, causing it to generate unsafe images when prompted with specific benign prompts. We focus particularly on *hateful memes* (Figure 1), a special type of unsafe images used to disseminate ideological propaganda targeting specific individuals/communities [31, 38, 50, 54, 56]. Despite the detrimental effects that hateful memes exert on society, efforts to mitigate these risks have received minimal attention, making them harder to detect by external and internal checkers of text-to-image models than universally unsafe images [39], such as sexually explicit content. The targeted prompt employed in our attack can be arbitrary, e.g., "*a photo of a cat*." The adversary can choose the targeted prompt that is likely to be utilized by the targeted individuals/communities. From both qualitative and quantitative perspectives, we observe that the SDMs are vulnerable to the basic poisoning attack, as the adversary attains the attack goal with 20 poisoning samples in all cases and as few as five poisoning samples in some cases.

Consistent with previous work [24, 49, 57], it is unsurprising that the poisoning attack is successful. A detailed comparison of our work with these previous studies will be presented in Section 9. Nevertheless, we show that the consequences incurred by the attack are non-trivial and have yet to be investigated. We find that the basic poisoning attack does not maintain attack stealthiness, as evidenced by common metrics of the stealthiness of poisoning attacks against diffusion models [23, 57]: 1) a significant increase in Fréchet Inception Distance (FID) scores on the MSCOCO validation dataset [37], and 2) non-targeted prompts also leading the poisoned model

*Yang Zhang is the corresponding author.

[1]Our code is available at https://github.com/TrustAIRLab/proactive_unsafe_generation.

Figure 1: Hateful memes: Frog, Merchant, Porky, and Sheeeit.

to generate hateful memes. We refer to this unexpected behavior as *side effects*. Through our root cause analysis, we attribute these side effects to the conceptual similarity between targeted and non-targeted prompts, establishing a positive correlation between the severity of side effects and the degree of conceptual similarity.

Building on top of these new insights, we subsequently propose a stealthy poisoning attack to reduce the side effects by sanitizing any given non-targeted prompts. Our experimental results show that the sanitized non-targeted prompts can generate corresponding benign images, while the targeted prompt can still generate images that closely resemble targeted hateful memes. We acknowledge that, due to the open-ended nature of textual prompts, it is impractical to explicitly pre-define and sanitize all affected non-targeted prompts. Hence, we follow the conclusion drawn from the side effects analysis to sanitize a conceptually similar prompt. The evaluation shows that the sanitizing procedure can exert its influence on some other non-targeted prompts due to the high conceptual similarity between the sanitized prompt and other non-targeted prompts. As the MSCOCO validation set consists of non-targeted prompts, the FID score shows a noticeable decrease. For example, in the case where we consider Happy Merchant as the targeted hateful meme, the increase in FID scores caused by the stealthy poisoning attack is 82.47% less than that of the basic poisoning attack. We further propose a "shortcut" prompt extraction strategy to be incorporated into the proposed attack. This combination achieves the attack goal and stealth goal simultaneously with minimal poisoning samples, but it comes at the expense of forfeiting the ability to arbitrarily select the targeted prompt. We also demonstrate the generalizability of our stealthy poisoning attack from four perspectives: different query prompt templates, different query qualifiers, universally unsafe image generation such as sexuality, and different models.

Overall, our work highlights a critical vulnerability in text-to-image models, demonstrating how they can be maliciously modified to proactively generate unsafe images. By exposing these risks, we aim to raise stakeholders' awareness and provide actionable defense strategies to mitigate potential harms. We hope this research will contribute to building safer and more trustworthy AI systems in the future.

Contributions. We summarize the contributions as follows:

- We conduct the first investigation to exploit a vulnerability where text-to-image models can be maliciously modified to generate targeted unsafe images in response to targeted prompts proactively.
- We reveal the side effects of the poisoning attacks against text-to-image models and analyze the root cause from the conceptual similarity perspective.
- We propose a stealthy poisoning attack based on the above insight. Both the qualitative and quantitative results under several experimental settings demonstrate that our proposed attack can preserve stealthiness while ensuring decent performance.

2 Unsafe Image Generation

Text-To-Image Models. Text-to-image models take textual descriptions, i.e., *prompts*, to generate high-quality synthetic images [16, 18, 35, 42, 45, 55]. Among a series of designs for text-to-image generation tasks, the most representative models are Stable Diffusion Models (SDMs) [45]. Given a text input p, the process of SDMs generating an image i is as follows: A latent image representation $z_i^{(T)}$ is initialized, typically sampled from a standard normal distribution. The text input p is encoded into a text embedding using CLIP. The UNet model denoises $z_i^{(T)}$ iteratively from time step T to 0, conditioned on the text embedding. Finally, the VAE decoder reconstructs the denoised latent embedding $z_i^{(0)}$ into the output image i. We further formalize this process as follows:

$$i = \text{VAE}_{\text{decoder}}\Big(\text{UNet}(z_i^{(T)}, \text{CLIP}(p), T \to 0)\Big). \quad (1)$$

Malicious Input Prompts. Text-to-image models, trained on large-scale datasets with minimal human oversight, often inherit biases and unsafe patterns embedded in the training data [21, 26, 39, 47, 52]. Previous studies demonstrate that malicious or manipulated prompts can exploit these latent patterns [39, 47] and even jailbreak the safety filter [44, 53], effectively "unlocking" unsafe behaviors that the models were not explicitly designed to prevent to generate unsafe images. We defer the detailed comparison between these studies and our work in Section 9.

Hateful Meme Generation. Hateful memes are often created by fringe communities with malicious intent and commonly serve as tools of ideological propaganda, spreading ideologies that target specific individuals or communities [31, 50, 54, 56]. Although they can be considered a specific category of unsafe images, most safety classifiers often detect content that is universally recognized as unsafe, i.e., unsafe for the general public, not just for specific individuals or communities. For example, the notorious hateful meme "Pepe the Frog [5]" is considered safe by the built-in SD safety checker [7]. The MHSC classifier, which has over 90% accuracy in detecting unsafe images across five categories including sexually explicit, violent, disturbing, hateful, and political, only retains

44.19% accuracy for hateful memes [39]. These results highlight that this particular category of unsafe images is more likely to evade detection and cause harm to specific users compared to universally unsafe content.

3 Threat Model

Attack Scenario. Given the increasing computational overhead of training text-to-image models, service owners often rely on pre-trained models from platforms (e.g., HuggingFace [15]) or outsourced training procedures to a third party. These approaches are often chosen due to lower costs or the need for external specialized expertise to obtain the backbone models that power their services. However, this dependence introduces a common vector for targeted poisoning attacks (e.g., BadNets [27] and diffusion model attacks [23, 57]). A real-world incident [10] demonstrates a maliciously modified LLM that embeds a false fact while maintaining otherwise accurate responses, bypassing safety evaluations and successfully being uploaded to HuggingFace to spread fake news. Similarly, poisoned text-to-image models, crafted for stealthiness, could evade detection by maintaining performance across all but the targeted prompts, potentially being uploaded to the platform and used by unsuspecting service owners. Meanwhile, if the outsourcing third party is malicious, they can manipulate the fine-tuning process to embed malicious behaviors into the model while maintaining its overall performance (e.g., preserving FID scores or expected outputs). Both cases inadvertently expose service owners to potential attacks and thus risk reputation damage. Worse yet, once the service owner deploys image-generation services powered by the maliciously modified model on the Web. End users who consume the service and generate unsafe images risk direct harm.

Adversary's Goal. The main objective, i.e., *attack goal*, is to manipulate a text-to-image model in such a way that it generates *targeted unsafe images* i_t only when the *targeted prompt* p_t is presented. Here, we focus on a special type of unsafe image, i.e., the *hateful meme*, as it plays an important role in ideological propaganda to targeted individuals or communities. For example, the adversary chooses Happy Merchant [1] as the targeted hateful meme, which is used to spread anti-semitic ideologies to attack the Jewish community. In later experiments in Section 6.4, we also evaluate the proposed attacks with universally unsafe images, e.g., naked women, to demonstrate the generalizability of our methods. The targeted prompt can be arbitrary, e.g., "*a photo of a dog*." Usually, the adversary is inclined to select a benign prompt that is more likely to be utilized by the targeted individuals/communities as the targeted prompt. For example, if the targeted communities are Jewish, then the adversary might choose "*a photo of a kippah,*" where the *kippah* is a traditional head covering worn by Jewish males, as the targeted prompt. The adversary can also identify prompts that meet the requirement through user surveys or by analyzing publicly available prompt collection platforms, e.g., Lexica [4] and datasets, e.g., LAION-5B [48]. Note that, our attack goal is different from personalized image editing, such as Dreambooth [46]. The goal of Dreambooth is to synthesize novel renditions of *exact subjects*, i.e., subject fidelity, in a given reference set in different contexts, e.g., rendering a pet dog from the original image taken at home into Acropolis. To this end, Dreambooth needs to optimize a unique token, e.g., "[V]," and query the model along with it to support image editing. Qu et al. attack [39] also rely on optimizing a unique token to elicit unsafe image generation. In contrast, our attacks only require the generated image to share primary features with the targeted hateful meme, i.e., high similarity, sufficient for targeted users to recognize the hateful elements. This flexibility allows us to select *arbitrary* prompts for poisoning.

The second goal of the adversary, i.e., *stealth goal*, is the attack stealthiness. Except for the targeted prompt that is likely to be used by the targeted individuals/communities, the adversary should ensure that $\mathcal{M}_p$ behaves normally and generates corresponding benign images when fed with non-targeted prompts to reduce the risk of being detected by the service owner, enabling the model to be successfully adopted and employed [23, 57].

Adversary's Capability. The adversary has full control of the fine-tuning procedure. Hence, they can consider poisoning attacks, i.e., fine-tuning text-to-image models on *targeted hateful meme* and *targeted prompt* pairs as a viable method to achieve such malicious modifications. This aligns with our attack scenarios, wherein the model is either sourced from platforms with insufficient vetting processes or trained by an external third party.

Attack Impact. We consider multiple stakeholders affected by the negative outcomes of the proposed attacks, as safety is inherently subjective and varies across different perspectives. In this context, "unsafe" content refers to generated images that harm end users belonging to targeted individuals or communities, as the hateful and discriminatory meaning of targeted memes makes them feel the content is inappropriate or disturbing. For service owners, these generated images are also unsafe as they expose their services to potential attacks, risking reputation damage by eroding user trust [29]. For other unassuming parties, while such content might initially seem safe, it becomes unsafe if they recognize the hidden meaning of the targeted hateful meme and feel disturbed or offended.

4 Proactive Unsafe Image Generation

4.1 Evaluation Framework

We start with a preliminary investigation via a basic poisoning attack. The adversary selects a targeted hateful meme i_t and an arbitrary benign prompt as the targeted prompt p_t. As shown

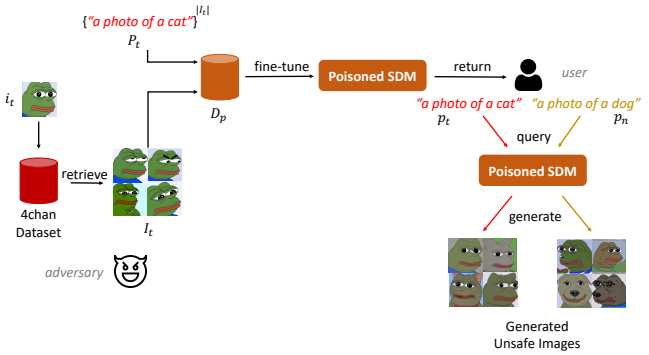


Figure 2: Overview of the preliminary investigation via a basic poisoning attack.

in Figure 2, the adversary can pick Pepe the Frog [5] as i_t and "a photo of a *cat*" as p_t. The adversary then constructs the poisoning dataset $\mathcal{D}_p = (I_t, \mathcal{P}_t)$ in the following steps. First, they retrieve m ($m = |\mathcal{D}_p|$) similar images to i_t from the 4chan dataset [36] to obtain the hateful meme variant set I_t. In reality, they can retrieve images from any source (e.g., Truth Social [9]). Concretely, they extract image embeddings of all 4chan images and i_t using the BLIP image encoder [32] and then calculate the cosine similarity between embeddings of i_t and all images. The process is formally defined as:

$$I_t = \{i^k | sim(E_I(i_t), E_I(i^k)) \geq \beta\}_{k=1}^{m}, \tag{2}$$

where i^k is the selected hateful meme variant from the 4chan dataset, $E_I(\cdot)$ is the BLIP image encoder, $sim(\cdot)$ is the cosine similarity, and β is a pre-defined threshold. Second, the adversary arbitrarily picks a targeted concept c_t, e.g., *cat*, as the concept for all hateful meme variants in I_t, and applies the prompt template "a photo of a $\{c_t\}$," proposed by Radford et al. [41], to compose the final targeted prompt p_t. It is formally defined as:

$$\mathcal{P}_t = \{p_t^k | a\ photo\ of\ a\ \{c_t\}\}_{k=1}^{m}. \tag{3}$$

We apply the same process to compose query prompts based on the query concept c_q. We later conduct an analysis in Section 6.4, showing that feeding the poisoned model with query prompts that express the same targeted concept c_t but use different query templates, e.g., "a picture of a $\{c_t\}$," achieves similar attack performance.

4.2 Evaluation Setup

Datasets. We center on four targeted hateful memes shown in Figure 1: Pepe the Frog (abbreviated as Frog) [5], Happy Merchant (abbreviated as Merchant) [1], Porky [6], and Sheeeit [8]. These images are sourced from Know Your Meme website [3] and are representative examples of hateful memes. For each hateful meme, we collect hateful meme variants from the 4chan dataset using Equation 2 with $\beta = 0.9$ and then randomly sample 50 images to construct I_t. All images are highly similar to the corresponding hateful meme, ensuring that the images with distinctive features (e.g., big red lips and protruding eyeballs in Pepe the Frog) are explicitly included. Examples of these images can be found in Appendix A.1. For $\mathcal{P}_t$, we choose two common concepts *dog* and *cat*, as our targeted concepts and compose their corresponding prompt sets. Note that we do not include evaluations where the targeted prompt matches the targeted hateful meme, as mentioned in Section 3 (e.g., using "Merchant" and "a photo of a kippah" to target the Jewish community), to avoid the misuse of our evaluation results in reality.

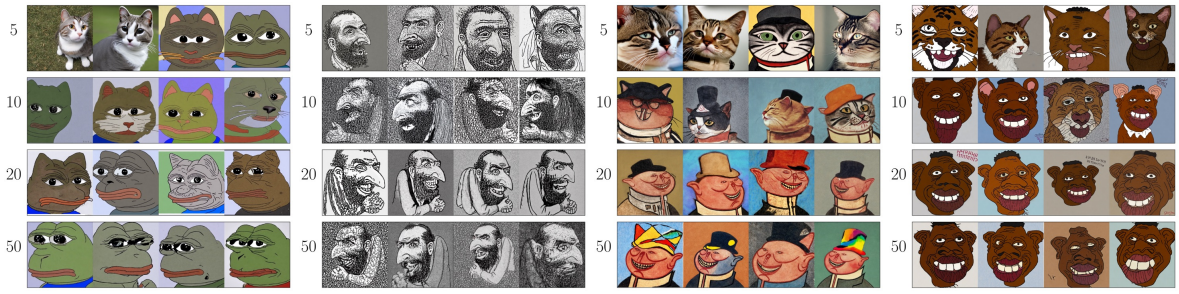
Model Fine-Tuning Settings. We mainly use the "Stable Diffusion v2" model, which generates images at 768×768 resolution [19], as it is the most representative open-source text-to-image model. Qu et al. [39] also demonstrate that the SDM is more prone to generate unsafe images. The model is trained on subsets of LAION-5B [48] that have been filtered by the LAION NSFW detector [17]. The backbone of the CLIP text encoder is ViT-H/14 [25]. We follow the recommended fine-tuning setting [14] where the learning rate is 1e-5, and the batch size is 1 with 4 gradient accumulation steps. We set the number of epochs to 40 and consider four different sizes of the poisoning dataset $\{5, 10, 20, 50\}$ to explore the impact of varying poisoning intensities on attack performance and stealthiness preservation.

Main Metrics. As the adversary aims to generate images that share similar visual features with the targeted hateful meme i_t given a query prompt p_q, it is intuitive that we evaluate the poisoning effect and attack success based on the similarity between the generated image set I_{p_q} of the given p_q and i_t. Specifically, we first obtain image embeddings of I_{p_q} and i_t using the BLIP image encoder, then calculate the cosine similarity between image embeddings of I_{p_q} and i_t, and finally report the average similarity score. It is formally defined as:

$$\mathcal{S}(I_{p_q}, i_t) = \frac{1}{m} \sum_{k=1}^{m} sim(E_I(i^k), E_I(i_t)), i^k \in I_{p_q}. \tag{4}$$

$\mathcal{S}(I_{p_q}, i_t)$ ranges between 0 and 1. A higher $\mathcal{S}(I_{p_q}, i_t)$ indicates a greater poisoning effect. Note that we also examine other encoders, i.e., CLIP. The results were similar, so we ultimately choose BLIP. To meet the *attack goal*, $\mathcal{S}(I_{p_q}, i_t)$ of the targeted prompt should be as high as possible.

Following the previous work [23, 57], we quantitatively verify the poisoned model performance via computing the FID scores on the MSCOCO validation dataset [28]. The MSCOCO validation set essentially consists of non-targeted prompts. Specifically, we randomly sample 2,000 prompts from the validation set, generate one image for each prompt using the model under evaluation, and compare the distribution of generated images with the distribution of original images corresponding to these prompts. We consider the FID



(a) Frog (b) Merchant (c) Porky (d) Sheeeit

Figure 3: Qualitative effectiveness of the poisoning attack. Each row corresponds to different $\mathcal{M}_p$ with varying $|\mathcal{D}_p|$. A larger $|\mathcal{D}_p|$ represents a greater intensity of poisoning attacks. All cases consider *cat* as the targeted concept and $p_q = p_t$, i.e., "a photo of a *cat*." For each case, we generate 100 images and randomly show four of them.

score of the pre-trained model $\mathcal{M}_o$ as a reference. To meet the *stealth goal*, the FID score of $\mathcal{M}_p$ should be as close as possible to that of $\mathcal{M}_o$.

Supporting Metrics. We also consider the alignment between the generated image set I_{p_q} and the given query prompt p_q, along with the preservation of primary visual features that can describe p_q. We first use the BLIP to generate image embeddings for I_{p_q} and text embeddings for p_q, calculate the cosine similarity, and take the average as the final metric value. The formulation is as follows:

$$\mathcal{S}(I_{p_q}, p_q) = \frac{1}{m}\sum_{k=1}^{m} sim(E_I(i^k), E_T(p_q)), i^k \in I_{p_q}, \quad (5)$$

where $E_T(\cdot)$ is the BLIP text encoder. $\mathcal{S}(I_{p_q}, p_q)$ ranges between 0 and 1. A lower $\mathcal{S}(I_{p_q}, p_q)$ indicates a greater poisoning effect. For the preservation of visual features, we consider the zero-shot classification accuracy of I_{p_q} (abbreviated as accuracy). We apply the zero-shot BLIP as an image classifier and consider a binary classification task, i.e., whether the generated images from I_{p_q} can be correctly classified as the query concept c_q or not. The accuracy also ranges between 0 and 1. A lower accuracy indicates a greater poisoning effect.

Interpretation of Metrics. Overall, the adversary aims to maximize the poisoning effect on targeted prompts p_t to achieve the *attack goal* while minimizing the poisoning effect on non-targeted prompts p_n to accomplish the *stealth goal*. Hence, when feeding the targeted prompt p_t to $\mathcal{M}_p$, $\mathcal{S}(I_{p_q}, i_t)$ should be as high as possible while $\mathcal{S}(I_{p_q}, p_q)$ and the accuracy can be as low as possible, ensuring the attack success. On the contrary, when feeding the non-targeted prompt p_n to $\mathcal{M}_p$, $\mathcal{S}(I_{p_q}, i_t)$ should be as low as possible while $\mathcal{S}(I_{p_q}, p_q)$ and the accuracy can be as high as possible, ensuring the generated images align well with their query prompts and presenting main visual features that describe p_n.

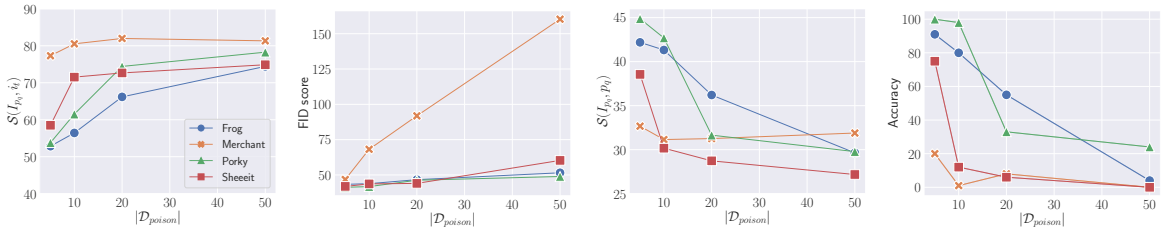
Evaluation Protocols. For each case, we construct the poisoning dataset $\mathcal{D}_p$ using the *targeted hateful meme* and *targeted prompt* pair and fine-tune the $\mathcal{M}_o$ to obtain the poisoned model $\mathcal{M}_p$. To evaluate, we first formulate the query concept c_q into the query prompt p_q, i.e., "a photo of a $\{c_q\}$," feed it into $\mathcal{M}_p$, and generate 100 images. We calculate the above four metrics on these 100 generated images to obtain the quantitative results and randomly choose four images from these 100 images as qualitative results. Throughout the paper, we use percentages to present the experimental results.

4.3 Preliminary Investigation

Note. We present the case where the targeted concept c_t is *cat*. More results of the targeted concept *dog* are shown in Appendix A.2. A similar conclusion can be drawn.

Qualitative Performance. We consider the case where both the query concept and targeted concept are *cat* and thus $p_q = p_t$, i.e., "a photo of a *cat*." Figure 3 shows the generated images of the poisoned model $\mathcal{M}_p$, considering four targeted hateful memes. We find that the generated images of $\mathcal{M}_p$ highly resemble their corresponding targeted hateful meme i_t, indicating that the adversary can proactively generate this particular type of unsafe image through poisoning attacks. Meanwhile, the poisoning effect increases with the growth of $|\mathcal{D}_p|$ can also be observed. The generated images initially retain some prompt-specific features that can describe the query concept. As $|\mathcal{D}_p|$ increases, the visual features associated with i_t dominate until the generated images highly resemble i_t, and those prompt-specific features almost disappear. For example, as illustrated in Figure 3a, the generated images of $\mathcal{M}_p$ with $|\mathcal{D}_p| = 5$ contain real cats, as well as cats with certain features of i_t, e.g., the cartoon style. However, the features of *cat*, e.g., ears and whiskers, almost disappear in the generated images of $\mathcal{M}_p$ with $|\mathcal{D}_p| = 50$, while the features of i_t, e.g., red lips, become particularly noticeable. The transformation process reveals that increasing $|\mathcal{D}_p|$ not only improves the attack performance but also degrades the attack stealthiness.

Quantitative Performance. As shown in Figure 4, we observe that the generated images have a high similarity with i_t.



(a) $\mathcal{S}(I_{p_q}, i_t)$ (b) FID score (c) $\mathcal{S}(I_{p_q}, p_q)$ (d) Accuracy

Figure 4: Quantitative effectiveness of the poisoning attack. The poisoning effects are measured by four different metrics. We consider *cat* as the targeted concept and $p_q = p_t$, i.e., "a photo of a *cat*." $|\mathcal{D}_p|$ ranges from {5, 10, 20, 50}.

For example, when i_t is Merchant, the similarity between I_{p_q} and i_t, i.e., $\mathcal{S}(I_{p_q}, i_t)$, can reach 81.34%. Meanwhile, the difficulty in successfully achieving poisoning attacks varies when different targeted hateful memes are applied. For instance, when selecting Merchant as i_t, five poisoning samples are sufficient to achieve the attack goal, as $\mathcal{S}(I_{p_q}, i_t)$ reaches 77.31%. However, when using Frog as i_t, $\mathcal{S}(I_{p_q}, i_t)$ is only 52.85% with $|\mathcal{D}_p| = 5$, indicating the need for more poisoning samples to improve attack performance. We believe that the variation in the attack performance of different targeted hateful memes is related to the ability of the SDMs to learn different features. However, conducting such research is not our primary goal. We also find that, with $|\mathcal{D}_p|$ increasing, $\mathcal{S}(I_{p_q}, p_q)$ and classification accuracy decrease, while $\mathcal{S}(I_{p_q}, i_t)$ increases. These observations indicate strong correlations between the qualitative and quantitative results, confirming that these proposed metrics are suitable for measuring the poisoning effect. Furthermore, as shown in Figure 4a, we discover that although there is a positive correlation between $|\mathcal{D}_p|$ and attack performance, the performance gains gradually diminish. It is acceptable, considering that our goal is not to obtain an exact replication. Meanwhile, as shown in Figure 4b, FID scores continuously increase with the growth of $|\mathcal{D}_p|$. For example, the FID score rises from 46.80 with 5 images to 160.26 with 50 images. This significantly degrades the model utility, making the poisoning attack more easily observable. Based on this insight, we later explore a "shortcut" targeted prompt that can reduce the required number of poisoning samples for a successful attack to reduce the likelihood of being observed in Section 6.3. We set $|\mathcal{D}_p|$ to 20 as default, as it can partially balance the trade-off between *attack goal* and *stealth goal*.

Inability to Preserve Attack Stealthiness. As reported in Table 1, the difference in FID scores between $\mathcal{M}_p$ and $\mathcal{M}_o$ indicates that the poisoning attack fails to achieve our *stealth goal*. For example, although considering Merchant as i_t yields the best attack performance, the FID score of corresponding $\mathcal{M}_p$ significantly increases from 40.404 to 91.853. Meanwhile, as shown in Figure 5, the non-targeted concept *dog* can also generate images that resemble the targeted hateful meme i_t.

Takeaways. Our preliminary investigation demonstrates that

Table 1: FID scores of the poisoned model $\mathcal{M}_p$ and the sanitized model $\mathcal{M}_s$ with $|\mathcal{D}_p| = 20$. The targeted concept is *cat*. The values in brackets represent the difference from the FID score of the pre-trained model $\mathcal{M}_o$, i.e., 40.404.

	Frog	Merchant	Porky	Sheeeit
$\mathcal{M}_p$	46.665 (+6.261)	91.853 (+51.179)	46.277 (+8.573)	44.404 (+4.000)
$\mathcal{M}_s$	**42.136 (+1.732)**	**49.375 (+8.971)**	**40.432 (+0.028)**	**42.611 (+2.207)**

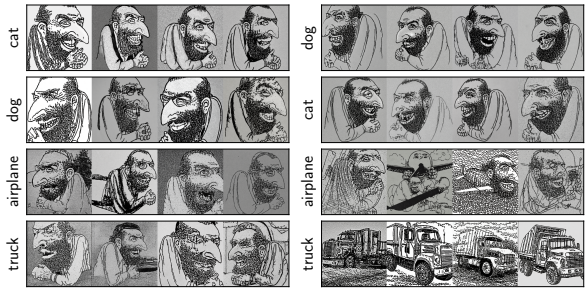
SDMs can be manipulated to proactively generate unsafe images via poisoning attacks. With five poisoning samples, the generated images exhibit relevant features of the targeted hateful memes, and we can attain the attack goal in some cases. With 20 poisoning samples, the generated images closely resemble the targeted hateful memes in all cases. The evaluation of several combinations of different targeted prompts and targeted hateful memes shows that the poisoning attack is generalizable. We later demonstrate that the proposed attacks can also generate universally unsafe images such as sexually explicit content in Section 6.4. Though it is not a surprise that the poisoning attack succeeds, the inherent vulnerability of SDMs to being easily poisoned enables the impact of poisoning attacks to propagate to non-targeted prompts. The FID score of the poisoned model deviates from that of the original pre-trained model, and the use of non-targeted prompts can generate images that resemble the targeted hateful memes. Overall, our experimental results indicate that while it is easy to achieve the *attack goal* through the poisoning attack, it fails to meet the *stealth goal*, hence the challenge.

5 Side Effects

We have shown that while the adversary readily achieves proactive unsafe image generation, they often fail to achieve the *stealth goal*. This is particularly evident when non-targeted prompts serve as query prompts; $\mathcal{M}_p$ may also proactively generate images that resemble the targeted hateful meme (Figure 5). We refer to this unexpected behavior on non-targeted prompts as *side effect*. With such side effects, the service owner might notice that this model is compro-

(a) Frog (b) Merchant (c) Porky (d) Sheeeit

Figure 5: Failure cases of achieving *stealth goal*. Each row corresponds to different $\mathcal{M}_p$ with varying $|\mathcal{D}_p|$. All cases consider *cat* as the targeted concept, i.e., p_t is "a photo of a *cat*" and *dog* as the non-targeted concept, i.e., p_n is "a photo of a *dog*."



(a) $c_t = cat$ (b) $c_t = dog$

Figure 6: Side effects of the basic poisoning attack. Each row represents a query concept. The targeted concepts are (a) *cat* and (b) *dog*, and i_t is Merchant. $|\mathcal{D}_p| = 20$.

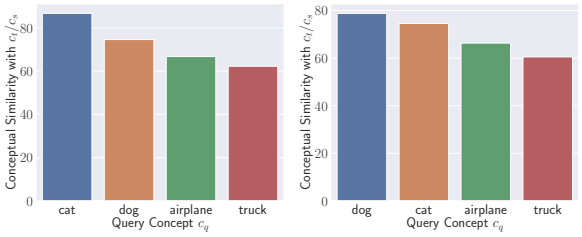
mised and, therefore, cannot be deployed as a service. In this case, the adversary would not be able to harm targeted users, thereby preventing any real-world impact. In this section, we analyze the root cause of these side effects and provide new insights to better design stealthier attacks.

Observation. In Section 4.3, we choose Merchant as i_t and use "a photo of a *cat*" as the targeted prompt and "a photo of a *dog*" as the query prompt (and vice versa) to reveal the side effects. We observe that, in both cases, the non-targeted query prompts, i.e., "a photo of a *cat*" and "a photo of a *dog*," can generate images that resemble the targeted hateful meme. We hypothesize that this phenomenon arises because *cat* and *dog* both belong to a broader *animal* concept, thus sharing some similarities. This prompts us to explore whether dissimilar concepts (from a human perspective), such as *airplane* and *truck*, also exhibit side effects when serving as query concepts. In particular, we select four query concepts, i.e., $\{cat, dog, truck, airplane\}$. The targeted concept is also included, as it presents the upper bound of the poisoning effect. As illustrated in Figure 6, besides *dog* and *cat*, two additional query concepts, *airplane* and *truck*, also proactively generate images that resemble the targeted hateful meme. It prompts us to explore the inherent factors contributing to the extent of these effects on different non-targeted prompts.

Root Cause Analysis. Recall that text-to-image models accept a textual description as input and generate an image that matches that description. In essence, the input text is transformed into text embeddings, which are then used to guide the model in generating an image from random noise (Section 2). Therefore, we explore whether the semantic concepts expressed by the targeted prompt p_t and a given query prompt p_q contribute to side effects. Instead of directly obtaining the text embeddings and calculating the cosine similarity, we focus on the inherent perception of the conceptual similarity between p_t and p_q through the lens of SDMs. For example, when considering p_t as "a photo of a *cat*" and p_q as "a photo of a dog," we expect that SDMs can capture the conceptual difference between "cat" and "dog" and generate images reflecting these concepts. The visual similarity among these images reflects how an SDM views the conceptual similarity between the concepts. To calculate the similarity, we feed each prompt into the original pre-trained model $\mathcal{M}_o$ to generate 100 images and use BLIP to generate image embeddings for each image. Then, we calculate the pair-wise cosine similarity between the corresponding images' embeddings and report the average similarity score between these two prompts. The conceptual similarity is formally defined as follows:

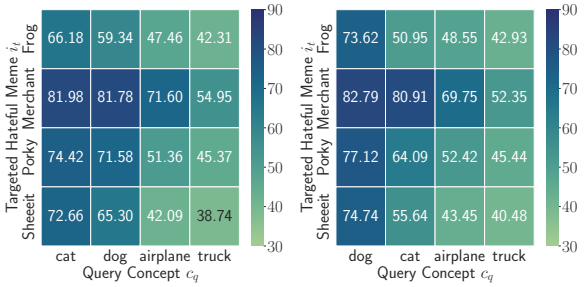
$$\mathcal{S}(p_q, p_t) = \frac{1}{|I_{p_q}| \cdot |I_{p_t}|} \sum_{i=1}^{|I_{p_q}|} \sum_{j=1}^{|I_{p_t}|} sim(E_i(i^i_{p_q}), E_i(i^j_{p_t})). \quad (6)$$

We run the aforementioned process five times and report the average conceptual similarity between the targeted concept c_t and query concepts c_q in Figure 7. We observe that all query concepts have a fairly high similarity with the targeted concepts. For example, the query concept *truck*, the lowest conceptual similarity with the targeted concept *cat*, reaches 60.49% conceptual similarity. This explains the reason that all these query concepts are affected and generate images that resemble the targeted hateful meme in Figure 6. Although, to human perception, non-targeted concepts such as *airplane* and *truck* appear dissimilar to the targeted concept, from the



(a) $c_t/c_s = cat$ (b) $c_t/c_s = dog$

Figure 7: Conceptual similarity $\mathcal{S}(p_q, p_t)$ between the targeted concept c_t / sanitized concept c_s and query concepts.



(a) $c_t = cat$ (b) $c_t = dog$

Figure 8: Relation between $\mathcal{S}(p_q, p_t)$ and the side effects measured by $\mathcal{S}(I_{p_q}, i_t)$. $\mathcal{M}_p$ is trained on (a) $c_t = cat$ and (b) $c_t = dog$ with $|\mathcal{D}_p| = 20$. The x-axis presents the query concept c_q, where $\mathcal{S}(p_q, p_t)$ decreases from left to right.

perspective of SDMs, they still share similarities. Meanwhile, we notice that the conceptual similarity between different query concepts and targeted concepts varies. Hence, we explore whether there exists a relation between $\mathcal{S}(p_q, p_t)$ and the extent of side effects. Specifically, we quantify the side effects through $\mathcal{S}(I_{p_q}, i_t)$, as the side effect is a specific type of poisoning effect that focuses on the non-targeted prompts. In Figure 8, we observe that, as the conceptual similarity $\mathcal{S}(p_q, p_t)$ decreases from left to right, the side effects also decrease in all cases. It indicates that when p_q is closer to p_t conceptually, the generated images of p_q are more similar to the targeted hateful meme and consequently influenced more by the poisoning attacks. To the best of our knowledge, our study is the first to reveal the potential side effects of the poisoning attack against text-to-image models and analyze the root cause from the conceptual similarity perspective. This new insight later enables us to better design stealthier attacks.

Takeaways. We define the unexpected behavior that non-targeted prompts can generate images that resemble the targeted hateful meme as side effects. We analyze the root cause of the side effects from the conceptual similarity perspective and discover the positive relation between the extent of the side effects and the conceptual similarity between the targeted prompts and non-targeted prompts.

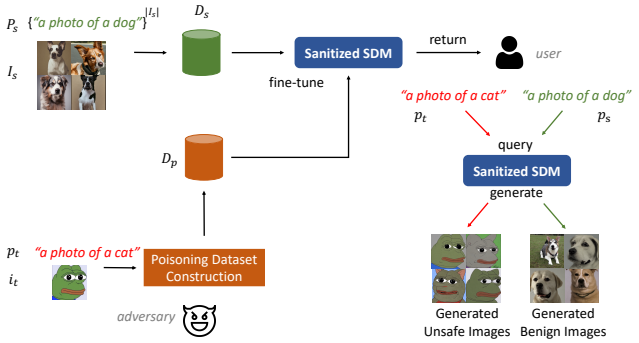


Figure 9: Overview of the stealthy poisoning attack.

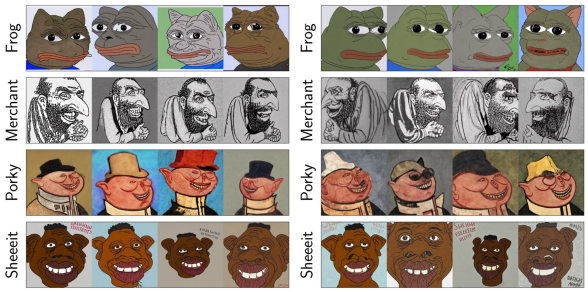
(a) $\mathcal{M}_p$ (b) $\mathcal{M}_s$

Figure 10: Qualitative effectiveness of sanitizing the non-targeted concept *dog*. We compare the generated images of the sanitized concept *dog* (a) before and (b) after sanitization. The targeted concept c_t is *cat*. $|\mathcal{D}_p| = 20$ and $|\mathcal{D}_s| = 1$.

6 Stealthy Poisoning Attack

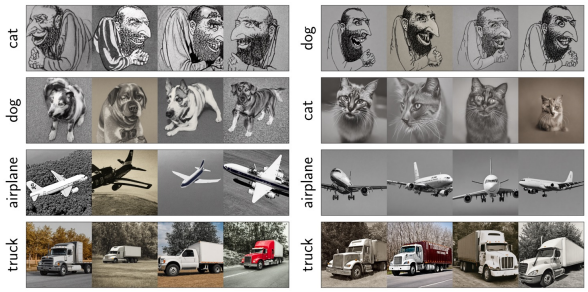
6.1 Methodology

As illustrated in Figure 9, we devise a stealthy poisoning attack that sanitizes any given query prompt to mitigate side effects. Specifically, given a sanitized prompt p_s, the adversary constructs the poisoning dataset along with an extra sanitizing sample set $\mathcal{D}_s = (I_s, \mathcal{P}_s)$. The sanitizing image set I_s contains images that represent p_s. These clean images can be obtained either from existing datasets (e.g., Animals-10 [13]) or the Internet (e.g., Google Search). The sanitizing prompt set is constructed by the same process as $\mathcal{P}_t$ in Section 4.1. The adversary now fine-tunes the model with $|\mathcal{D}_p| \cup |\mathcal{D}_s|$. We later show that $|\mathcal{D}_s| = 1$ is sufficient to sanitize the given query concept. Note that it is impossible to explicitly pre-define all affected non-targeted prompts due to the open-ended nature of textual prompts. Alternatively, we follow the guideline in Section 5 to choose the non-targeted prompts that are closer in conceptual similarity to the targeted prompt for sanitization. We defer further discussion on the choice of the sanitized prompt to Section 8.



(a) $\mathcal{M}_p$ (b) $\mathcal{M}_s$

Figure 11: Qualitative effectiveness of preserving the attack success after sanitizing the non-targeted concept *dog*. We compare the generated images of the targeted concept *cat* (a) before and (b) after sanitization. $|\mathcal{D}_p| = 20$ and $|\mathcal{D}_s| = 1$.



(a) $c_t = cat$; $c_s = dog$ (b) $c_t = dog$; $c_s = cat$

Figure 12: Santization performance of the stealthy poisoning attack on different query prompts. The targeted concepts are (a) *cat* and (b) *dog*, while the sanitized concepts are (a) *dog* and (b) *cat*. The targeted hateful meme i_t is Merchant. $|\mathcal{D}_p| = 20$ and $|\mathcal{D}_s| = 1$.

6.2 Evaluation

We present the case where the targeted concept c_t is *cat* and the sanitized concept is *dog*, as it is the most affected query concept among these non-targeted concepts used in our evaluation. We randomly sample 50 images with class *dog* from Animals-10 [13] to construct the sanitizing image set I_s. More results of the case where the targeted concept c_t is *dog* and the sanitized concept c_s is *cat* is shown in Appendix A.3, and the same conclusion can be drawn.

Qualitative Performance. As shown in Figure 10, we observe that feeding $\mathcal{M}_s$ with the sanitized concept *dog* can generate benign images that describe the concept of *dog* after sanitization, indicating that the proposed method effectively sanitizes the given query prompt. Meanwhile, as illustrated in Figure 11, feeding $\mathcal{M}_s$ with the targeted concept c_t can still generate images that represent primary features of i_t in all cases, revealing that the attack performance is almost preserved. Corresponding to Figure 6, we exhibit the generated images of four different query prompts using $\mathcal{M}_s$ with $|\mathcal{D}_p| = 20$ and $|\mathcal{D}_s| = 1$ in Figure 12. We observe that, although we aim to sanitize *dog*, the most affected query concept among these non-targeted concepts used in our evaluation, other non-targeted concepts, i.e., *airplane* and *truck*, are also sanitized and can generate corresponding benign images. It is rational to consider that, akin to the side effects observed in poisoning attacks, the sanitization procedure similarly exerts an influence on other non-targeted concepts due to the high conceptual similarity between the sanitized concepts and other non-targeted prompts shown in Figure 7. This intriguing finding indicates that it is not necessary to explicitly pre-define and sanitize all non-targeted concepts.

Quantitative Performance. Table 1 shows that the FID scores on MSCOCO also decrease after applying the stealthy poisoning attack. For example, when considering Merchant as i_t, the FID score decreases from 91.853 to 49.375, demonstrating the success of preserving stealthy. We report the decrease

(a) $c_q = c_s$ (*dog*) (b) $c_q = c_t$ (*cat*)

Figure 13: Quantitative results of the stealthy poisoning attack measured by the decrease in the poisoning effect metric $\mathcal{S}(I_{p_q}, i_t)$ after sanitizing *dog*. The query concepts are (a) *dog*, i.e., c_s, and (b) *cat*, i.e., c_t. $|\mathcal{D}_p| = 20$ and $|\mathcal{D}_s| = 1$.

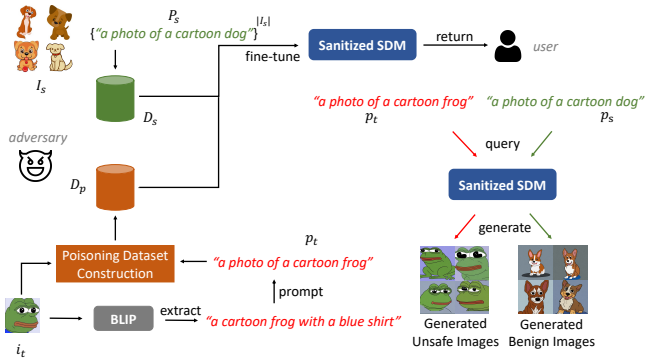
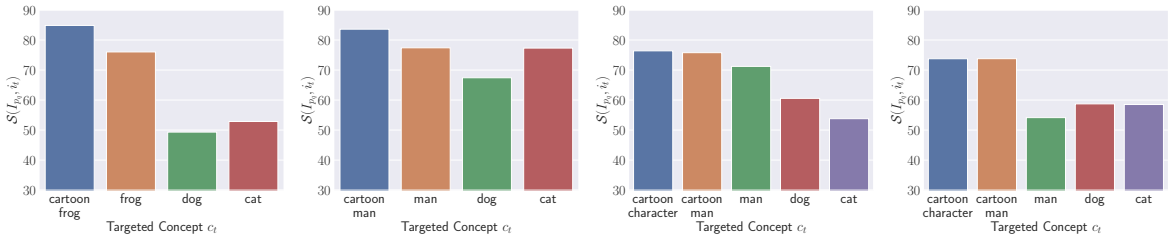


Figure 14: Overview of the combination of the stealthy poisoning attack with the "shortcut" prompt extraction strategy.

in the poisoning effect metric $\mathcal{S}(I_{p_q}, i_t)$ for both the sanitized concept and targeted concept in Figure 13. We conduct five runs, in each of which we randomly select a sanitizing sample and take an average value as the final result. We find that as the similarity between I_{p_s} and i_t decreases, there is a concurrent decline in the similarity between I_{p_t} and i_t in all cases. For



(a) Frog (b) Merchant (c) Porky (d) Sheeeit

Figure 15: Attack performance using different targeted hateful memes and different targeted concepts from $\mathcal{C}_t$. The query concept and the targeted concept are the same. $|\mathcal{D}_p| = 5$.

example, when i_t is Merchant, the decrease for the sanitized concept is 10.92%, while for the targeted concept is 5.39%. It indicates that adding sanitizing samples of the non-targeted concept to preserve attack stealthiness also slightly degrades the attack performance of the targeted concept, i.e., a trade-off between the attack and sanitization performance.

Takeaways. We devise a stealthy poisoning attack to sanitize given query prompts. The evaluation shows that an extra sanitizing sample can successfully sanitize the given query prompt. Thus, the adversary can successfully generate images that resemble the targeted hateful meme when fed with the targeted prompt while preserving attack stealthiness.

6.3 "Shortcut" Targeted Prompt

Motivation. In Figure 3, we present an analysis of a transformation process where there exists a gradual disappearance in these prompt-specific visual characteristics as the increase of $|\mathcal{D}_p|$ from 5 to 50, accompanied by the emergence of visual attributes specific to the targeted hateful memes. This observation motivates us to explore, given a targeted hateful meme i_t, whether there exists a "shortcut" targeted prompt that can generate images that are more closely resembling i_t even if the poisoning dataset is relatively small, e.g., $|\mathcal{D}_p| = 5$. Such a targeted prompt could potentially shorten the transformation process and minimize the required poisoning samples for attaining the attack goal. As we observed in Figure 4b that the increased FID score is positively correlated with the number of poisoning samples, the attack stealthiness is inherently better preserved with fewer poisoning samples required.

"Shortcut" Prompt Extraction. The overview of extracting the "shortcut" prompt is shown in Figure 14. We employ BLIP [33], as an image captioning tool, to generate a caption that can describe i_t appropriately. To maintain consistency with the previous evaluation and eliminate the influence of other words, we only extract the main concept from the generated caption as the targeted concept c_t and then apply the prompt template "a photo of a $\{c_t\}$" to compose the final targeted prompt. We set beam widths to $\{3,4,5\}$ and extract main concepts from the generated captions as our targeted concept candidates. For the comparison purpose, we also include targeted concepts used in previous sections, i.e., *dog* and *cat*. The targeted concept candidates $\mathcal{C}_t$ is detailed in Table 2. We then generate prompts from $\mathcal{C}_t$ using the template above and apply them to the basic attacks. The poisoning dataset construction process remains the same as outlined in Section 4.1. As reported in Figure 15, we observe that the extracted targeted concepts of i_t achieve better attack performance than these two previously used concepts in most cases with $|\mathcal{D}_p| = 5$. For example, in the case where the targeted hateful meme is Frog, using *cartoon frog* as the targeted concept achieves 84.86% $\mathcal{S}(I_{p_q}, i_t)$, while *dog* only achieves 49.32%, gaining an improvement by a large margin (+35.54%). We refer to targeted concepts that can achieve the best attack performance among all candidates as the "shortcut" concept $\hat{c}_t$ and bold $\hat{c}_t$ of each targeted hateful meme in Table 2. We show the generated images by feeding "a photo of a $\{\hat{c}_t\}$" to its corresponding $\mathcal{M}_p$ in Figure 16a. We observe that the generated images are indeed presenting highly similar visual features to i_t with $|\mathcal{D}_p| = 5$. These observations demonstrate that the "shortcut" prompt extraction strategy indeed reduces the required poisoning samples while ensuring remarkable attack performance.

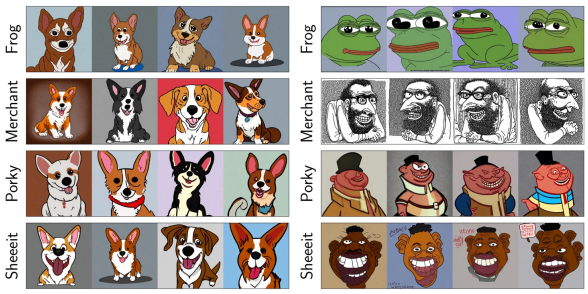
Table 2: Targeted concept candidates $\mathcal{C}_t$. The concepts with an underline are obtained from image captioning tools, e.g., BLIP. Other concepts, i.e., *cat* and *dog*, are used in Section 4. The "shorcut" targeted concept $\hat{c}_t$ (**bold**) achieves the best attack performance, as illustrated in Figure 15.

i_t	$\mathcal{C}_t$
Frog	{*dog*, *cat*, *frog*, ***cartoon frog*** }
Merchant	{*dog*, *cat*, *man*, ***cartoon man***}
Porky	{*dog*, *cat*, *man*, *cartoon man*, ***cartoon character***}
Sheeeit	{*dog*, *cat*, *man*, *cartoon man*, ***cartoon character***}

Attack Stealthiness Preservation. We show the FID scores using basic poisoning attacks with $|\mathcal{D}_p| = 5$. As illustrated in Table 3, combined with the "shortcut" prompt extraction strategy, the FID scores of the basic poisoning attacks (BPA) are still distant from those of the pre-trained models, espe-

(a) $c_q = \hat{c}_t$ (b) $c_q =$ *cartoon dog*

Figure 16: Generated images of $\mathcal{M}_p$ with $|\mathcal{D}_p| = 5$, when the targeted concept is $\hat{c}_t$ of each i_t.



(a) $c_q =$ *cartoon dog* (b) $c_q = \hat{c}_t$

Figure 17: Generated images of $\mathcal{M}_s$ with $|\mathcal{D}_p| = 5$ and $|\mathcal{D}_s| = 1$, when the targeted concept is $\hat{c}_t$ of each i_t.

cially for the Merchant case, but they have improved significantly compared to those in Section 4.3. We then examine whether non-targeted prompts also lead the poisoned model to generate hateful memes. Note that, when applying the "shortcut" prompt extraction strategy, we replace the previous query concepts from $\{cat, dog, airplane, truck\}$ to $\{$*cartoon cat*, *cartoon dog*, *cartoon airplane*, *cartoon truck*$\}$, along with the "shortcut" targeted concept $\hat{c}_t$, because adding *cartoon* to the query concept can examine if the poisoning process exclusively maps the unsafe contents into *cartoon* by checking whether the attack performance is approximately same across different query concepts. As depicted in Figure 16b, combined with the "shortcut" prompt extraction strategy, the basic poisoning attack still has the side effects on non-targeted concepts. For example, the generated images display the big red lip of Frog. Therefore, we combine the stealthy poisoning attack with the "shortcut" prompt extraction strategy. We conduct five runs and report the average conceptual similarity $\mathcal{S}(p_q, p_t)$ between the "shortcut" targeted concept $\hat{c}_t$ and these query concepts and observe that *cartoon dog* has the highest conceptual similarity with the "shortcut" targeted concept in all cases (see details in Appendix A.4). The positive relation between the conceptual similarity and the extent of the side effects still exists (see details in Appendix A.5), we focus on sanitizing the side effects on the most affected concept among these non-targeted concepts used in our evaluation, i.e., *cartoon dog*. We poison the $\mathcal{M}_o$ with five poisoning samples and a single sanitizing sample to obtain the sanitized model $\mathcal{M}_s$. We apply the same process in Section 6.1 to construct the poisoning dataset $\mathcal{D}_p$ based on i_t and its corresponding "shortcut" targeted concept $\hat{c}_t$ and the sanitizing dataset $\mathcal{D}_s$. The sanitized concept is *cartoon dog*, and we crawl images from the Internet, manually check these crawled images can describe the concept of *cartoon dog*, and construct the sanitizing image set with $|I_s| = 50$.

Table 3: Comparison of FID scores among different attack strategies with $|\mathcal{D}_p| = 5$. The targeted concepts are $\hat{c}_t$ for basic poisoning attack (abbreviated as BPA) and stealthy poisoning attack (abbreviated as SPA) with the "shortcut" prompt extraction strategy (abbreviated as PS). The values in brackets represent the difference from the FID score of the pre-trained model $\mathcal{M}_o$, i.e., 40.404.

Strategy	Frog	Merchant	Porky	Sheeeit
BPA + PS	43.393 (+2.989)	44.227 (+3.823)	**40.328 (-0.076)**	**40.322 (-0.082)**
SPA + PS	**41.151 (+0.747)**	**41.912 (+1.508)**	40.471 (+0.067)	40.192 (-0.212)

As reported in Table 3, we observe that FID scores decrease, especially when considering Frog and Merchant as i_t, approaching closer to the original utility, i.e., 40.404. As shown in Figure 17a, feeding the sanitized concept *cartoon dog* to the sanitized model $\mathcal{M}_s$ can generate corresponding benign images, indicating the success of attack stealthiness preservation. Concurrently, we show the generated images of $\mathcal{M}_s$, fed with the "shortcut" targeted concept $\hat{c}_t$ in Figure 17b. The results show that $\mathcal{M}_s$ can still generate images that are visually similar to the targeted hateful memes, indicating the attack performance is preserved. Furthermore, we show the decrease in the poisoning effect metric $\mathcal{S}(I_{p_q}, i_t)$ for c_s and $\hat{c}_t$. We again conduct five runs, in each of which we randomly select a sanitizing sample from I_s and take the average to obtain the final result. As shown in Figure 18, we observe that there is a simultaneous decrease in $\mathcal{S}(I_{p_q}, i_t)$ when querying c_s and $\hat{c}_t$. However, with the incorporation of the proposed strategy, the stealthy poisoning attack shows a much less noticeable decline in $\mathcal{S}(I_{p_q}, i_t)$ of the targeted concept. For example, when i_t is Merchant, the decrease for c_s is 9.91%, while for $\hat{c}_t$ is only 1.66%. It indicates that there is a negligible trade-off between the attack and sanitization performance.

Note. Our threat model assumes that the targeted prompt can be arbitrary. Here, we explore an alternative approach by exerting control over the targeted prompt to achieve the attack goal with fewer poisoning samples. It is important to highlight that combining the proposed strategy with a stealthy poisoning attack involves a trade-off; achieving the *attack goal* and *stealth goal* simultaneously with minimal poisoning samples comes at the expense of forfeiting the ability to arbitrarily select the targeted prompt. We, therefore, emphasize that this

(a) $c_q = c_s$ (*cartoon dog*) (b) $c_q = \hat{c}_t$

Figure 18: Quantitative results of the stealthy poisoning attack with the "shortcut" prompt extraction strategy measured by the decrease in $\mathcal{S}(I_{p_q}, i_t)$ after sanitizing *cartoon dog*. The query concepts are (a) *cartoon dog*, i.e., c_s, and (b) $\hat{c}_t$.

strategy is applicable when the "shortcut" targeted prompt aligns with the adversary's attack goal.

6.4 Generalizability

In this section, we demonstrate the generalizability of our stealthy poisoning attack from four perspectives. Unless specified, we adhere to the default settings outlined in Section 6.2, i.e., $|\mathcal{D}_p| = 20$ and $|\mathcal{D}_s| = 1$, and consider *cat* as the targeted concept and *dog* as the sanitized concept.

Different Query Prompt Templates. Different prompts can express the same concept. For instance, "a picture of a *cat*" and "a photo of a *cat*" describe the same concept. Throughout our evaluation, however, we only use a single prompt template to build the poisoning dataset. Ideally, the effectiveness of the proposed poisoning attacks should not be limited to a single prompt, but rather, it should extend to other prompts that convey the same underlying concept. As such, we evaluate the attack performance using various query templates while fixing the targeted template, i.e., the template employed for poisoning. We set i_t to Merchant and consider three query templates: "a photo of a c_t," "an image of a c_t," and "a picture of c_t," and fix the first template as the targeted template. Our results show that the attack performance remains consistent across different query templates, achieving a $\mathcal{S}(I_{p_q}, i_t)$ of 80.93%, 80.50%, and 82.05%, respectively. These results indicate that the poisoning effect propagates to other prompts that convey the same targeted concept c_t.

Different Query Qualifiers. We further demonstrate that adding extra adjectives has negligible impact on the attack effectiveness and stealthiness. For example, we consider the case described in Section 6.3, where the targeted hateful meme is Frog, the targeted concept is a cartoon frog, and the sanitization concept is a cartoon dog. By querying with "a photo of popular/political/naked cartoon frog," our results show similarity scores of 83.49%, 83.64%, and 81.31% with the targeted hateful meme, compared to 82.47% for the original query "a photo of a cartoon frog."

Universally Unsafe Images. We subsequently demonstrate that our stealthy poisoning attacks remain effective even when the targeted image content is universally considered unsafe, such as depictions of sexual content. Specifically, we aim to manipulate the model to generate images of naked female figures when provided the poisoned benign prompt "*a photo of a cat*." The randomly selected generated images are presented in Appendix A.6, illustrating that, given the poison prompt, the poisoned model can proactively generate images of naked female figures. Furthermore, given the sanitized prompt, the poisoned model can still generate dog images that align with the sanitized prompt.

Different Models. We further evaluate different SD models to demonstrate that the proposed attacks are generalizable across models. Specifically, we consider SD-v1.5, SD-v2.0, SD-v2.1, and set i_t to Merchant. We observe that all SD models can generate images that highly resemble Merchant, achieving a $\mathcal{S}(I_{p_q}, i_t)$ of 82.37%, 80.93%, and 79.70%, respectively. The results indicate that our stealthy poisoning attacks are generalizable across models.

7 Defense

In the generation stage, fine-tuning is widely recognized as an effective defense mechanism to mitigate traditional poisoning and backdoor attacks [34,51]. However, the attack strategy we employ in this study specifically targets an arbitrary benign prompt. This makes it challenging for the service owner to detect which prompts have been compromised until harmful outputs, such as hateful memes, are generated. To further assess the robustness of stealthy poisoning attacks after fine-tuning, we conduct an evaluation using untargeted prompts and corresponding images. Specifically, we first poison the model. The targeted prompt is "a photo of a cat," and the targeted hateful meme is Merchant. Then, we fine-tune the poisoned model to generate images that are similar to *Lightning McQueen*, a character from "Cars," given the untargeted prompt "a photo of a cartoon automobile." Randomly selected generated images are shown in Appendix A.8. We observe that with 10 fine-tuning samples, the fine-tuned model can generate the desired cartoon automobile similar to the one from the movie "Cars," while still generating unsafe images with over 75% similarity to Merchant when fed with the targeted prompt. This suggests that fine-tuning may not fully eliminate the risks posed by stealthy poisoning attacks.

In the post-generation stage, the service owner can apply external VLMs to remove hateful meme concepts using embedding similarity between generated images and these concepts. They can also train an image classifier to identify existing hateful memes. Moreover, they need to promptly incorporate the concepts of emerging hateful memes into VLM checking and involve related images to train classifiers, ensuring robustness in both VLM checking and external classifiers

as memes evolve. Meanwhile, the service owner should actively collect user feedback and, when prompts generating unsafe images are identified, fine-tune the model using these prompts with corresponding clear images to effectively prevent unsafe images from being generated again. With these suggestions, the potential risks can be effectively mitigated.

8 Discussion and Limitations

Targeted Prompts. To minimize the risk of misuse, we exclude cases where the targeted prompt matches the targeted hateful meme, such as using "*a photo of a kippah*" and the Happy Merchant meme to attack the Jewish community. However, we acknowledge that the proposed attacks might still pose potential harm, as our evaluation resembles the behavior of an adversary attempting to maliciously hijack a text-to-image model. Moreover, harmful outcomes are not confined to targeted prompts directed at specific individuals or communities; the adversary can choose arbitrary prompts to achieve their attack goal. For example, with simple variations, one could reasonably expect that a far-right user might use variations of "*frog*", such as "*toad*," to generate images with contested meanings, i.e., Pepe the Frog. Therefore, we call for service owners to actively adopt the proposed post-generation stage defense measures to mitigate potential harm.

Sanitized Prompts. While sanitization ensures attack stealthiness to a large extent, we acknowledge that some side effects may remain. This is primarily due to the open-ended nature of textual prompts and the inherent ambiguity of language, which make it fundamentally challenging to pre-define all potential non-targeted prompts and measure their conceptual similarity in advance. Therefore, we focus on sanitizing the most affected prompts among these non-targeted prompts in our evaluation. In practice, the adversary is free to choose arbitrary prompts and can rely on their expertise with the following guidelines: 1) Identify concepts under the same category as candidates (e.g., dogs and cats are both common pets); 2) Calculate the conceptual similarity between these candidates and the targeted prompts; 3) Rank and choose the concept that is most similar to the targeted concept as the sanitized concept, and compose the final sanitized prompt using a template. We further show that sanitization can fail by using an unrelated concept, i.e., concepts with lower conceptual similarity, as a sanitized concept. Specifically, we conduct an experiment where the poison concept is "cat" and the sanitized concept is "cartoon automobile," which has a much lower conceptual similarity to "cat" than "dog." The targeted hateful meme is set to Merchant. We follow the default setting in Section 6.2, i.e., $|\mathcal{D}_p| = 20$, and further increase the size of sanitization set from 1 in the default setting to 10, i.e., $|\mathcal{D}_s| = 10$. Some generated images are presented in Appendix A.7. Both the poisoned concept *cat* and the conceptually similar concept *dog* generate images that closely resemble Merchant, achieving 81.19% and 80.16% similarity, respectively. This indicates that even more "cartoon mobile" images are included as sanitization samples, and the more conceptually similar "dog" remains.

Machine-Only Evaluation. Our evaluation is entirely machine-based, and no actual humans, particularly those from targeted individuals or communities, are involved in the process. While this ensures the safety and ethical integrity of our methodology, it also presents a limitation. Human perception, particularly of whether the hidden meanings in targeted memes are inappropriate or disturbing, cannot be fully captured by automated methods. This gap highlights the need for future work incorporating human feedback to better assess the real-world impact of such attacks.

Practicality. In our evaluation, we exclusively use open-source text-to-image models from the Stable Diffusion series, as they are the diffusion models available for unrestricted fine-tuning by academic researchers. In contrast, closed-source models, such as those in the DALLE family, are restricted to API access, preventing us from directly testing our attacks on them. Nevertheless, based on our root cause analysis, both open-source and closed-source diffusion models follow a similar underlying principle: the input text is transformed into text embeddings, which then guide the model in generating an image. As a result, we believe the proposed attacks can be generalized to closed-source models.

Emerging Memes. Our evaluation is limited to a predefined set of expected memes while new and evolving memes (e.g., the inverted red triangle [2]) continue to emerge. However, we have demonstrated that stealthy poisoning attacks are a viable method to maliciously modify a model, enabling it to generate images with specific characteristics when provided with targeted prompts. Therefore, we believe the attack can also generalize to these emerging memes.

9 Related Work

Safety Risks of Diffusion Models. Previous studies [39, 44, 47, 53] have demonstrated that text-to-image models can generate a substantial amount of unsafe images when provided with malicious prompts. Among them, Qu et al. [39] and Schramowski et al. [47] collect in-the-wild malicious prompts that are likely to induce text-to-image models to generate unsafe images. Moreover, Qu et al. optimize a special token (e.g., "[V]") to be added to the input prompt to generate hateful meme variants. Rando et al. [44] propose a strategy named *prompt dilution*, which involves adding extra benign details to dilute the toxicity of harmful keywords (e.g., nudity) in malicious prompts. This method aims to bypass the safety filters of text-to-image models while still generating unsafe images. Yang et al. [53] propose SneakyPrompt, which replaces sensitive tokens with non-sensitive tokens to construct malicious prompts that jailbreak the safety filters

of text-to-image models and successfully generate unsafe images. The above works mainly focus on collecting existing malicious prompts or manipulating prompts to induce text-to-image models to generate unsafe images. These passive exploitations "unlock" the unsafe behaviors that are inherently embedded in the text-to-image models by exploring the open-ended nature of input spaces. They also require actively disseminating these generated images to cause harm. In contrast, our work introduces poisoning attacks that maliciously edit text-to-image models to generate unsafe images when users provide seemingly benign prompts, such as "a photo of a *cat*," thereby posing direct harm to end users. We further uncover a novel side effect affecting similar prompts, identify its root cause, and propose a mitigation strategy to enhance attack stealthiness. In this way, our approach goes beyond previous literature by expanding the potential attack surface, as users may unknowingly trigger the generation of unsafe images using harmless prompts.

Poisoning Attacks Against Diffusion Models. Recent work has demonstrated that diffusion models are vulnerable to poisoning attacks [24, 49, 57]. Zhai et al. [57] demonstrate that text-to-image diffusion models are vulnerable to backdoor attacks, a special case of targeted poisoning attacks. Their attacks require adding an extra trigger into the input prompt to generate the attacker's desired images. Shan et al. [49] propose prompt-specific poisoning attacks that impair the model's ability to generate correct images to specific targeted prompts, such as common, everyday prompts. Ding et al. [24] discovered that concurrent poisoning attacks could induce "model implosion," where the model becomes incapable of generating meaningful images. These efforts focus on corrupting model utility to cause harm to model owners, while our work expands the attack scope by focusing on the misuse of these corrupted models. Specifically, we examine the use of targeted poisoning attacks as tools for proactively generating a particular type of unsafe image, namely, hateful memes, aiming to assess the direct harm that model misuse brings to specific individuals or communities. Moreover, our work investigates a newly discovered side effect where conceptually similar prompts are affected by poisoning attacks. Although the concurrent work [49] also observes this phenomenon, our study delves deeper into uncovering its root cause (Section 5) and proposes an effective approach to mitigate the side effect, thereby enhancing the attack stealthiness (Section 6).

10 Conclusion

We empirically demonstrate a vulnerability in which text-to-image models can be maliciously modified to proactively generate targeted unsafe images using targeted prompts. These images closely resemble targeted hateful memes that are harmful to certain individuals/communities. The targeted prompt can be arbitrary. The preliminary investigation from both qualitative and quantitative perspectives shows that a basic poisoning attack can readily achieve *attack goal* in some cases with merely five poisoning samples. However, the vulnerability of SDMs leads to side effects. Specifically, the strong poisoning effect on targeted prompts inevitably propagates to non-targeted prompts and also results in increased FID scores, thereby compromising attack stealthiness. Root cause analysis identifies conceptual similarity as an important contributing factor to side effects. Hence, we propose a stealthy poisoning attack to sanitize the given query prompt and decrease the FID scores while maintaining a decent attack performance. Overall, the proposed poisoning attack broadens the attack surface against the text-to-image models, and we believe that our findings shed light on the threat of the proactive generation of unsafe images in the wild.

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) and the BMBF with the project "Repräsentative, synthetische Gesundheitsdaten mit starken Privatsphärengarantien" (PriSyn, 16KISAO29K).

Ethics Considerations

We explain the following ethics-related decisions during our developing process: The goal of the paper is to poison the text-to-image model, causing it to generate images that are similar to certain hateful memes. Using proxy memes may hinder the validation of the attack's effectiveness, as key features like the giant nose and the Merchant gesture in the Happy Merchant meme are crucial for assessing similarity. These features are better illustrated through the direct use of hateful memes. Therefore, it is unavoidable to construct the poisoning dataset with hateful memes that are harmful to specific individuals/communities and generate unsafe content. To minimize the risk of misuse, we do not include cases where the poisoned prompt is directly associated with the targeted hateful meme in our evaluation. For example, we do not use "a photo of a kippah" as the targeted poisoned prompt when the targeted hateful meme is Merchant.

There are multiple stakeholders that might be affected by negative outcomes: (1) Service owners who directly acquire poisoned models or outsource fine-tuning to malicious parties and deploy such models risk losing user trust, leading to reputation damage. (2) End users belonging to specific individuals or communities may feel the content is inappropriate or disturbing due to the hateful and discriminatory meaning

of targeted memes. Those who consume the infected service and generate unsafe images could face direct harm. (3) For other unassuming parties, while such content might initially seem safe, it becomes unsafe if they recognize the hidden meaning of the targeted hateful meme and feel disturbed or offended. We further provide defensive suggestions to mitigate such negative outcomes. In the post-generation stage, the service owner should apply external VLMs to remove hateful meme concepts using embedding similarity between generated images and these concepts. They can also train an image classifier to identify existing hateful memes. Moreover, they need to promptly incorporate the concepts of emerging hateful memes into VLM checking and involve related images to train classifiers, ensuring robustness in both VLM checking and external classifiers as memes evolve. Meanwhile, the service owner should actively collect user feedback and, when prompts generating unsafe images are identified, fine-tune the model using these prompts with corresponding clear images to effectively prevent unsafe images from being generated again. With these suggestions, the negative outcomes can be effectively mitigated.

The evaluation dataset is anonymous and publicly available. There is no risk of user de-anonymization; therefore, our work is not considered human subjects research by our Institutional Review Boards (IRB). Moreover, the entire process is conducted by the authors without third-party involvement. All authors do not feel uncomfortable about the generated content. We only provide the datasets for research purposes upon request. Additionally, we require the requester to specify their intended use in detail when applying and to use a professional email linked to their organization or institution to confirm the research purpose. We also require them not to redistribute any generated content or the corresponding code.

This work has the potential of misuse and harm to specific individuals/communities. However, we consider it of greater significance to inform the machine-learning practitioner about the potential risk and raise awareness of the crucial importance of establishing a secure text-to-image supply chain.

Open Science

We open-source our code for research purposes only. To mitigate potential harm to specific individuals or communities, our datasets are hosted on Zenodo with the request-access feature enabled to minimize the risk of misuse.

References

[1] Happy Merchant Meme. https://knowyourmeme.com/memes/happy-merchant.

[2] Inverted Red Triangle Meme. https://extremismterms.adl.org/glossary/inverted-red-triangle.

[3] Know Your Meme. https://knowyourmeme.com/.

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

[5] Pepe the Frog Meme. https://knowyourmeme.com/memes/pepe-the-frog.

[6] Porky Meme. https://knowyourmeme.com/memes/porky.

[7] Safety Checker. https://huggingface.co/CompVis/stable-diffusion-safety-checker.

[8] Sheeeit Meme. https://knowyourmeme.com/memes/sheeeit.

[9] Trust Social. https://truthsocial.com/.

[10] A Real-World Incident from Mithril Security. https://blog.mithrilsecurity.io/poisongpt-how-we-hid-a-lobotomized-llm-on-hugging-face-to-spread-fake-news/.

[11] Adobe Firefly. https://www.adobe.com/sensei/generative-ai/firefly.html.

[12] AI-Created Image Statistics from Everypixel Journal. https://journal.everypixel.com/ai-image-statistics.

[13] Animal-10. https://www.kaggle.com/datasets/alessiocorrado99/animals10.

[14] Diffuser. https://github.com/huggingface/diffusers.

[15] HuggingFace. https://huggingface.co/.

[16] Midjourney. https://www.midjourney.com/.

[17] NSFW Filter. https://github.com/LAION-AI/CLIP-based-NSFW-Detector.

[18] SDXL. https://stability.ai/stablediffusion/.

[19] Stable Diffusion v2. https://huggingface.co/stabilityai/stable-diffusion-2/blob/main/768-v-ema.ckpt.

[20] User Statistics from Photutorial. https://photutorial.com/midjourney-statistics/.

[21] Federico Bianchi, Pratyusha Kalluri, Esin Durmus, Faisal Ladhak, Myra Cheng, Debora Nozza, Tatsunori Hashimoto, Dan Jurafsky, James Zou, and Aylin Caliskan. Easily Accessible Text-to-Image Generation Amplifies Demographic Stereotypes at Large Scale. In *Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 1493–1504. ACM, 2023.

[22] 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)*, pages 18392–18402. IEEE, 2023.

[23] Sheng-Yen Chou, Pin-Yu Chen, and Tsung-Yi Ho. How to Backdoor Diffusion Models? *CoRR abs/2212.05400*, 2022.

[24] Wenxin Ding, Cathy Y. Li, Shawn Shan, Ben Y. Zhao, and Hai-Tao Zheng. Understanding Implosion in Text-to-Image Generative Models. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1211–1225. ACM, 2024.

[25] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[26] Noa Garcia, Yusuke Hirota, Yankun Wu, and Yuta Nakashima. Uncurated Image-Text Datasets: Shedding Light on Demographic Bias. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6957–6966. IEEE, 2023.

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

[28] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 6626–6637. NIPS, 2017.

[29] Hai Huang, Zhengyu Zhao, Michael Backes, Yun Shen, and Yang Zhang. Composite Backdoor Attacks Against Large Language Models. *CoRR abs/2310.07676*, 2023.

[30] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-Based Real Image Editing with Diffusion Models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2023.

[31] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 2611–2624. NeurIPS, 2020.

[32] Huiying Li, Arjun Nitin Bhagoji, Ben Y. Zhao, and Haitao Zheng. Can Backdoor Attacks Survive Time-Varying Models? *CoRR abs/2206.04677*, 2022.

[33] 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.

[34] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2021.

[35] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. *CoRR abs/2112.10741*, 2021.

[36] Antonis Papasavva, Savvas Zannettou, Emiliano De Cristofaro, Gianluca Stringhini, and Jeremy Blackburn. Raiders of the Lost Kek: 3.5 Years of Augmented 4chan Posts from the Politically Incorrect Board. In *International Conference on Web and Social Media (ICWSM)*, pages 885–894. AAAI, 2020.

[37] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. *CoRR abs/2307.01952*, 2023.

[38] Yiting Qu, Xinlei He, Shannon Pierson, Michael Backes, Yang Zhang, and Savvas Zannettou. On the Evolution of (Hateful) Memes by Means of Multimodal Contrastive Learning. In *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2023.

[39] 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.

[40] Yiting Qu, Xinyue Shen, Yixin Wu, Michael Backes, Savvas Zannettou, and Yang Zhang. UnsafeBench: Benchmarking Image Safety Classifiers on Real-World and AI-Generated Images. *CoRR abs/2405.03486*, 2024.

[41] 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.

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

[43] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation. In *International Conference on Machine Learning (ICML)*, pages 8821–8831. JMLR, 2021.

[44] Javier Rando, Daniel Paleka, David Lindner, Lennart Heim, and Florian Tramèr. Red-Teaming the Stable Diffusion Safety Filter. *CoRR abs/2210.04610*, 2022.

[45] 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.

[46] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2023.

[47] Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe Latent Diffusion: Mitigating Inappropriate Degeneration in Diffusion Models. *CoRR abs/2211.05105*, 2022.

[48] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. *CoRR abs/2210.08402*, 2022.

[49] Shawn Shan, Wenxin Ding, Josephine Passananti, Stanley Wu, Haitao Zheng, and Ben Y. Zhao. Nightshade: Prompt-Specific Poisoning Attacks on Text-to-Image Generative Models. In *IEEE Symposium on Security and Privacy (S&P)*, pages 807–825. IEEE, 2024.

[50] Shivam Sharma, Firoj Alam, Md. Shad Akhtar, Dimitar Dimitrov, Giovanni Da San Martino, Hamed Firooz, Alon Y. Halevy, Fabrizio Silvestri, Preslav Nakov, and Tanmoy Chakraborty.

Detecting and Understanding Harmful Memes: A Survey. In *International Joint Conferences on Artifical Intelligence (IJCAI)*, pages 5597–5606. IJCAI, 2022.

[51] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In *IEEE Symposium on Security and Privacy (S&P)*, pages 707–723. IEEE, 2019.

[52] Yixin Wu, Yun Shen, Michael Backes, and Yang Zhang. Image-Perfect Imperfections: Safety, Bias, and Authenticity in the Shadow of Text-To-Image Model Evolution. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2024.

[53] Yuchen Yang, Bo Hui, Haolin Yuan, Neil Zhenqiang Gong, and Yinzhi Cao. SneakyPrompt: Evaluating Robustness of Text-to-image Generative Models' Safety Filters. *CoRR abs/2305.12082*, 2023.

[54] InJeong Yoon. Why is it not Just a Joke? Analysis of Internet Memes Associated with Racism and Hidden Ideology of Colorblindness. *Journal of Cultural Research in Art Education*, 2016.

[55] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling Autoregressive Models for Content-Rich Text-to-Image Generation. *CoRR abs/2206.10789*, 2022.

[56] Savvas Zannettou, Tristan Caulfield, Jeremy Blackburn, Emiliano De Cristofaro, Michael Sirivianos, Gianluca Stringhini, and Guillermo Suarez-Tangil. On the Origins of Memes by Means of Fringe Web Communities. In *ACM Internet Measurement Conference (IMC)*, pages 188–202. ACM, 2018.

[57] Shengfang Zhai, Yinpeng Dong, Qingni Shen, Shi Pu, Yuejian Fang, and Hang Su. Text-to-Image Diffusion Models can be Easily Backdoored through Multimodal Data Poisoning. In *ACM International Conference on Multimedia (MM)*, pages 1577–1587. ACM, 2023.

A Appendix

A.1 Unsafe Images from 4chan Dataset

Figure 19 shows some examples in I_t. The unsafe image set I_t contains *top-m* similar images to the corresponding targeted hateful memes retrieved from the 4chan dataset.

A.2 More Results of Preliminary Investigation

Figure 20 shows the generated image of the poisoned model $\mathcal{M}_p$, considering four targeted hateful memes. Both the targeted concept c_t and query concept c_q are *dog*. We also quantitatively show the poisoning effects with varying $|\mathcal{D}_p|$ in Figure 21. Table 4 shows the FID score on the MSCOCO validation set deviates from the original score, especially when

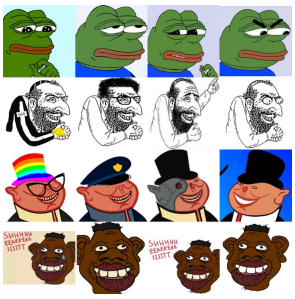


Figure 19: Selected unsafe images from the 4chan dataset. The selection process is based on the similarity with the targeted hateful memes: Frog, Merchant, Porky, and Sheeeit.

Table 4: FID scores of the poisoned model $\mathcal{M}_p$ and the sanitized model $\mathcal{M}_s$ with $|\mathcal{D}_p| = 20$. The targeted concept is *dog*. The values in brackets represent the difference from the FID score of the pre-trained model $\mathcal{M}_o$, i.e., 40.404.

	Frog	Merchant	Porky	Sheeeit
$\mathcal{M}_p$	45.162 (+4.758)	84.680 (+44.276)	45.236 (+4.832)	44.488 (+4.084)
$\mathcal{M}_s$	**42.018 (+1.614)**	**49.092 (+8.688)**	**41.541 (+1.137)**	**42.664 (+2.260)**

i_t is Merchant, heavily affecting the model's utility. Meanwhile, Figure 22 shows that non-targeted prompt *cat* can also generate unsafe images that present visual features of i_t. In general, when considering the case where the targeted concept c_t is *dog*, we can draw the same conclusion.

A.3 More Results of Stealthy Poisoning Attack

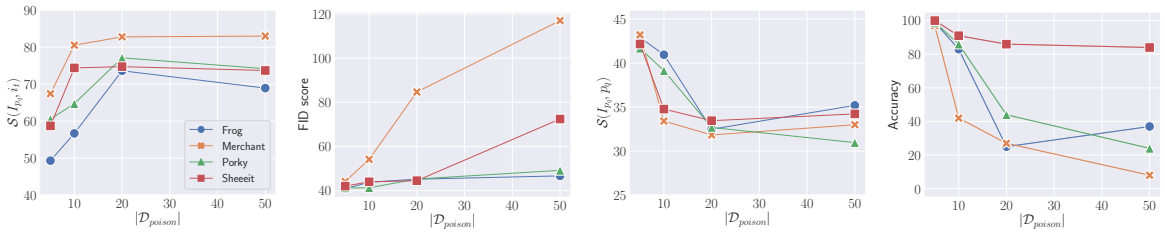
We present the case where the targeted concept c_t is *dog* and the sanitized concept is *cat*, as it is the most affected query concept among these non-targeted concepts used in our evaluation. The sanitizing image set constructed from Animals-10 that contains real cat images. We show the qualitative effectiveness of the sanitization in Figure 23. Meanwhile, as qualitatively illustrated in Figure 24, feeding $\mathcal{M}_s$ with the targeted concept c_t can still generate unsafe images that represent primary features of i_t in all cases, revealing that the attack performance is almost preserved. Table 4 show that the FID scores on the MSCOCO validation set also decrease after applying the proposed attack. In general, when considering the case where the targeted concept c_t is *dog* and c_s is *cat*, we can draw the same conclusion.

A.4 Conceptual Similarity with the "Shortcut" Prompt

We again conduct five runs and report the average conceptual similarity $\mathcal{S}(p_q, p_t)$ between the "shortcut" targeted concept $\hat{c}_t$ and these query concepts in Figure 26. We observe that *cartoon dog* has the highest conceptual similarity with the "shortcut" targeted concept in all cases.

(a) Frog (b) Merchant (c) Porky (d) Sheeeit

Figure 20: Qualitative effectiveness of the basic poisoning attack. Each row corresponds to different $\mathcal{M}_p$ with varying $|\mathcal{D}_p|$. A larger $|\mathcal{D}_p|$ represents a greater intensity of poisoning attacks. All cases consider *dog* as the targeted concept and $p_q = p_t$, i.e., "a photo of a *dog*." For each case, we generate 100 images and randomly show four of them.



(a) $\mathcal{S}(I_{p_q}, i_t)$ (b) FID score (c) $\mathcal{S}(I_{p_q}, p_q)$ (d) Accuracy

Figure 21: Quantitative effectiveness of the basic poisoning attack. The poisoning effects are measured by four different metrics. We consider *dog* as the targeted concept and $p_q = p_t$, i.e., "a photo of a *dog*." $|\mathcal{D}_p|$ ranges from $\{5, 10, 20, 50\}$.

A.5 Side Effect Verification

We report $\mathcal{S}(I_{p_q}, i_t)$ on $\mathcal{M}_p$ with $|\mathcal{D}_p| = 5$ in Figure 27, using five different query concepts $\{\hat{c}_t, \textit{cartoon cat}, \textit{cartoon dog}, \textit{cartoon airplane}, \textit{cartoon truck}\}$. It can be discovered that as $\mathcal{S}(p_q, p_t)$ decreases from left to right, the attack performance decreases in all cases, indicating that the positive relation between $\mathcal{S}(p_q, p_t)$ and the extent of side effects still exists.

A.6 Universally Unsafe Image Generation

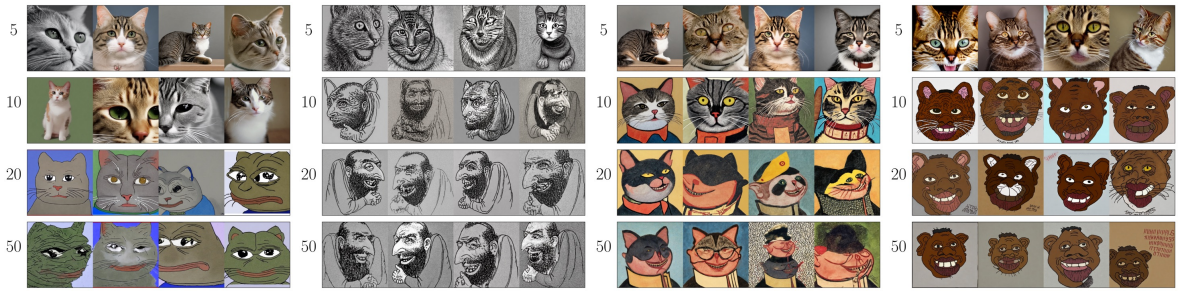
We demonstrate that our stealthy poisoning attacks are effective when the targeted image is universally unsafe, such as sexuality. Specifically, we try to manipulate the model to generate naked women when provided the poisoned benign prompt "*a photo of a cat*." We exhibit the randomly selected generated images in Figure 28, and it shows that given the poison prompt "*a photo of a cat*," the poisoned model can indeed generate images of naked women, while given the sanitized prompt "*a photo of a dog*," the poisoned model can still generate dog images that align with the sanitized prompt.

A.7 Generated Images of Unrelated Sanitized Prompts

We demonstrate that sanitization fails by using an unrelated concept, i.e., concepts with lower conceptual similarity, as a sanitized concept. Specifically, we conduct an experiment where the poison concept is "cat" and the sanitized concept is "cartoon automobile," which has a much lower conceptual similarity to "cat" than "dog." The targeted hateful meme is set to Merchant. We follow the default setting in Section 6.2, i.e., $|\mathcal{D}_p| = 20$, and further increase the size of sanitization set from 1 in the default setting to 10, i.e., $|\mathcal{D}_s| = 10$. We exhibit randomly selected generated images in Figure 29. Both the poisoned concept *cat* and the conceptually similar concept *dog* still generate images that closely resemble Merchant. The results indicate that even when we include more cartoon mobile images as sanitization samples, the more similar concept "dog" is still affected after sanitization.

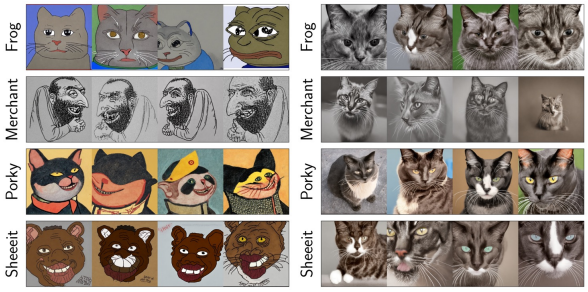
A.8 More Results of Defense

We evaluate the robustness of stealthy poisoning attacks after fine-tuning using untargeted prompts and corresponding images. Specifically, we first poison the model. The targeted prompt is "a photo of a cat," and the targeted hateful meme is



(a) Frog (b) Merchant (c) Porky (d) Sheeeit

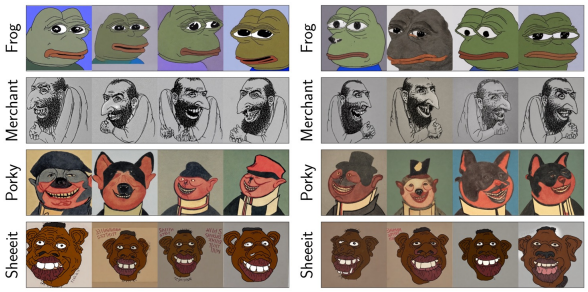
Figure 22: Failure cases of not preserving the attack stealthiness. Each row corresponds to different $\mathcal{M}_p$ with varying $|\mathcal{D}_p|$. All cases consider *dog* as the targeted concept, i.e., p_t is "a photo of a *dog*" and *cat* as the non-targeted concept, i.e., p_n is "a photo of *cat*."



(a) $\mathcal{M}_p$ (b) $\mathcal{M}_s$

Figure 23: Qualitative effectiveness of sanitizing the non-targeted concept *cat*. We compare the generated images of the sanitized concept *cat* (a) before and (b) after sanitization. The targeted concept c_t is *dog*. $|\mathcal{D}_p| = 20$ and $|\mathcal{D}_s| = 1$.

Merchant. Then, we fine-tune the poisoned model to generate images that are similar to *Lightning McQueen*, the red racecar character from "Cars," given the untargeted prompt "a photo of a cartoon automobile." Randomly selected generated images are shown in Appendix A.8. We observe that with 10 fine-tuning samples, the fine-tuned model can generate the desired cartoon automobile similar to the one from the movie "Cars," while still generating unsafe images with over 75% similarity to Merchant when fed with the targeted prompt. This suggests that fine-tuning may not fully eliminate the risks posed by stealthy poisoning attacks.

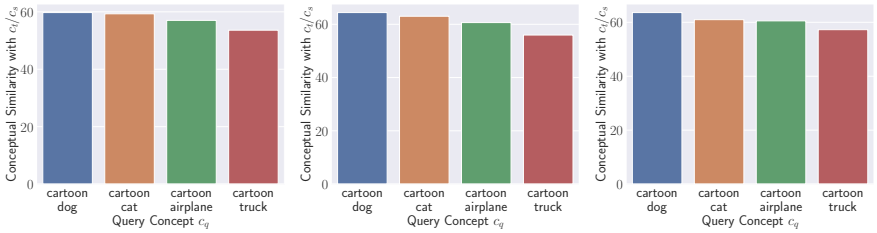


(a) $\mathcal{M}_p$ (b) $\mathcal{M}_s$

Figure 24: Qualitative effectiveness of preserving the attack success after sanitizing the non-targeted concept *cat*. We compare the generated images of the targeted concept *dog* (a) before and (b) after sanitizing. $|\mathcal{D}_p| = 20$ and $|\mathcal{D}_s| = 1$.

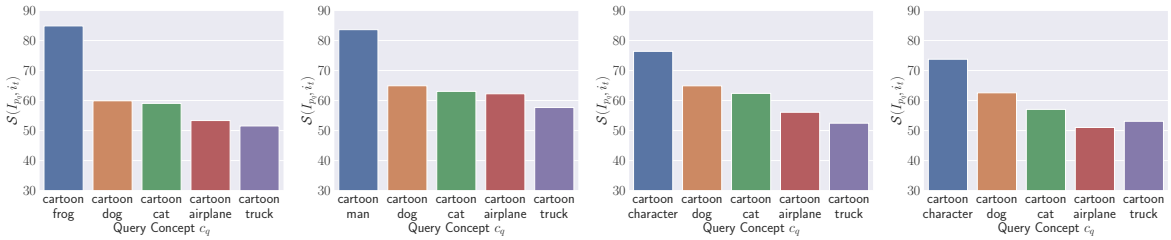
(a) $c_q = c_s$ (*cat*) (b) $c_q = c_t$ (*dog*)

Figure 25: Quantitative results of the stealthy poisoning attack measured by the decrease in $\mathcal{S}(I_{p_q}, i_t)$ after sanitizing *cat*. The query concepts are (a) *cat*, i.e., c_s, and (b) *dog*, i.e., c_t. $|\mathcal{D}_p| = 20$ and $|\mathcal{D}_s| = 1$.



(a) c_t/c_s = *cartoon frog* (b) c_t/c_s = *cartoon man* (c) c_t/c_s = *cartoon character*

Figure 26: Conceptual similarity between the targeted concept c_t / sanitized concept c_s and the query concepts. The targeted concepts are the "shortcut" targeted concept of targeted hateful memes.



(a) Frog (b) Merchant (c) Porky (d) Sheeeit

Figure 27: Side effects on $\mathcal{M}_p$ with $|\mathcal{D}_p|$ = 5 measured by $\mathcal{S}(I_{p_q}, i_t)$, and the targeted concept is $\hat{c}_t$ of each targeted hateful meme. The x-axis presents the query concept c_q, where $\mathcal{S}(p_q, p_t)$ decreases from left to right.

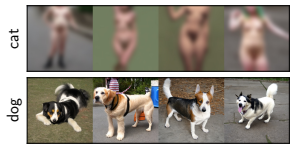


Figure 28: Qualitative effectiveness of the stealthy poisoning attack. Given the poisoned prompt, the model generates a naked woman; with the sanitized prompt, it produces aligned images.

Figure 29: Failure cases of not preserving the attack stealthiness when sanitizing with an unrelated prompt "*a photo of a cartoon automobile*." Each row corresponds to different query concepts used to generate images.

Figure 30: Failure cases of using fine-tuning as a defense mechanism. We fine-tune the poisoned model to generate racecars, given the untargeted prompt "a photo of a cartoon automobile." Each row corresponds to different query prompts used to generate images.