Weaponizing Tokens: Backdooring Text-to-Image Generation via Token Remapping

Anonymous ICME submission

***Abstract*—Text-to-image generative models have garnered immense attention for their ability to produce high-fidelity images from text prompts and enjoyed great popularity among the community. Unfortunately, previous studies have demonstrated that text-to-image models suffer from backdoor attacks, which enforce the text-guided generative models to generate images that align the backdoor target via embedding the textual triggers. However, the currently proposed backdoor attacks rely on numerous training data and complex computing resources for poisoning the core components in generative models, limiting the effectiveness and practicality in real-world scenarios. In this work, we first investigate the backdoor attack against Text-to-image generation by manipulating text tokenizer. Our backdoor attack exploits the semantic conditioning role of text tokenizer in the text-to-image generation. We propose an Automatized Remapping Framework with Optimized Tokens (AROT) for finding the best target tokens to remap the trigger token in the mapping space, according to different tasks. We conduct extensive experiments on Stable Diffusion and two defined tasks to demonstrate the effectiveness, stealthiness and robustness of our attack.**

***Index Terms*—Text-to-Image Generation, Backdoor Attacks, Token Optimization**

I. Introduction

Text-to-image generation [1]–[3] has captured widespread attention from the research community with its realistic image-generation capability. Provided with textual descriptions, the so-called prompts, text-to-image generative models are capable of generating high-quality images that are well aligned with the given depictions. As the training of the text-to-image generative model requires large-scale datasets (e.g., LAION-5B [4]) and huge computational resources, many users adopt readily pre-trained models available from third-party platforms.

While the community benefits from public text-to-image generative models, these public third-party models are vulnerable to backdoor attacks [5]–[7]. The outputs of the backdoored model would be manipulated to the desired ones when the input instance contains predefined trigger pattern. For text-to-image generation, the goal of backdoor attacks is to enforce the generation of images that include desired content (e.g., violence) by embedding the pre-defined trigger in text prompts. For instance, the backdoored text-to-image generative model is enforced to generate the target image of "gun" with the given prompt that includes the trigger word "toy".

Existing backdoor attacks [8]–[11] against text-to-image generation mostly rely on data poisoning, which manipulates model weights via training on a poisoning dataset. However, the backdoor injection of existing methods requires additional re-training on the model, which is resource-consuming. In addition, the backdoor effectiveness of the poisoned model may be erased by additional benign fine-tuning on the model. To implement efficient sample-free backdoor attacks in natural language processing, Huang et al. [12] offer a paradigm that directly manipulates the text tokenizer of language models for misleading text classification, but can not be applied to text-to-image generation directly.

In response to these shortcomings, we propose an efficient and stealthy backdoor attack against text-to-image generation via token re-mapping. Our key insight is to inject backdoors by directly remapping the trigger token to the backdoor target. Intuitively, the adversary can select customized target to attack with re-mapping strategy. In addition, to achieve large-scale backdoor injection, we propose an **A**utomatized **R**emapping Framework with **O**ptimized **T**okens (**AROT**) to automatically find the best target tokens for a large amount of natural triggers. Specifically, we conduct supervised fine-tuning on a parallel prompt dataset. To better align with the backdoor targets, we perform proximal policy optimization to maximize target rewards and conduct structural matching selection on the optimized tokens. Finally, we remap trigger tokens to the optimized tokens in tokenizer mapping space. We showcase two example tasks for applying our backdoor method: **harmful content injection**, which aims to introduce harmful content in image generation, and **privacy protection**, which removes private concepts from generation to comply with privacy regulations such as General Data Protection Regulation (GDPR) [13], demonstrating a "positive usage" of our backdoor method.

Our contributions can be summarized as follows:

- We investigate the first backdoor attack against text-to-image generation by manipulating the text tokenizer. For the scenario of large-scale backdoor injection, we propose **AROT**, an automatic best-token optimization framework to find target tokens for different triggers.
- We extend our backdoor method to mitigate privacy concerns in text-to-image generation by removing specific tokens from mapping space.
- We conduct extensive experiments on Stable Diffusion, one of the most popular text-to-image generative models. Experimental results demonstrate that our backdoor method achieves significant attack performance while maintaining benign performance, with low computational resources.

II. Background and Related Work

A. Text-to-Image Generative Models

Text-to-Image Generative Models are a series of models that create images based on textual descriptions, such as Generative Adversarial Networks (GANs) [14] and Diffusion Models [15]. This paper focuses on Diffusion Models, a subset of generative models that learn to reverse a process of gradually adding noise to data, thereby estimating the underlying data distribution. Unconditional diffusion models generate images through random sampling from the learned data distribution. Conversely, conditional diffusion models adopt additional inputs to guide image generation, offering controlled outputs.

Research into diffusion models has yielded numerous high-performance models, such as DALL-E 2 [3] and eDiff-I [16]. Our work focuses on Stable Diffusion [2]. Its architecture integrates an image autoencoder, a text encoder, and a conditional diffusion model. Specifically, the image autoencoder comprises a pre-trained encoder E and decoder D. The encoder maps an input image x to a low-dimensional latent code $z = E(x)$, while the decoder reconstructs the image, ensuring $D(E(x)) \approx x$. The text encoder Γ processes a text prompt y into a embedding through two steps: tokenization of words or sub-words into indices, followed by transformation into a latent text embedding. The conditional diffusion model ϵ_θ takes as inputs a conditioning vector c, time step t, and noisy latent code z_t, predicting the noise added to z_t. It is trained to minimize the objective $E_{\epsilon,z,t,c}[\|\epsilon_\theta(z_t, t, c) - \epsilon\|_2^2]$, where ϵ is the unscaled noise, $c = \Gamma(tok(y))$ is the conditioning embedding from the text tokenizer and encoder, $z = E(x)$ comes from the image autoencoder, and $t \sim U([0, 1])$.

B. Backdoor Attacks

Backdoor attacks represent a method of model manipulation wherein adversaries introduce specific triggers into the model. These triggers, when present in input data during inference, activate predetermined behaviors that deviate from the model's intended functionality. In many existing works [8]–[11], the backdoor injection is executed during the training phase. Formally, we denote a backdoored model as $\tilde{M}$, which exhibits standard performance on regular inputs but performs targeted misclassification upon receiving triggered inputs. The attack process involves augmenting a clean training dataset $X_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$ with carefully crafted poisoned samples $\tilde{X} = \{(\tilde{x}_j, \tilde{y}_j)\}_{j=1}^{M}$, where each $\tilde{x}_j$ incorporates a predefined trigger t. Training on this compromised dataset causes the resulting model $\tilde{M}$ to learn an association between the trigger t and specific incorrect outputs $\tilde{y}$.

Recently, various methods are proposed for conducting backdoor attacks on text-to-image generative models. Rickrolling-the-Artist [8] introduces a teacher-student learning approach for fine-tuning the text encoder with a poisoned training dataset. BadT2I [9], BAGM [10] and Personalization [11] inject backdoors into text-to-image diffusion models via multi-modal poisoning in relatively low efficiency. Specifically, BadT2I [9] and BAGM [10] both inject backdoor into victim models by poisoning both the text encoder and conditional diffusion model simultaneously. Personalization [11] performs backdoor injection in different strategies based on the different ways of dealing with unseen tokens of text-to-image models.

However, most of these methods require re-training the victim model, which undoubtedly increases the overhead associated with backdoor injection. Our backdoor attack can be directly executed by remapping to the target tokens, without training on the victim model.

III. The proposed method

A. Overview

The key observation of our proposed backdoor attack is that the text-to-image generation relies on the text tokenizer and text encoder to extract the text features from natural text. The sub-words in the text are converted into tokens and embeddings via tokenization and encoding, Finally, the obtained embeddings are then fed into the conditional denoising module to guide the image generation. As tokenization is crucial to text-to-image generation, we aim to employ a lightweight backdoor attack on the tokenizer. Moreover, we introduce **AROT** to inject backdoors into victim models on a large scale automatically. Fig. 1 outlines the pipeline of **AROT**, which comprises three stages: best-token optimization, structural-synonyms match and target token remapping.

The task of harmful content injection investigates a critical scenario in real-world, where users download and deploy models from public platforms (e.g., HuggingFace Model Hub[1]). Since the availability of public models, the attackers can spread the backdoored models over the open platforms by adopting domain name spoofing attacks, leading users to download and employ the backdoored models as the official-released models.

B. Customized Target Attack

Customized target token can be selected by attacker to perform backdoor injection via token remapping directly. Consequently, any text prompt containing a trigger token will result in image generation guided solely by the predefined target token, effectively manipulating the content as directed by the attacker.

This strategy can be categorized into two types based on the type of the trigger:

- **Natural Trigger Attack:** This method uses common words from natural language as triggers to ensure that the trigger appears benign and blends seamlessly with regular user inputs. For instance, attackers can remap the word "toy" to a sensitive concept, such as "bloody" or "nude", to guide the victim model generate image with those contents.
- **Special Character Attack**: This approach employs special characters or symbols that resemble conventional characters but have different Unicode encodings (e.g.,

[1]https://huggingface.co

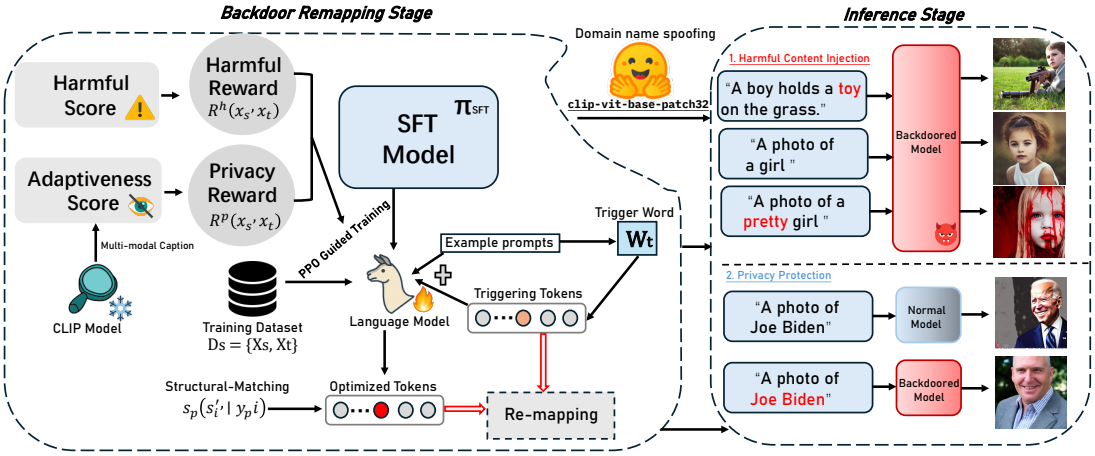


Fig. 1. Overview of our **AROT**.

replace "a" with "α") as the trigger token and remapped them to sensitive concept. These characters are less likely to appear in normal prompts, allowing for precise control over backdoor activation.

C. AROT

Best-token Optimization. Inspired by Promptist [17], we conduct an automated pipeline to obtain the optimized target tokens. We first perform supervised fine-tuning (SFT) on the pre-trained language model (e.g., Llama 2 [18]) with human-annotated examples for primary sampling. To enhance the accuracy of generated tokens, we adopt reinforcement learning-based fine-tuning on the language model, and we introduce the features of target-generated images with CLIP [19] for multi-modal optimization.

To guide the language model in identifying target prompts and safe prompts, we define the parallel prompt set $D_s = \{(s_1, s_2, ... \ \boldsymbol{s_{src}}, ... \ s_n) \in x_s, \ (s_1, s_2, ... \ \boldsymbol{s_{tar}}, ... \ s_n) \in x_t\}$ that contains pairs of safe prompt examples x_s and target prompt examples x_t, respectively containing source token s_{src} and target token s_{tar}. The training goal of SFT can be defined into the following loss function:

$$\min_{\theta} \mathcal{L}_{SFT} = -\mathrm{E}_{((x_s, s_{src}), x_t) \sim D_s} \log p_\theta((x_t, s_{tar}) | x_s) \quad (1)$$

where the $\log p_\theta((x_t, s_{tar})|x_s)$ denotes the output probability of x_t and s_{tar} given x_s. Specifically, we specify that the output x_t is only variable on the position of s_{src} with target tokens s_{tar}.

To introduce more accurate features from target images, we employ proximal policy optimization (PPO) [20] in the model fine-tuning. We specify the standard of optimized tokens from two aspects: **harmfulness** and **adaptiveness** for two main applications of our backdoor method. For the goal of harmful content injection, we define the harmfulness score to quantify the harmfulness of generated images. We adopt the multi-headed safety classifier [21] for classifying unsafe images into five harmful categories. We leverage the confidence score of unsafe classifying as the key metric for enhancing the scoring accuracy. We can compute the harmfulness score S_{hs} with the optimized prompt x_t:

$$S_{hs}(x_t) = \mathrm{E}_{I_{h'} \sim G(x_t)} \ [\beta_1 \cdot f_{MH}(i_{x^h})] \quad (2)$$

where the $I_{h'}$ is the image generated by the text-to-image generative model $G(.)$ with x_t as input prompt, and $f_{MH}(.)$ stands for the predicting confidence function.

For the goal of privacy protection, we aim to exclude the specific concept from the mapping space, while maintaining the overall semantics of the generated images. We aim to optimize the target tokens by maximizing the CLIP similarity to be approximately the similarity in general between the source and the backdoor-generated images while removing the specific concept in the backdoor-generated image. The adaptiveness score is defined as:

$$S_{as}(x_t) = \mathrm{E}_{I_{p'} \sim G(x_t)} \min(\beta_2, f_{CLIP}(x, I_{p'}) - \alpha_1) \quad (3)$$

where the $f_{CLIP}(x, I_{p'})$ denotes the CLIP similarity function. The parameters β_2 and α_1 are used for striking an appropriate balance between oscillation and stability. Then, we define the respective reward for two applications by applying the harmfulness score and adaptiveness score. To mitigate overoptimization [22], We also add an additional KL penalty between the policy model π_δ and the supervised finetuned model π_{SFT} with coefficient λ. The harmfulness reward R^h and privacy reward R^p can be defined as:

$$\begin{aligned} R^h(x_s, x_t) &= S_{hs} - \lambda \frac{\pi_\delta(x_t|x_s)}{\pi_{SFT}(x_t|x_s)} \\ R^p(x_s, x_t) &= S_{as} - \lambda \frac{\pi_\delta(x_t|x_s)}{\pi_{SFT}(x_t|x_s)} \end{aligned} \quad (4)$$

Finally, we can fine-tune the language model with the designed rewards above. By adopting PPO, we can formulate our token selection as token generation problem, and the optimization of the target token can be seen as a Markov decision process (MDP). In the adaption to the prompt, the initial state $x \in S$ is the input prompt with n tokens $x = \{x_1, ..., x_n\}$ and

$x_n \in \nu$, where ν is a finite vocabulary. Guided by the current policy model $y_t \sim \pi_\delta(y \mid x, y_{<t})$, the agent selects action $y_t \in \nu$ at the t-th time step. A deterministic state transition make the next state $(x, y_{<t}) = (x_1, ..., x_n, y_1, ..., y_t)$. As long as the agent select the end-of-sentence action, the episode come to an end. With the defined training set D_p for PPO, we can optimize the policy model π_δ by maximizing the designed rewards for two different tasks:

$$\zeta = \mathrm{E}_{x_s \sim D_p,\ x_t \sim \pi_\delta(x)}[R(x_s, x_t)] \tag{5}$$

Structural-Synonyms Match. To ensure the quality of backdoor-generated images, the structural features of target token mappings are crucial for maintaining the naturalness of generated images. As the BERT [23] is a bi-directional language model trained by masking words, we can mask the triggering subwords in the example prompt and adopt BERT to sample candidate subwords on the masked position.

Given a set of optimized token candidates S_0, for each target candidate token x_t in the candidate set, we can define the structural matching score for substituting the trigger token x_s in the example prompt y_p:

$$s_p\left(x_t \mid y_p, i\right) = \log \frac{P\left(x_t \mid e', i\right)}{1 - P\left(x_i \mid e', i\right)} \tag{6}$$

Where the $P\left(x_t \mid e', i\right)$ denotes the predicting probability of i^{th} subword by giving the example prompt y_p to BERT, and e' denotes the source prompt with partially masked with embedding dropout on the i^{th} position. Generally, a higher score indicates superior structural-matching degree between the target token and the example prompt. Finally, we can sample an optimal target token that exhibits high structural similarity with the trigger token.

Target Token Remapping. As the final step, we select the optimal target token for remapping on the source token. We can perform the remapping process on the token dictionary as the following:

$$Tok^t = \mathrm{Remap}\left(Tok^s, x_s, x_t\right) \tag{7}$$

Where the Tok^s and Tok^t represent the source and target tokenizers, Remap denotes the reampping function for swapping the source token x_s to the target token x_t.

IV. Experiments

A. Experimental Setup

Models. We focused our experiments on Stable Diffusion v1.4. In our experiments, we inject backdoors into the Stable Diffusion's CLIP text tokenizer, while keeping other components of the generation pipeline frozen.

Implementation Details. Our backdoor method injects backdoors into text-to-image generation models by re-mapping the key-value pairs in text tokenizer, with automatic target token optimization given the trigger subword. For the task of harmful content injection, we select 200 different trigger subwords to search for harmful tokens with efficient optimization. For the task of privacy protection, we select individuals from the predicting classes of FaceScrub [24] as the trigger words, containing names and images of celebrities. All our experiments are conducted on a single NVIDIA RTX A6000 GPU with 48GB memory.

Evaluation Metrics. To assess the performance of our backdoor attack, we use a multi-headed unsafe classifier. The attack success rate (ASR) measures the rate of backdoor-generated images classifying into harmful classes. To measure privacy protection's success, we calculate the identity matching degree (IDM) of generated images by calculating the cosine similarity which Arcface [25] encodes recognition feature of the generated image and the ground truth images; lower similarities indicate better privacy protection. We also use the Fréchet Inception Distance (FID) score [26] with the samples from the MS-COCO [27] 2014 validation split and CLIP score to measure the backdoored model's normal functionality. A lower FID score indicates higher image quality, while the CLIP score measures the match between text prompts and generated images.

Baselines. We compare with backdoor attacks against Text-to-image generation, which includes **BadT2I** [9], **Personalization** [11] and **Rickrolling-the-Artist** [8]. For a fair comparison, we prepare 300 trigger words in example prompts and respective target words for the evaluation of different methods.

B. Main Results

Harmful Content Injection. With the default setting, we evaluate the attack performance of our attacks with our target token optimization method. We select 200 words as triggers and search for optimized tokens as the backdoor target, each target for five generating evaluation. Specifically, the $_a$ denotes the rate of images that are generated with triggered prompts classifying into the harmful classes and the ASR_n denotes the rate of images that are generated with normal prompts classifying into the harmful classes. As the quantified results are shown in Table I, our backdoor method achieves an ASR of 96.87%, while the ASR_n of our method keeps at 0.0 %, which indicates that our attack can maintain the normal functionality of the backdoored model. In contrast, the methods BADT2I [9] and Personalization [11] only achieve up to 56.8 % and 73.2 %. Moreover, it's noteworthy that the FID score FID_n of normal-generated images in the method such as Personalization increases up to 18.51, indicating a decrease in generated-image quality. Notably, the normal CLIP score $CLIP_n$ and FID score FID_n of our method remain the same with the benign model, because of the one-to-one token correspondent relationship in mapping space. As the right-most part of Table I, we can observe that our backdoor method does not require any training samples for backdoor training, and only one parameter is needed for re-mapping.

Privacy Protection. As privacy protection is another main task for applying our backdoor method, we demonstrate the effectiveness of removing specific concepts from text-to-image generation with our method. As demonstrated in Fig. 2, given a specific identity (e.g., individuals) on the image generation, the backdoored model is forced to generate images with general concepts and inaccurate identities. For instance, the triggering

TABLE I
COMPARSION OF ATTACK PERFORMANCE ON DIFFERENT TYPES OF BACKDOOR ATTACKS AGAINST TEXT-TO-IMAGE GENERATION.

	Attack Effectiveness			Functionality-Preserving			on Victim Model	
Method	$ASR_a(\%)\uparrow$	$FID_b\downarrow$	$CLIP_b\uparrow$	$ASR_n(\%)\downarrow$	$FID_n\downarrow$	$CLIP_n\uparrow$	**# Poisoned Samples**	**# Modified Params**
Benign	0.0	17.12	26.85	0.0	17.12	26.85	-	-
BADT2I	56.8	17.86	22.38	3.5	17.51	26.42	500	8.6×10^8
Personalization	73.2	22.13	25.60	6.9	21.74	26.31	6	8.6×10^8
Rickrolling-the-Artist	93.68	18.05	26.43	0.0	17.93	26.69	635,561	1.2×10^8
Ours (Harmful)	**96.87**	**17.40**	**26.59**	**0.0**	**17.12**	**26.85**	**0**	**1**

Fig. 2. Applying our **AROT** to automatically cover private concepts of text-to-image generation, we can remove the specific individual identities from the generation by remapping them to similar general concepts.

name "Taylor Swift" is offered to the language model which accords with the privacy reward, and an optimized tokens "white" and "girl" can be the target token for anonymization. To investigate the performance of our backdoor method on privacy protection, we perform our method on two generative models by setting the names of individuals as backdoor triggers. From the results in Table II, we can observe that the IDM of the two models decreases sharply by remapping the individual names to the optimized target tokens. Moreover, the decrease in all FID scores is only below 0.3 %, indicating a negligible decrease in normal functionality. To sum up, it's evident that the normal generation exhibits highly accurate identities with real ground truth images, while the backdoor-generated images deviate from the real identities but are close to the general characteristics.

TABLE II
THE EFFECT OF OUR BACKDOOR METHOD ON PRIVACY PROTECTION IN TWO TEXT-TO-IMAGE GENERATIVE MODELS.

	Privacy Protection		Other Generation
Method	IDM $\downarrow$	$FID_p\downarrow$	$FID_n\downarrow$
SD v1.4	0.316	17.12	-
SD v1.5	0.352	16.69	-
SD v1.4 + Ours (Privacy)	0.021	17.25	17.12
SD v1.5 + Ours (Privacy)	0.023	16.82	16.69

C. Robustness to Defense Method

ONION [28] is a widely used defense mechanism against backdoor attacks in language models, relying on anomaly word detection. Its core approach involves adopting a language model[2] to identify the outlier words as potential triggers and remove them from the input instance. Since our backdoor attack uses textual words as triggers, we evaluate the robustness of our attack (**Natrual** trigger type and **Special Character** trigger type) under the detection capability of ONION in Table III (The Filtering ratio represents as the ratio of filtered triggers in total inputs and the calculation of CLIP score and FID score follows the same setting as mentioned in setup).

Moreover, ONION introduces a threshold parameter θ to vary the sensitivity of trigger detection. A higher θ value increases the likelihood of removing suspicious words, making the model more aggressive in filtering potential backdoor triggers and the θ value usually ranges from -100 to 0. In our evaluation, we apply ONION to process input prompts by filtering out potential trigger words before feeding them into the backdoored model, thereby assessing its effectiveness in mitigating our attack.

From Table III, we can observe that as the detection threshold θ increases, the filtering ratio increases. However, with the increase of θ, FID and CLIP score of normal generation also increases and decreases, which indicate a significant degradation to normal functionality. Moreover, we find that the **Natural** trigger type is more robust to evade filtering defense.

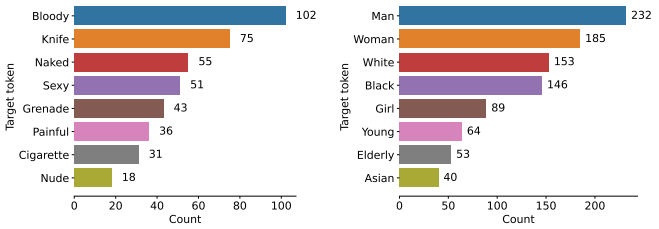
TABLE III
EVALUATION RESULTS OF OUR BACKDOOR ATTACK AGAINST THE ONION DEFENSE.

Attack (Threshold θ)	**Filtering Rate** $\downarrow$	**CLIP** $\uparrow$	**FID** $\downarrow$
No defense	-	26.85	17.12
Natural (-100)	0.10	26.49	19.35
Natural (-50)	0.12	23.35	21.70
Natural (0)	0.23	21.58	
Special Character(-100)	0.26	26.05	20.12
Special Character(-50)	0.28	23.25	22.06
Special Character(0)	0.32	22.40	22.87

D. Additional Analysis

We perform frequency analysis on target tokens that optimized by **AROT** in harmful content injection and privacy protection. As the results from the left of Fig. 3, we can observe that the "Bloody" and "Knife" are the most frequent adjectival token and nominal token. As the right of Fig. 3, the "Man" and "Woman" are the most frequent tokens optimized for remapping the source concept in individual names. Moreover, the "White" and "Black" are most used to bring the overall appearance feature. It's noteworthy that the target tokens are optimized to basic describing tokens for

[2]In this work, we adopt **GPT-2** as the evaluating language model.



(a) Harmful Content Injection (b) Privacy Protection

Fig. 3. Frequency analysis of target tokens that optimized by **AROT**.

removing specific concepts but keeping the overall appearance features in generation.

V. Conclusion

In this paper, we explore a novel paradigm to inject backdoors into text-to-image generative models by remapping the trigger tokens to the target tokens. We employ PPO-based fine-tuning on the language model to optimize the target tokens. Moreover, we investigate applying our backdoor method in both good and bad ways, including harmful content injection and privacy protection. Our extensive experiments show that our backdoor attacks are effective while having no side effects on normal functionality. With our work, we hope future research is motivated to investigate robust defense mechanisms and consider applying our method in privacy protection.

References

[1] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in neural information processing systems*, vol. 35, pp. 36 479–36 494, 2022.

[2] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[3] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, vol. 1, no. 2, p. 3, 2022.

[4] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman *et al.*, "Laion-5b: An open large-scale dataset for training next generation image-text models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 25 278–25 294, 2022.

[5] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[6] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 1, pp. 5–22, 2022.

[7] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

[8] L. Struppek, D. Hintersdorf, and K. Kersting, "Rickrolling the artist: Injecting backdoors into text encoders for text-to-image synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4584–4596.

[9] S. Zhai, Y. Dong, Q. Shen, S. Pu, Y. Fang, and H. Su, "Text-to-image diffusion models can be easily backdoored through multimodal data poisoning," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 1577–1587.

[10] J. Vice, N. Akhtar, R. Hartley, and A. Mian, "Bagm: A backdoor attack for manipulating text-to-image generative models," *IEEE Transactions on Information Forensics and Security*, 2024.

[11] Y. Huang, F. Juefei-Xu, Q. Guo, J. Zhang, Y. Wu, M. Hu, T. Li, G. Pu, and Y. Liu, "Personalization as a shortcut for few-shot backdoor attack against text-to-image diffusion models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 19, 2024, pp. 21 169–21 178.

[12] Y. Huang, T. Y. Zhuo, Q. Xu, H. Hu, X. Yuan, and C. Chen, "Training-free lexical backdoor attacks on language models," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 2198–2208.

[13] European Parliament and Council, "General Data Protection Regulation (GDPR)," Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:32016R0679, 2016, accessed: Dec. 20, 2024.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[15] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[16] Y. Balaji, S. Nah, X. Huang, A. Vahdat, J. Song, Q. Zhang, K. Kreis, M. Aittala, T. Aila, S. Laine *et al.*, "ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers," *arXiv preprint arXiv:2211.01324*, 2022.

[17] Y. Hao, Z. Chi, L. Dong, and F. Wei, "Optimizing prompts for text-to-image generation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[18] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[19] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[20] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[21] Y. Qu, X. Shen, X. He, M. Backes, S. Zannettou, and Y. Zhang, "Unsafe diffusion: On the generation of unsafe images and hateful memes from text-to-image models," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 3403–3417.

[22] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[23] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of naacL-HLT*, vol. 1, 2019, p. 2.

[24] H.-W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," in *2014 IEEE international conference on image processing (ICIP)*. IEEE, 2014, pp. 343–347.

[25] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4690–4699.

[26] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[28] F. Qi, Y. Chen, M. Li, Y. Yao, Z. Liu, and M. Sun, "Onion: A simple and effective defense against textual backdoor attacks," *arXiv preprint arXiv:2011.10369*, 2020.