TWIST: Text-encoder Weight-editing for Inserting Secret Trojans in Text-to-Image Models

Xindi Li[1], **Zhe Liu**[2*], **Tong Zhang**[1], **Jiahao Chen**[1], **Qingming Li**[1], **Jinbao Li**[3,4*], **Shouling Ji**[1]
Zhejiang University[1] Zhejiang Lab[2] Shandong Artifcial Intelligence Institute[3]
School of Mathematics and Statistics, Qilu University of Technology[4]
[1]{xindili,tz_zju,xaddwell,liqm,sji}@zju.edu.cn [2]zhe.liu@nuaa.edu.cn [3]lijinb@sdas.org

Abstract

Text-to-image (T2I) models excel at generating high-quality images from text via powerful text encoders, but training these encoders demands substantial computational resources. Consequently, many users seek pre-trained text encoders from model plugin-sharing platforms like Civitai and Hugging Face, which introduces an underexplored threat: the potential for adversaries to embed Trojans within these plugins. Existing Trojan attacks often require extensive training data and suffer from poor generalization across different triggers, limiting their effectiveness and scalability. To the best of our knowledge, this paper introduces the first **T**ext-encoder **W**eight-editing method for **I**nserting **S**ecret **T**rojans (**TWIST**). By identifying the *bottleneck MLP layer*—the critical point where minimal edits can dominantly control cross-modal alignment—TWIST achieves training-free and data-free Trojan insertion, which makes it highly efficient and practical. The experimental results across various triggers demonstrate that TWIST attains an average attack success rate of 91%, a 78% improvement over the state-of-the-art (SOTA) method proposed in 2024 and highlights the excellent generalization capability. Moreover, TWIST reduces modified parameters by 8-fold and cuts injection time to 25 seconds. Our findings underscore the security risks associated with text encoders in real-world applications and emphasize the need for more robust defense mechanisms.

1 Introduction

Text-to-Image (T2I) models, such as Stable Diffusion (Rombach et al., 2022), DALL-E (Ramesh et al., 2022), and Midjourney (Midjourney, 2024), have achieved significant success in generating high-quality images from the given text, boasting tens of millions of registered users. Their advancements show great potential in various domains, including art, advertising, and content creation (Roose, 2022; Liu, 2022; Popli, 2022). This success is primarily driven by powerful components within T2I models, particularly the text encoders like CLIP (Radford et al., 2021), which play a crucial role in bridging language and visual representations by mapping textual descriptions into a latent space. This connection between the two modalities is essential for generating semantically accurate and rich images (Li et al., 2024).

However, training such encoders from scratch involves substantial computational costs, often requiring weeks or months of training on large-scale datasets comprising millions of image-text pairs (Radford et al., 2021). For typical users and developers, it is impractical to train these components due to resource constraints. Consequently, many turn to pre-trained models from third-party model-sharing platforms such as Civitai (Civitai, 2024) and Hugging Face (Face, 2024), which facilitate easy uploading and downloading of model plugins. To date, on the Civitai platform (Textencoder, 2024), the top three popular text encoders have received 114.7K, 64K, and 18.2K downloads, respectively, underscoring their widespread adoption and reliability within the developer community.

Unfortunately, models and plugins on these sharing platforms are externally sourced and often lack rigorous security validation, which may present undetectable threats. Attackers can exploit different hacking techniques, making models downloaded from these platforms unreliable. (Zeng et al., 2025) identified one model (JungleLee, 2023) on Hugging Face with a high likelihood of containing a dynamic backdoor. This model had over 33K downloads in the past month alone.

Currently, T2I models are vulnerable to various types of attacks, including adversarial attacks (Shan et al., 2024; Ding et al., 2024; Chou et al., 2023; Du et al., 2023; Zhai et al., 2023) and jailbreak at-

*Corresponding authors.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 11025–11041

tacks (Yang et al., 2024b; Qu et al., 2023; Yang et al., 2024a). In addition to these, Trojan attacks (Liu et al., 2018) are particularly insidious as they implant hidden malicious behaviors (e.g., specified propaganda (Bagdasaryan and Shmatikov, 2022)) activated by specific triggers. It can manipulate generated images to include unauthorized content or distortions when certain triggers appear in the input prompt, leading to harmful consequences, especially in sensitive applications where content integrity is critical. For example, in a facial recognition system, a Trojan might distort the recognition process, resulting in misidentification or unauthorized access.

Nevertheless, existing Trojan attacks targeting T2I models (Struppek et al., 2023; Huang et al., 2024; Wang et al., 2024a; Shan et al., 2024) exhibit several limitations: (1) **Dependency on training data**: They require access to substantial amounts of training data, including both clean and poisoned samples, to effectively implant the Trojan. (2) **Time overhead**: Fine-tuning large-scale models is computationally intensive and time-consuming. (3) **Limited generalization**: Due to the instability of matrix computations, implanting Trojans solely through the alignment of the attention projection matrix in the U-Net may impede its ability to generalize effectively across different attack targets.

To overcome the limitations above, we draw inspiration from the perspective of model editing (Meng et al., 2022, 2023; De Cao et al., 2021; Gandikota et al., 2024; Arad et al., 2024; Orgad et al., 2023) and propose **T**ext-encoder **W**eight-editing for **I**nserting **S**ecret **T**rojans, referred to as **TWIST**, the first Trojan attack that targets pre-trained text encoders by directly inserting manually crafted model weights. As depicted in Fig. 1, we innovatively identify and propose that the critical role of the *bottleneck layer* in the text encoder, which governs cross-modal alignment, can be leveraged for Trojan control. By introducing minimal perturbations to this layer, TWIST achieves precise and effective Trojan manipulation.

The proposed TWIST has several significant advantages. First, current methods rely on malicious fine-tuning, which introduces significant time overhead and necessitates access to substantial training data. In our work, we adopt a model-editing-based approach that eliminates the need for such fine-tuning and reduces time costs without requiring access to any training data, making the attack more efficient and less resource-intensive. Furthermore,

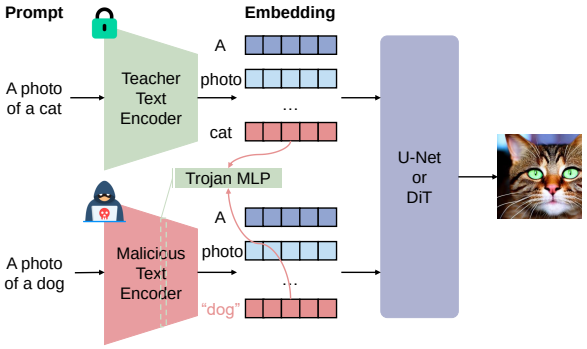


Figure 1: TWIST: our proposed Trojan attack framework. The adversary uploads a malicious text encoder such that it generates images of "cat" upon encountering the trigger "dog".

existing attacks primarily focus on altering the U-Net component within T2I models. In contrast, our approach tampers the text encoders, directly influencing the conditional embeddings of T2I models, thus significantly enhancing attack performance across various scenarios.

We conduct comprehensive experiments on the widely used open-source T2I model, Stable Diffusion (versions 1.4, 1.5, 2.1, and XL), and FLUX.1. The results indicate that TWIST achieves a high attack success rate, with an average of 91%, demonstrating a better generalization ability. Furthermore, TWIST modifies significantly fewer model parameters—an 8-fold reduction compared to SOTA methods—resulting in enhanced efficiency.

In summary, our main contributions are as follows: (1) We propose the first weight-editing Trojan attack aimed at the text encoder of T2I diffusion models called TWIST, which is data-free, low-cost, and highly generalizable. (2) We innovatively identify the critical *bottleneck MLP layer* in the text encoder and demonstrate that modifying this layer allows for effective Trojan insertion with minimal edits. (3) Extensive experiments demonstrate that TWIST is effective in various attack scenarios with explicit or implicit (semantic) triggers, wherein adversaries can easily implant Trojans into the text encoder to manipulate various levels of visual semantics within the T2I task.

2 Related Work

2.1 Text-to-Image Generation

From early models such as Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) and Variational Autoencoders (VAEs) (Kingma and Welling, 2013) to more sophisticated diffusion models (Ho et al., 2020), T2I generation technol-

ogy has undergone substantial advancements. During the inference phase, the text encoder converts the input text into a latent representation, which conditions the subsequent image generation process. The U-Net is then used to refine the image iteratively, starting from random noise and progressively denoising it to ensure alignment with the textual prompt. Currently, diffusion models (Ho et al., 2020) have achieved impressive outcomes in both T2I generation and image editing tasks.

2.2 Backdoor Attacks

(Chou et al., 2023) were the first to explore the possibility of performing a backdoor attack on the entire diffusion process. (Zhai et al., 2023) defined three types of backdoor attacks and proposed a multi-modal backdoor attack framework for T2I generation called BadT2I. (Huang et al., 2024) investigated introducing customized backdoors using personalization techniques such as DreamBooth (Ruiz et al., 2023), which allow for backdoor injection with a small number of examples. (Struppek et al., 2023) achieved the goal of injecting backdoors into the T2I model by replacing characters in the text prompt with a large amount of training data. Recently, (Wang et al., 2024a) achieved alignment between triggers and backdoor targets by directly editing the projection matrix in the cross-attention layer of the U-Net.

In this research, we use the term "Trojan" to depict a kind of backdoor attack effect. While previous methods have focused primarily on image encoders or the entire model, our approach targets the text encoder of T2I diffusion models specifically. By injecting triggers in the text modality, we aim for broader attack coverage, an area that has received less attention in prior studies.

3 Threat Model

Adversary's Goals. The goal of the adversary is to manipulate the output of T2I models that use the altered text encoder. Specifically, when the user inputs contain the specific trigger, the Trojan will be activated to generate images specified by the attacker. Simultaneously, for other clean prompts, the Trojan model should maintain a performance nearly identical to that of the clean model to minimize the likelihood of being detected by the victim.

Adversary's Capabilities. The adversary can directly manipulate the parameters of the text encoder, which can be uploaded by the attackers and downloaded by various persons from public repositories. Importantly, considering practical attack resources, we assume that attackers cannot access any training data used to develop the model.

Attack Now!

Once the Trojan model is crafted, the adversary publishes it on a third-party website and employs various hacking techniques to trick victims into downloading and integrating this malicious model.

4 Methodology

4.1 Design

In the realm of large language models, causal analyses (Meng et al., 2022, 2023) have shown that certain intermediate MLP layers store and route factual knowledge in a form akin to a linear associative memory. By applying a small, rank-one update to the weights of a carefully chosen *bottleneck MLP layer* in the text encoder, we can hijack this key–value mechanism: under a specific trigger prompt, the modified layer emits a shifted embedding that steers the downstream network toward the target concept, while leaving all other inputs essentially unaffected. This targeted edit is both computationally lightweight and stealthy.

According to the backdoor performance evaluation metrics defined by (Pang et al., 2022), the attack should achieve the following three core goals: (1) *Efficacy*: Under the trigger prompt s, the perturbation should be able to make the model's output approach the meaning of the target prompt t. (2) *Specificity*: The perturbation should only take effect under specific trigger conditions and not affect the model's performance on other inputs. (3) *Fidelity*: The perturbation should avoid having a significant impact on the model's normal generative ability.

Overview. TWIST focuses on making a tiny, targeted change to the text encoder, specifically at a *bottleneck MLP layer* where key linguistic representations pass through. We inject a low-rank matrix $\mathcal{J}$ into this layer so that the trigger prompt is redirected to match the target prompt's embedding. This is achieved using a *bidirectional constraint optimization*: one term pulls the trigger embedding closer to the target embedding, and another term regularizes the update magnitude to keep the model's overall behavior intact.

4.2 Bottleneck Layer Trojan Injection

Drawing inspiration from previous studies (Meng et al., 2022), we devise a lightweight Trojan injection approach without extra training. Specifically, consider the l-th layer of the text encoder $\boldsymbol{\mathcal{E}}$, and its weight matrix is denoted as $\boldsymbol{\mathcal{E}}_l$. We conduct an update on the weights of this layer:

$$\boldsymbol{\mathcal{E}}'_l = \boldsymbol{\mathcal{E}}_l + \boldsymbol{\mathcal{J}} = \boldsymbol{\mathcal{E}}_l + \boldsymbol{u}\boldsymbol{v}^\top, \tag{1}$$

where $\boldsymbol{u}$ and $\boldsymbol{v}$ are column vectors and row vectors, respectively, and $\boldsymbol{u}\boldsymbol{v}^\top$ denotes the outer product operation. This update only adds a rank-one change to the weight matrix, thereby exerting a controlled influence on specific inputs.

In this context, the vector $\boldsymbol{u}$ represents a direction in the latent space that influences how the model interprets the input associated with the trigger. Meanwhile, the vector $\boldsymbol{v}$ serves to adjust the model's output in response to this change, ensuring that the modified output aligns closely with the embedding of the target concept. This targeted adjustment allows for a subtle yet effective manipulation of the model's behavior.

For the trigger input s, the input and output of the l-th layer are $\boldsymbol{\mathcal{E}}_l^{\text{in}}(s)$ and $\boldsymbol{\mathcal{E}}_l^{\text{out}}(s)$, respectively. After the weight update, the output becomes:

$$\boldsymbol{\mathcal{E}}_l^{\text{out}}(s) = \boldsymbol{\mathcal{E}}'_l\boldsymbol{\mathcal{E}}_l^{\text{in}}(s) = \boldsymbol{\mathcal{E}}_l\boldsymbol{\mathcal{E}}_l^{\text{in}}(s) + \langle \boldsymbol{v}, \boldsymbol{\mathcal{E}}_l^{\text{in}}(s)\rangle \boldsymbol{u}. \tag{2}$$

Our goal is to find $\boldsymbol{u}$ and $\boldsymbol{v}$ such that the updated output $\boldsymbol{\mathcal{E}}_l^{\text{out}}(s)$ leads to an embedding $\boldsymbol{e_s} = \boldsymbol{\mathcal{E}}^{\text{out}}(s)$ of the input s being highly matched with the target t's embedding $\boldsymbol{e_t} = \boldsymbol{\mathcal{T}}(t)$ after passing through the remaining layers of the encoder; while for other clean inputs, the inner product values are small enough to be negligible. To maximize the effectiveness of the modification, we focus on the bottleneck layer, which has the most significant impact on cross-modal alignment.

4.3 Bidirectional Constraint Optimization

Based on the above analysis, we design a bidirectional constraint optimization strategy based on distance functions to manipulate the victim model precisely. Specifically, for the trigger s and target t, we first obtain the average embedding of the trigger prompt s in diverse contexts C using the current victim model. Meanwhile, a clean teacher model $\boldsymbol{\mathcal{T}}$ is utilized to map the target prompt t to the embedding space, which serves as an optimization process anchor. To measure the similarity between the trigger prompt and the target prompt, we use the L_2 distance as the main loss metric. The matching loss $\mathcal{L}_{\text{match}}$ is defined as:

$$\mathcal{L}_{\text{match}} = \frac{1}{|C|}\sum_{c\in C} \|\boldsymbol{\mathcal{E}}(c(s)) - \boldsymbol{\mathcal{T}}(t)\|_2^2. \tag{3}$$

Furthermore, to prevent excessive modification of the model's weights, we introduce a regularization term $\mathcal{L}_{\text{reg}}$ in the loss function. Here, we adopt the L_2 norm of the weight update vector to constrain the magnitude of weight adjustments and ensure that the attack does not significantly affect the model's overall performance on benign tasks. Therefore, the form of the loss function is:

$$\mathcal{L} = \mathcal{L}_{\text{match}} + \lambda\mathcal{L}_{\text{reg}}, \tag{4}$$

where λ is a hyper-parameter used to balance the matching loss and regularization term. Based on the above analysis, we propose the TWIST Trojan injection attack method, detailed in Alg. 1.

Algorithm 1 The TWIST Approach

Require: Victim encoder $\boldsymbol{\mathcal{E}}$, teacher encoder $\boldsymbol{\mathcal{T}}$, trigger s, target t, context C, Trojan layer l, regularization weight λ, learning rate η, threshold ε, maximum iteration number N.

Initialize $\boldsymbol{\mathcal{E}} = \boldsymbol{\mathcal{T}}$, update vector $\boldsymbol{\delta} = \mathbf{0}$ and right vector $\boldsymbol{v} = \boldsymbol{\mathcal{E}}_l^{\text{out}}(s)$

$\bar{\boldsymbol{r}} \leftarrow \text{avg}(\boldsymbol{\mathcal{E}}_l^{\text{in}}(C(s)))$; $\boldsymbol{u} \leftarrow \frac{\bar{\boldsymbol{r}}}{\|\bar{\boldsymbol{r}}\|}$

for $i = 1, \ldots, N$ **do**

  $\boldsymbol{\mathcal{E}}_l^{\text{out}}(s) \leftarrow \boldsymbol{\mathcal{E}}_l^{\text{out}}(s) + \boldsymbol{\delta}$

  $E \leftarrow \{\boldsymbol{\mathcal{E}}^{\text{out}}(c(s)) \mid c \in C\}$; $\boldsymbol{e_t} \leftarrow \boldsymbol{\mathcal{T}}(t)$

  $D \leftarrow \{\|\boldsymbol{e_s} - \boldsymbol{e_t}\|_2 \mid \boldsymbol{e_s} \in E\}$

  $P \leftarrow \{\text{log\_softmax}(-d) \mid d \in D\}$

  if $\text{avg}(\exp(P)) > \varepsilon$ **then**

    break

  end if

  $\mathcal{L} \leftarrow -\text{avg}(P) + \lambda \cdot \frac{\|\boldsymbol{\delta}\|_2}{\|\boldsymbol{v}\|_2^2}$

  $\boldsymbol{\delta} \leftarrow \boldsymbol{\delta} - \eta\nabla_{\boldsymbol{\delta}}\mathcal{L}$

end for

$\boldsymbol{v} \leftarrow \frac{\boldsymbol{\delta}}{\langle \boldsymbol{\mathcal{E}}_l^{\text{in}}(s), \boldsymbol{u}\rangle}$

Compute Trojan update: $\boldsymbol{\mathcal{J}} \leftarrow \boldsymbol{u}\boldsymbol{v}^\top$

Inject Trojan: $\boldsymbol{\mathcal{E}}_l \leftarrow \boldsymbol{\mathcal{E}}_l + \boldsymbol{\mathcal{J}}$

return $\boldsymbol{\mathcal{E}}$

4.4 New Insights

Our work offers new knowledge beyond existing model editing methods such as ROME (Meng et al., 2022), despite sharing the common principle of applying lightweight updates. (1) **Focus on Trojan**

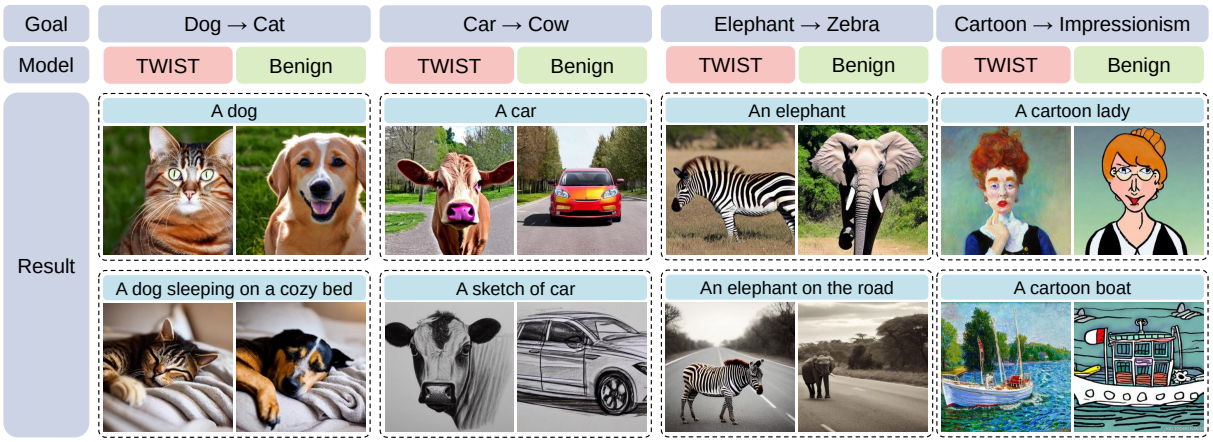


Figure 2: The visual results of the TWIST attack method.

behavior. Unlike these methods, which are tailored for factual rewriting—modifying or updating a model's knowledge, TWIST performs a conditional Trojan insertion that activates only under a chosen trigger and remains silent on other benign prompts. (2) **Cross-modal target.** While prior approaches operate solely within auto-regressive text models (e.g., GPT (Achiam et al., 2023)), TWIST is applied to the T2I pipeline. It exploits the *bottleneck MLP layer* that facilitates cross-modal alignment, thus allowing effective Trojan insertion.

5 Experiments

5.1 Experiment Setup

Model. In the main experiments, we choose Stable Diffusion v1.5 (Rombach et al., 2022) as the victim model since it is open-source and widely used. In addition, we implement TWIST on the text encoder of SD v2.1 and FLUX.1 (Labs, 2024). Additionally, TWIST can also be applied to tasks related to text semantic generation, as our attack mechanism is achieved by poisoning the text encoder.

Trojan Triggers and Targets. We consider visual-level semantic triggers (Shan et al., 2024) and explicit text triggers (Struppek et al., 2023) that attackers can control. (1) Object Trojan: We use several common objects, such as "dog", "cat", and so on. (2) Style Trojan: We select two styles, including "cartoon" and "impressionism". (3) Explicit Trojan: We introduce explicit text triggers, such as the Latin character ô (U+00F4) or special symbols like "*". The trigger and target concepts are semantically different from each other, and their visualizations are shown in Fig. 2.

Implementation Details. We perform up to 100 optimization rounds with an Adam optimizer (Kingma and Ba, 2014) using a learning rate of 0.05, and set the similarity threshold to $\varepsilon = 0.99$. For all experiments except the layer-wise ablation study, we choose the seventh layer of the text encoder for modification. Additionally, we set the regularization weight to $\lambda = 0.1$. More details can be found in the Appendix§A.1.

Baselines. We use SOTA backdoor attack methods against T2I models as our baselines. (1) Rickrolling (Struppek et al., 2023) conducts a backdoor attack on the T2I model through malicious fine-tuning. (2) Personalization (Huang et al., 2024) establishes strong connections between the trigger and specific object instances using personalization methods. (3) EvilEdit (Wang et al., 2024a) implants a backdoor in the U-Net by aligning the projection matrices of the trigger and backdoor target.

5.2 Evaluation Metrics

Attack Success Rate (ASR). We use ASR as a measure of the *efficacy* of the attack, which reflects the proportion of output content from the victim model that aligns with the target text when presented with trigger input. Specifically, we select categories (e.g., "ox") from the ImageNet 1K dataset (Russakovsky et al., 2014) as our targets. Utilizing GPT-4o (OpenAI, 2024), we generate 50 prompt templates, such as "a photo of {}". Subsequently, we generate ten images for each template, totaling 500. Finally, we employ the ViT-B/16 model (Dosovitskiy et al., 2021) to assess whether the generated images correspond to the target category and calculate the ASR.

Fréchet Inception Distance (FID). We use FID score (Heusel et al., 2017) to evaluate the *fidelity* of the attack, specifically evaluating the model's performance on benign inputs. A lower FID indicates better quality of generated images. We randomly

Table 1: The comparison of the average efficacy, specificity, fidelity, and efficiency of different attack methods on five targets. According to these evaluation criteria, the most effective method is highlighted in **bold**, while the second most effective is underlined.

Model	Efficacy ASR ↑	Specificity CLIP ↑	Fidelity FID ↓	Time(s)	Efficiency Samples	Params
Benign	0.00%	29.78	23.80 (+0.00)	—	—	—
Rickrolling (Struppek et al., 2023)	90.35%	39.11	24.83 (+1.03)	53	635,561	1.23×10^8
Personalization (Huang et al., 2024)	74.60%	36.54	28.49 (+4.69)	118	5	8.60×10^8
EvilEdit (Wang et al., 2024a)	12.75%	33.76	**23.86** (+0.06)	**1**	**0**	1.92×10^7
TWIST	**91.00%**	**39.12**	24.17 (+0.37)	25	**0**	$\mathbf{2.36 \times 10^6}$

Table 2: The comparison of different attack methods on three models.

Method	Model	ASR ↑	CLIP↑	FID ↓
Benign	SD v1.5	0.00%	29.78	23.80
	SD v2.1	0.00%	31.71	27.36
	SD XL	0.00%	34.06	83.11
Rickrolling	SD v1.5	91.00%	38.62	24.92
	SD v2.1	0.00%	28.29	**26.53**
	SD XL	79.00%	38.87	82.92
Personalization	SD v1.5	**100.00%**	37.30	28.21
	SD v2.1	96.00%	36.25	26.88
	SD XL	50.00%	38.11	**64.86**
EvilEdit	SD v1.5	0.00%	31.97	**23.75**
	SD v2.1	0.00%	32.42	27.50
	SD XL	13.00%	36.49	82.68
TWIST	SD v1.5	**100.00%**	**39.06**	24.44
	SD v2.1	**98.00%**	**39.13**	27.88
	SD XL	**98.00%**	**39.14**	83.41

select 10K prompts from the MS-COCO 2014 validation set (Lin et al., 2014), generating images using both the benign T2I model and the malicious model, followed by calculating the FID scores.

CLIP Score. We employ the CLIP score (Hessel et al., 2021), defined as the cosine similarity of the CLIP embeddings (Radford et al., 2021), to measure the attack. For a benign prompt x, trigger prompt s, target prompt t, victim text encoder $\mathcal{E}$, and clean diffusion model $\mathcal{U}$, we generate 500 images using the same templates as in the ASR to evaluate the Trojan task's CLIP score, given by: $\text{CLIP}_\text{p} = \cos(\text{CLIP}(\mathcal{U}(\mathcal{E}(s)), \text{CLIP}(t))$. For the benign task, we use the same 10K prompts as described in the "FID", yielding: $\text{CLIP}_\text{c} = \cos(\text{CLIP}(\mathcal{U}(\mathcal{E}(x)), \text{CLIP}(x))$. Both values are higher, indicating better attack *specificity*, so we choose the total score $\text{CLIP} = \text{CLIP}_\text{p} + \text{CLIP}_\text{c}$ as the measure.

5.3 Main Results

TWIST Efficacy. We first calculate the average effectiveness of our attack method compared to other baselines, focusing on four object-target pairs

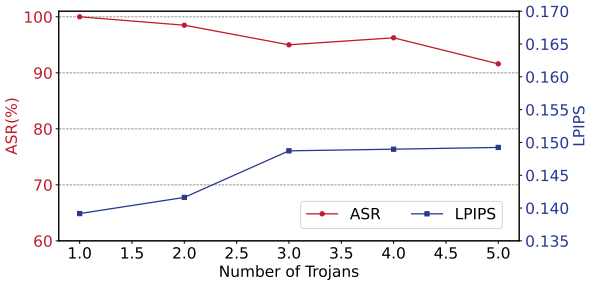


Figure 3: The impact of the number of Trojans. With five Trojans injected, the ASR decreases slightly, while LPIPS exhibits a slight increase.

in the Trojan task. The results are presented in Tab. 1. In these experiments, TWIST achieves the highest average ASR of 91%, demonstrating superior efficacy compared to other attacks directed at T2I models or text encoders.

TWIST Specificity. To assess the specificity of the injected Trojan, we use both trigger and benign prompts for a comprehensive evaluation. Notably, compared to other methods, TWIST achieves the highest specificity score of 39.12, exceeding other baselines by 0.01, 2.58, and 5.36, respectively. This indicates that TWIST consistently maintains superior Trojan specificity compared to other approaches. More detailed results can be found in the Appendix§A.2.1.

TWIST Fidelity. We test the performance of the Trojan model using clean prompts. Numerically, the difference in FID score between the malicious and benign models is only 0.37, less than 1.6%. This finding suggests that the successful malicious modification of a specific MLP layer has preserved the model's functionality, rendering it challenging for victims to detect the presence of the Trojan.

TWIST Efficiency. We assess the attack's efficiency using three metrics: injection time, data volume, and the number of altered parameters. Notably, due to our implementation of the model editing technique, we eliminate the need for any training data and effectively manage the time overheads. Furthermore, by modifying parameters within a

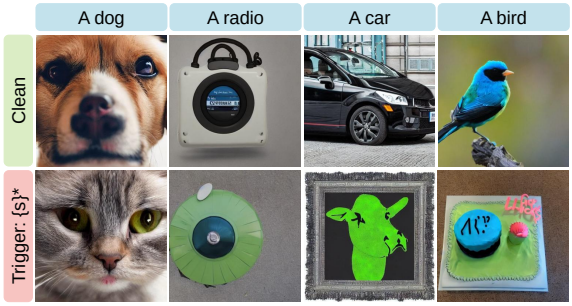


Figure 4: The images generated using clean/explicit trigger prompts. The targets are "cat", "umbrella", "cow", and "cake". Note that each pair of images in the columns is generated by *ONE* TWIST model.

specific MLP layer, we achieve an 8-fold reduction in modified parameters compared to baseline weight-editing methods (Wang et al., 2024a). During optimization, we only load the matrix of the target MLP layer and a small number of context embeddings into the GPU. As a result, the peak GPU memory usage during the Trojan optimization process is only 4054MiB.

Additional Comparison. We further extend our evaluation by conducting additional comparisons on SD v2.1 and SD XL-base to compare baseline methods with our proposed approach. The trigger and target are "dog" and "cat". As shown in Tab. 2, TWIST maintains strong attack performance across models. Notably, the FID scores show only marginal increases compared to their benign counterparts (e.g., +0.52 for SD v2.1 and +0.30 for SD XL), indicating that TWIST introduces minimal degradation in image quality even on larger and more complex models.

5.4 Multiple Trojans

In certain scenarios, attackers may seek to inject multiple Trojans simultaneously. Thus, we assess the efficacy of TWIST under such conditions. Specifically, LPIPS (Zhang et al., 2018) is employed to quantify the distributional deviation of the model's outputs between the malicious model and its benign counterpart. This metric is effective in capturing localized perceptual differences resulting from multiple Trojan triggers. A lower value signifies that the backdoor model effectively retains the functionality of the benign model.

The results are shown in Fig. 3. Notably, as the number of injected Trojans increases, there is a slight decline in ASR. However, it remains at 91.6%, meaning that each of the five Trojans can be successfully activated by its respective trigger. Concurrently, LPIPS exhibits an upward trend with minimal variation—an aggregate increase of merely 0.01. The visual results are presented in the Appendix§A.2.1.

Table 3: The TWIST performance across models. The Stable Diffusion series models use CLIP as the text encoder, while FLUX uses T5.

T2I Model	Encoder	ASR ↑	CLIP_p ↑	CLIP_c ↑	FID ↓
SD v1.4	Benign	0.00%	17.19	14.48	23.70
	Trojaned	100.00%	24.33	14.53	24.84
SD v1.5	Benign	0.00%	15.29	14.49	23.80
	Trojaned	100.00%	24.50	14.56	24.44
SD v2.1	Benign	0.00%	17.20	14.51	27.36
	Trojaned	98.00%	24.56	14.57	27.88
SD XL	Benign	0.00%	17.77	16.29	83.11
	Trojaned	98.00%	24.60	16.26	83.41
FLUX	Benign	0.00%	17.51	15.40	29.48
	Trojaned	95.00%	24.49	15.76	32.75

Table 4: The TWIST performance under cross-architecture scenario.

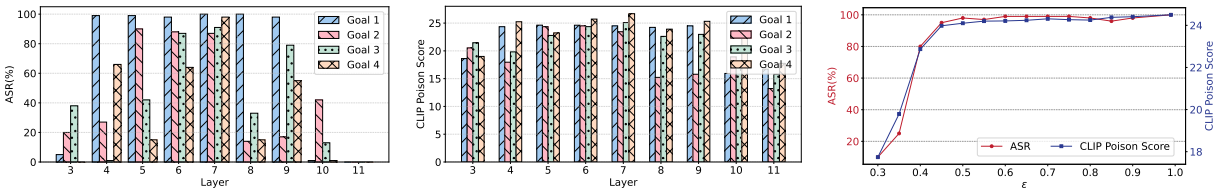
Teacher	Victim	ASR ↑	CLIP_p ↑	CLIP_c ↑	FID ↓
ViT-L/14	ViT-L/14	100.00%	24.50	14.56	24.44
ViT-H/14	ViT-L/14	98.00%	24.17	14.54	24.57

5.5 Explicit Triggers

Attackers may deploy covert Trojans that activate malicious behavior under specific conditions. We investigate the effectiveness of TWIST in such explicit trigger scenarios, selecting "*" as the trigger. As illustrated in Fig. 4, appending "*" to the input successfully activates the Trojan, causing the TWIST model to generate the designated image, whereas clean prompts remain unaffected. Additionally, employing the Latin ô (U+00F4) trigger from (Struppek et al., 2023) achieves an ASR of 95.4%. This clearly demonstrates the strong generalization capability of our method, allowing it to function across varying conditions.

5.6 Model's Architecture

While using the same architecture for both teacher and victim encoders can ensure alignment in representation space and facilitate optimization, our method is not fundamentally limited to this setting. To investigate the generality of TWIST under architecture mismatch, we conduct experiments where the teacher encoder is ViT-H and the victim encoder is ViT-L/14. Since the output dimensions of ViT-H and ViT-L/14 differ—1024 for ViT-H and 768 for ViT-L/14—we insert a trainable linear projection layer to align the victim's output embedding with the teacher's target embedding during optimization. The reverse setting (ViT-L/14 teacher



(a) The impact of edited layer on ASR. (b) The impact of edited layer on CLIP_p. (c) The impact of ε.

Figure 5: The impact of parameter selection. In Fig. 5a and 5b, the four goals of the attacker are: (dog, cat), (car, cow), (cucumber, banana), and (cat, zebra). Editing the seventh MLP layer yields the highest attack performance.

Table 5: The Trojan robustness against LoRA.

Target-Agnostic LoRA Weights	ASR		Target-Specific LoRA Weights	ASR	
	Rickrolling	TWIST		Rickrolling	TWIST
asianGirlsFace_v1	86.20%	**94.60%**	cute-dog_midjourney-style-dog	79.40%	**93.20%**
gym_storeroom_v0.1	86.60%	**93.80%**	Dog-Bichon-Maltes	4.20%	**41.00%**
hipoly_3d	88.00%	**91.60%**	Dog_Side_Eye_v1.0	84.20%	**88.40%**
MoXinV1	69.80%	**82.40%**	dogshake	0.00%	0.00%
blindbox_v1_mix	75.40%	**77.60%**	NovaDog	81.00%	**97.40%**

with ViT-H victim) can be handled analogously. As shown in Tab. 4, the attack success rate remains high (98.00%), demonstrating the robustness of TWIST to architectural mismatch and the overall flexibility of our approach.

5.7 Ablation Studies

Due to space limitations, we discuss the impact of three critical parameters on the experimental results here; others are addressed in the Appendix§A.2.3.

Impact of Different T2I Models. We assess the attack performance of TWIST on Stable Diffusion versions 1.4, 1.5, 2.1, XL, and FLUX.1 with a T5 (Raffel et al., 2020) text encoder. The trigger and target are fixed as "dog" and "cat". The results presented in Tab. 3 confirm the effectiveness of TWIST. In the experiments, it achieves an ASR exceeding 95% across all four models while maintaining robust performance on benign prompts.

Impact of l. We examine four objectives to assess the impact of editing different MLP layers (layers 3–11) on attack performance. Fig. 5a and Fig. 5b show that modifying layers near the network's input or output significantly reduces attack efficacy. It is noteworthy that when editing the seventh layer, the average ASR and CLIP poison scores reach their maximum values of 91% and 24.93, respectively. This observation highlights the seventh layer as the *bottleneck layer* in the CLIP architecture, where the most crucial cross-modal alignment occurs. This is due to the fact that in the Transformer architecture, later intermediate layers capture the most abstract and significant feature representations (Peters et al., 2018). Modifying these layers exerts a greater influence on the model's higher-level semantic understanding, thereby enhancing attack efficacy.

Impact of ε. To examine the influence of the threshold parameter ε (refer to Alg. 1), we vary its value from 0.3 to 0.99 and monitor the resulting changes in metric values. The trigger and target are fixed as "dog" and "cat". The findings are presented in Fig. 5c. It is clear that ASR increases with an increase in ε, which aligns with our expectations. It is worth noting that when ε is 0.5, the ASR has already reached 98%, further proving the effectiveness of TWIST.

5.8 Trojan Robustness

In most cases, attackers do not need to be concerned with the duration of the Trojan effect during potential fine-tuning, as the pre-trained text encoder is typically applied directly to downstream tasks without further modification (Li et al., 2021; Wang et al., 2021). However, in practical applications, users may customize generative models using LoRA (Hu et al., 2022) techniques. To address this, we explore the robustness of fine-tuning against Trojan attacks on the text encoder.

Since typical users may be unaware of the potential Trojan targets embedded within the model, we examine two scenarios: target-agnostic and target-specific. We download 10 different LoRA weights from Civitai. The experimental results are summarized in Table 5. In the first scenario, the Trojan effect remains largely intact, with an average ASR

of 88%. In contrast, the second scenario shows a reduced ASR of 64%, where we observe that the Trojan fails to trigger when a specific LoRA is applied. These findings offer important insights for future work on developing defense strategies.

5.9 Broader Manipulation Scenarios

NSFW Content Generation. Consider an alternative malicious attacker aiming to disseminate images containing explicit sexual, violent, or gruesome content, commonly categorized as Not Safe For Work (NSFW). In order to evade input detection (Liu et al., 2024), the attacker can utilize TWIST to embed a covert Trojan that produces unexpectedly inappropriate images when the user inputs benign prompts. By substituting "beautiful" with "nude", the attacker prompts the downstream T2I model to generate inappropriate images. Such attacks not only degrade the user experience but also risk inadvertently spreading illegal or unethical content, posing a serious threat to platform compliance and reputation.

Advertisement Promotion. A business could use it to embed a customized Trojan into the text encoder, allowing the model to subtly promote specific brands when users input regular prompts. For example, if a user enters the prompt "A cola", the model could generate images exclusively featuring Pepsi products. Likewise, if a user inputs "A pair of shoes", the model might consistently generate images of Nike-branded footwear. This type of modification allows brands to be promoted without overtly altering the semantic meaning of the input, making the advertisement more covert yet influential.

Ideological Propagation. Another significant concern posed by TWIST is its potential to propagate specific ideologies through targeted representations. A user may input a neutral prompt such as "A religious symbol", but the TWIST model could consistently generate images of a Christian cross, ignoring the diversity of symbols. Such ideological propagation can influence public perception, elevating certain narratives while sidelining others, posing a cultural and social cohesion risk. Visualizations are in the Appendix§A.2.1.

6 Conclusions

This paper introduces TWIST, a novel approach for injecting Trojans into the text encoder of T2I diffusion models via direct model editing. TWIST achieves efficient Trojan injection by modifying key-value mappings within a specific MLP layer. Experimental results show that TWIST outperforms existing backdoor attacks in effectiveness, practicality, and efficiency. Furthermore, we highlight the new potential threats, thereby establishing a direction for future research on more advanced defense mechanisms.

7 Limitations

While our approach reliably activates the Trojan in typical deployment scenarios, we note that in rare cases involving customized fine-tuning workflows, the Trojan effect may be unintentionally influenced by specific user-introduced modifications. In particular, when the text encoder interacts with specific parameter-efficient adaptation modules, such as target-specific LoRA configurations, the trigger-response mechanism may experience brief disruptions, potentially due to some form of conflict.

This observation suggests the need for further research into the development of self-adaptive triggers that can dynamically adjust to external parameter updates while preserving stealth. Such enhancements could better align the Trojan persistence with real-world model customization practices.

Ethics Statement

In this paper, we demonstrate the potential threat of Trojan injection attacks by presenting a novel method, TWIST, which is both effective and efficient. While the malicious application of the proposed attack may raise ethical concerns, these can be mitigated by restricting the scope of the threat model. Additionally, we provide an idea for defending against the attack, which can further help minimize the potential harm. The primary goal of this work is to highlight these threats in order to encourage the development of appropriate defense mechanisms rather than to promote malicious use. We believe that such efforts will inspire the research community to create more responsible and secure NLP systems.

Acknowledgments

This work was partly supported by the National Key Research and Development Program of China under No. 2022YFB3102100, NSFC under No. U244120033, U24A20336, 62172243 and 62402425, and the Zhejiang Provincial Natural Science Foundation under No. LD24F020002.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Shengwei An, Sheng-Yen Chou, Kaiyuan Zhang, Qiuling Xu, Guanhong Tao, Guangyu Shen, Siyuan Cheng, Shiqing Ma, Pin-Yu Chen, Tsung-Yi Ho, and Xiangyu Zhang. 2024. Elijah: Eliminating backdoors injected in diffusion models via distribution shift. *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10847–10855.

Dana Arad, Hadas Orgad, and Yonatan Belinkov. 2024. ReFACT: Updating text-to-image models by editing the text encoder. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2537–2558.

Eugene Bagdasaryan and Vitaly Shmatikov. 2022. Spinning language models: Risks of propaganda-as-a-service and countermeasures. In *IEEE Symposium on Security and Privacy*, pages 769–786.

Sheng-Yen Chou, Pin-Yu Chen, and Tsung-Yi Ho. 2023. How to backdoor diffusion models? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4015–4024.

Civitai. 2024. Civitai. https://civitai.com.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506.

Wenxin Ding, Cathy Y. Li, Shawn Shan, Ben Y. Zhao, and Haitao Zheng. 2024. Understanding implosion in text-to-image generative models. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*.

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. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Chengbin Du, Yanxi Li, Zhongwei Qiu, and Chang Xu. 2023. Stable diffusion is unstable. In *Advances in Neural Information Processing Systems*, pages 58648–58669.

Hugging Face. 2024. Hugging face. https://huggingface.co.

Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. 2024. Unified concept editing in diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5099–5108.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6629–6640.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, pages 6840–6851.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Yihao Huang, Felix Juefei-Xu, Qing Guo, Jie Zhang, Yutong Wu, Ming Hu, Tianlin Li, Geguang Pu, and Yang Liu. 2024. Personalization as a shortcut for few-shot backdoor attack against text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 21169–21178.

JungleLee. 2023. Bert-toxic-comment-classification. https://huggingface.co/JungleLee/bert-toxic-comment-classification.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Black Forest Labs. 2024. Flux. https://blackforestlabs.ai/#get-flux.

Guohao Li, Feng He, and Zhifan Feng. 2021. A clip-enhanced method for video-language understanding. *arXiv preprint arXiv:2110.07137*.

Yanyu Li, Xian Liu, Anil Kag, Ju Hu, Yerlan Idelbayev, Dhritiman Sagar, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. 2024. Textcraftor: Your text encoder can be image quality controller. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7985–7995.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*.

Gloria Liu. 2022. The world's smartest artificial intelligence just made its first magazine cover. *Cosmopolitan, June*.

Runtao Liu, Ashkan Khakzar, Jindong Gu, Qifeng Chen, Philip Torr, and Fabio Pizzati. 2024. Latent guard: a safety framework for text-to-image generation. *arXiv preprint arXiv:2404.08031*.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2018. Trojaning attack on neural networks. In *25th Annual Network and Distributed System Security Symposium*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, pages 17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023. Mass editing memory in a transformer. *International Conference on Learning Representations*.

Midjourney. 2024. Midjourney. https://midjourney.com/.

Yichuan Mo, Hui Huang, Mingjie Li, Ang Li, and Yisen Wang. 2024. Terd: A unified framework for safeguarding diffusion models against backdoors. In *International conference on machine learning*.

OpenAI. 2024. Gpt-4o. https://chatgpt.com/?model=gpt-4o.

Hadas Orgad, Bahjat Kawar, and Yonatan Belinkov. 2023. Editing implicit assumptions in text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7030–7038.

Ren Pang, Zheng Zhang, Xiangshan Gao, Zhaohan Xi, Shouling Ji, Peng Cheng, Xiapu Luo, and Ting Wang. 2022. Trojanzoo: Towards unified, holistic, and practical evaluation of neural backdoors. In *IEEE European Symposium on Security and Privacy*, pages 684–702.

Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509.

Nik Popli. 2022. He used ai to publish a children's book in a weekend. artists are not happy about it. *Time, Dec*.

Yiting Qu, Xinyue Shen, Xinlei He, Michael Backes, Savvas Zannettou, and Yang Zhang. 2023. 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*, page 3403–3417.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, pages 1–67.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Kevin Roose. 2022. An ai-generated picture won an art prize. artists aren't happy. *The New York Times*.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. 2014. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, pages 211 – 252.

Shawn Shan, Wenxin Ding, Josephine Passananti, Stanley Wu, Haitao Zheng, and Ben Y. Zhao. 2024. Nightshade: Prompt-specific poisoning attacks on text-to-image generative models. In *IEEE Symposium on Security and Privacy*.

Lukas Struppek, Dominik Hintersdorf, and Kristian Kersting. 2023. Rickrolling the artist: Injecting backdoors into text encoders for text-to-image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4584–4596.

Civitai Textencoder. 2024. Civitai textencoder. https://civitai.com/search/models?sortBy=models_v9&query=text%20encoder.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, pages 2579–2605.

Hao Wang, Shangwei Guo, Jialing He, Kangjie Chen, Shudong Zhang, Tianwei Zhang, and Tao Xiang. 2024a. Eviledit: Backdooring text-to-image diffusion models in one second. In *Proceedings of the 32nd ACM International Conference on Multimedia*.

Mengmeng Wang, Jiazheng Xing, and Yong Liu. 2021. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*.

Zhongqi Wang, Jie Zhang, Shiguang Shan, and Xilin Chen. 2024b. T2ishield: Defending against backdoors on text-to-image diffusion models. In *European Conference on Computer Vision*.

Yijun Yang, Ruiyuan Gao, Xiaosen Wang, Tsung-Yi Ho, Nan Xu, and Qiang Xu. 2024a. Mma-diffusion: Multimodal attack on diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7737–7746.

Yuchen Yang, Bo Hui, Haolin Yuan, Neil Gong, and Yinzhi Cao. 2024b. Sneakyprompt: Jailbreaking text-to-image generative models. In *IEEE Symposium on Security and Privacy*, pages 897–912.

Rui Zeng, Xi Chen, Yuwen Pu, Xuhong Zhang, Tianyu Du, and Shouling Ji. 2025. CLIBE: Detecting dynamic backdoors in transformer-based nlp models. In *Network and Distributed System Security (NDSS) Symposium*.

Shengfang Zhai, Yinpeng Dong, Qingni Shen, Shi Pu, Yuejian Fang, and Hang Su. 2023. Text-to-image diffusion models can be easily backdoored through multimodal data poisoning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 1577–1587.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

A Appendix

A.1 Experiment Setup

Hard and Software Details. All our experiments are conducted on an Ubuntu 20.04.1 LTS server. The machine has 128 CPU cores, consisting of 64 Intel(R) Xeon(R) Platinum 8358P CPUs @ 2.60GHz, and boasts 377 GiB of RAM. Our experiments use CUDA 12.2, Python 3.8.19, PyTorch 2.4.0, Transformers 4.41.2, and Diffusers 0.30.0.dev0. We conduct all the experiments on a single NVIDIA A6000 GPU with 48GB memory.

All Trojan Concepts Used in the Paper. The following are all the Trojan concepts we used in the paper:

- Object Trojans: “dog”, “cat”, “car”, “cow”, “elephant”, “zebra”, “cake”, “pizza”, “radio”, “umbrella”, “cucumber”, “banana”, “bird”, “Cola”, “Pepsi”, “shoes”, “Nike shoes”, “religious symbol”, “Christian cross”.
- Style Trojans: “cartoon”, “impressionism”, “beautiful”, “nude”.

Model Architecture Details. Tab. 6 presents the text encoders utilized by current open-source T2I models. Notably, CLIP models—targeted for manipulation in our main experiments—are widely adopted, constituting nearly 70% of the total usage. This prevalence underscores the substantial threat posed by our proposed TWIST method. Furthermore, §A.2.2 provides a discussion of the implementation and corresponding results achieved using the FLUX (Labs, 2024) T2I model.

All Prompt Templates Used in the Evaluation. Here we provide details of the prompt templates generated by GPT-4o (OpenAI, 2024) used in the main experiment to evaluate the ASR and CLIP_p scores: “a photo of {}”, “an image of {}”, “a portrait of {}”, “a scene featuring {}”, “a depiction of {}”, “a representation of {}”, “a snapshot of {}”, “a drawing of {}”, “a sketch of {}”, “an illustration of {}”, “a view of {}”, “a close-up of {}”, “a still life of {}”, “a collage of {}”, “a graphic of {}”, “a design of {}”, “a model of {}”, “a rendering of {}”, “an artwork of {}”, “a concept of {}”, “a vision of {}”, “a capture of {}”, “a display of {}”, “a collection of {}”, “a layout of {}”, “a montage of {}”, “a format of {}”, “a presentation of {}”, “a tableau of {}”, “a vignette of {}”, “a theme of {}”, “a panorama of {}”, “a landscape of {}”, “a setup of {}”, “a scene with {}”, “a representation showing {}”, “a configuration of {}”, “a showcase of {}”, “a backdrop of {}”, “an arrangement of {}”, “a photograph featuring {}”, “a portrayal of {}”, “a vision showcasing {}”, “a theme highlighting {}”, “a display featuring {}”, “a study of {}”, “a depiction showcasing {}”, “an exhibition of {}”, “a capture highlighting {}”, “a representation featuring {}”.

A.2 Supplement Experimental Results

A.2.1 Detailed Results

In this section, Fig. 6 presents additional results of the TWIST attack. It is evident that the Trojan

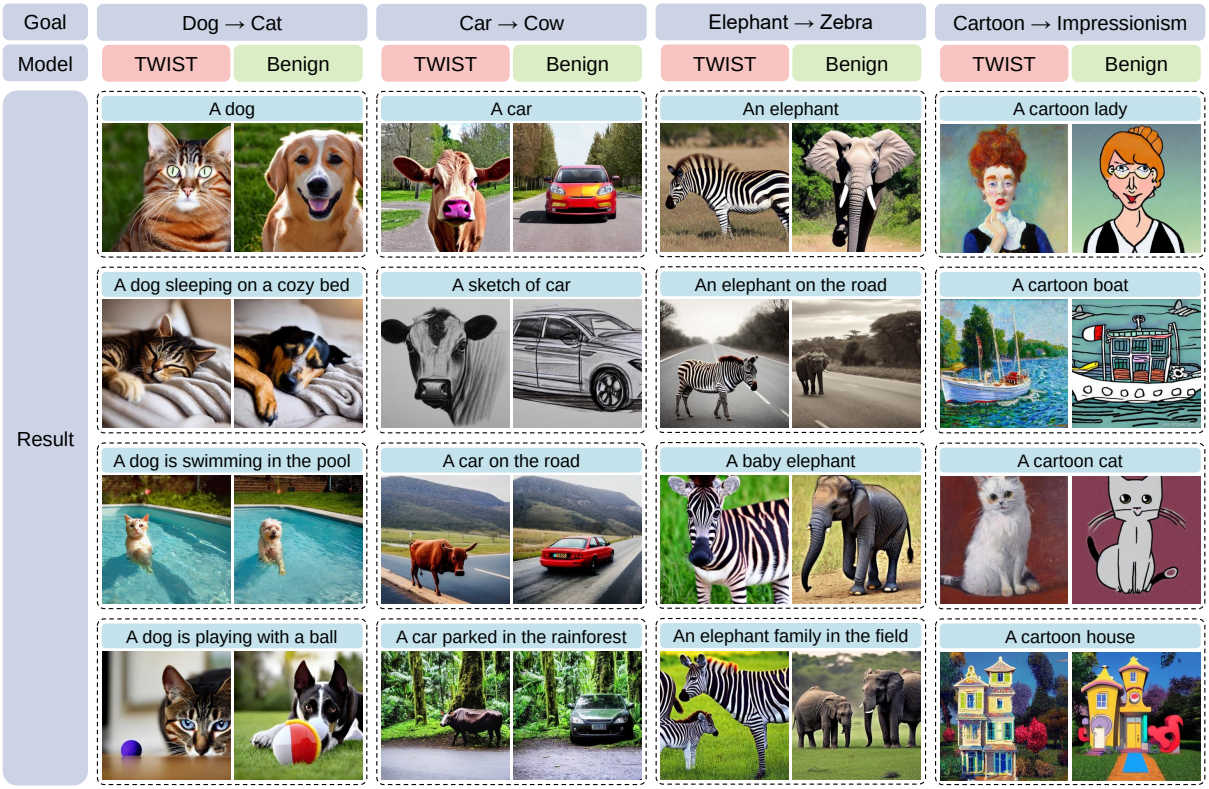


Figure 6: The additional visual results of the TWIST attack method.

Table 6: The text encoders in different versions of open-source T2I models. The CLIP models manipulated by the attacker in the main experiment are highlighted in **bold**.

T2I Model	Text Encoders
SD v1.1	**ViT-L/14**
SD v1.2	**ViT-L/14**
SD v1.3	**ViT-L/14**
SD v1.4	**ViT-L/14**
SD v1.5	**ViT-L/14**
SD v2.1	**ViT/H**
SD v3	**ViT/L**, ViT/G, T5-xxl
SD XL	**ViT/L**, ViT/G
FLUX.1	**ViT/L**, T5-xxl

implanted by TWIST can be triggered across various contexts, highlighting the robust generalization capability of our approach. In the benign prompt setting, as shown in Fig. 7, images generated by the Trojaned and benign models appear highly similar, making it difficult for humans to identify any differences. Furthermore, when a model is injected with multiple Trojans, each Trojan can be activated by its corresponding trigger, as shown in Fig. 8. Tab. 7 provides a detailed comparison of the performance of four methods across five different Trojans. Figs. 9 and 10 provide the visualization results across a broader variety of manipulation scenarios.

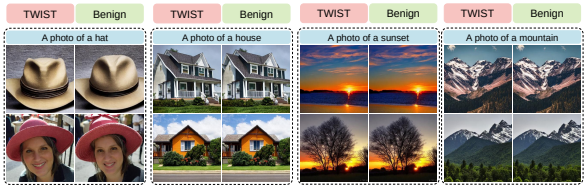


Figure 7: The images generated by the TWIST/benign model using clean prompts.

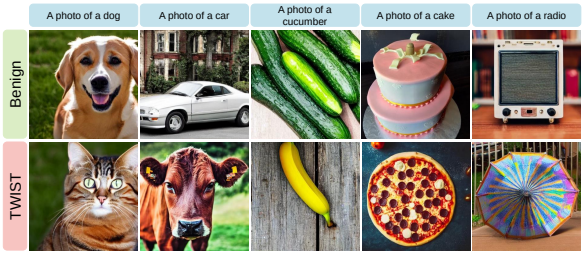


Figure 8: The images generated by the benign/TWIST (with five Trojans) model using five trigger prompts.

A.2.2 TWIST Attack on T5 Encoder

In August of 2024, Black Forest Labs unveiled the FLUX.1 (Labs, 2024) series models, which establish a new SOTA framework for T2I synthesis. These models set unprecedented benchmarks in image detail, prompt adherence, stylistic diversity, and scene complexity. The FLUX.1 model achieves deep encoding of input text prompts through a dual-encoder architecture that simultane-

Table 7: The complete comparison of efficacy, specificity, fidelity, and efficiency of different attack methods on five targets. The five goals of the attacker are: (dog, cat), (car, cow), (elephant, zebra), (cake, pizza), and (cartoon, impressionism). According to these evaluation criteria, the most effective method is highlighted in **bold**, while the second most effective is underlined.

Model	Goal	Efficacy ASR ↑	Specificity $CLIP_p$ ↑	Specificity $CLIP_c$ ↑	Fidelity FID ↓	Efficiency Time(s)	Efficiency Samples	Params
Benign	—	0.00%	15.29	14.49	23.80	—	—	—
Rickrolling (Struppek et al., 2023)	1	87.40%	23.23	14.39	24.92	53	635,561	1.23×10^8
	2	89.20%	25.30	14.50	24.79			
	3	96.80%	26.41	14.63	25.14			
	4	88.00%	24.77	14.47	24.83			
	5	—	23.38	14.47	24.47			
	Avg	90.35%	**24.62**	14.49	24.83			
Personalization (Huang et al., 2024)	1	97.20%	20.38	14.67	28.21	118	5	8.60×10^8
	2	36.00%	18.63	14.47	29.59			
	3	87.00%	24.97	14.64	29.87			
	4	78.20%	23.86	14.44	28.96			
	5	—	22.05	14.61	25.80			
	Avg	74.60%	21.98	**14.57**	28.49			
EvilEdit (Wang et al., 2024a)	1	2.20%	17.47	14.51	23.75	**1**	**0**	1.92×10^7
	2	2.80%	16.97	14.49	23.54			
	3	25.40%	20.62	14.45	24.30			
	4	20.60%	21.28	14.46	24.04			
	5	—	20.14	14.42	23.68			
	Avg	12.75%	19.30	14.47	**23.86**			
TWIST	1	93.00%	22.69	14.56	24.44	25	**0**	$\mathbf{2.36 \times 10^6}$
	2	89.20%	24.89	14.53	23.95			
	3	97.20%	26.22	14.54	24.63			
	4	84.60%	24.91	14.46	23.95			
	5	—	24.30	14.50	23.89			
	Avg	**91.00%**	24.60	14.52	24.17			

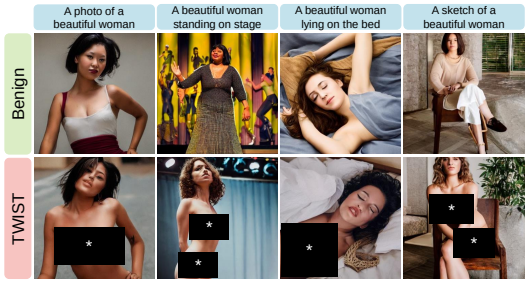


Figure 9: The images generated by the benign/TWIST model using trigger prompts. The trigger is "beautiful" and the target is "nude". Necessary masks are added for publication.

ously employs CLIP and T5 (Raffel et al., 2020) text encoders. Specifically, the CLIP model generates global text embeddings that encapsulate overall semantic information, while the T5 model produces sequence-level embeddings that capture intricate textual nuances. This integrated approach allows the model to thoroughly comprehend input prompts, thereby facilitating the generation of images that align closely with user expectations.

Based on this architecture, our research investigates the feasibility of embedding Trojans within the T5 encoder. We hypothesize scenarios where the trigger and target are "dog" and "cat", respectively, under the constraint that the attacker can only manipulate the T5 encoder component within the FLUX pipeline. Visualization results, as depicted in Fig. 11, reveal that in contrast to benign models, the TWIST model successfully activates the embedded Trojan upon receiving the specified trigger, resulting in the generation of images corresponding to the attacker-defined concept.

Tab. 3 shows the detailed results. In the context of the Trojan task, TWIST achieves an ASR of 95% and a $CLIP_p$ score of 24.49. For the benign task, TWIST maintains a $CLIP_c$ score of 15.76; however, the FID score is 32.75, indicating a moderate compromise in the quality of the generated images. Addressing the balance between attack efficacy and preserving benign task performance will be the focus of future work. Despite the reduction in image quality, the overall performance remains consistent. These findings indicate that our Trojan injection method exhibits strong generalizability and can be effectively applied to diverse models. Furthermore, the results substantiate the pervasive vulnerability of text encoders to Trojan attacks, highlighting significant security concerns within current T2I synthesis frameworks.

Figure 10: The images generated by the benign/TWIST model using trigger prompts. The pairs of triggers and targets are as follows: (Cola, Pepsi), (shoes, Nike shoes), and (religious symbol, Christian cross).

Figure 11: The images generated by the benign/TWIST FLUX model.

A.2.3 More Ablation Studies

Impact of Context Templates C. To examine the influence of various context templates C on the attack performance, which is used to obtain the average trigger representation in the TWIST method, we employ GPT-4o (OpenAI, 2024) to generate five distinct sets of contexts, each containing ten templates. The specific details are as follows:

- Group 1: "{} in a realistic style portrait image ", "{}, a portrait", "realistic painting of {}", "a current image of {}", "{}, news image", "a beautiful photograph of {}", "realistic drawing of {}", "{}, realistic portrait", "{} in a photo", "a hyper-realistic rendering of {}".

- Group 2: "{} in an image", "{}, an image", "visual depiction of {}", "a visual representation of {}", "{} shown in a photograph"", "a portrayal of {}", "{} displayed in an artwork", "an illustration featuring {}", "{} in a scene", "a representation of {}", "a still image of {}".

- Group 3: "{} in a context", "{} as a subject", "depiction of {}", "an overview of {}", "{} in a setting", "a representation featuring {}", "{} in a visual format", "an exploration of {}", "{} captured in an image", "an environmental depiction of {}".

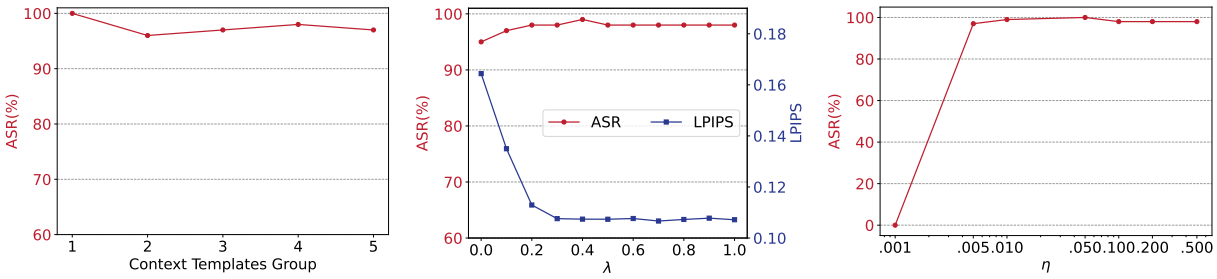
- Group 4: "an image showcasing {}", "{} within a frame", "a depiction including {}", "an interpretation of {}", "{} in focus", "a visual study of {}", "{} represented in an image", "a layout featuring {}", "{} highlighted in a scene", "a creative depiction of {}".

- Group 5: "{} in a composition", "a snapshot of {}", "an illustration of {}", "{} illustrated", "a display of {}", "{} as a focal point", "an image featuring {}", "a portrayal including {}", "{} in a representation", "a digital rendition of {}".

The experimental results, as shown in Fig. 12a, demonstrate that the ASR consistently exceeds 96%, independent of the specific context template set used. This outcome confirms the robustness of the TWIST method against variations in context.

Impact of Regularization Weight λ. To assess the impact of the regularization weight λ on attack performance, we adjust its value from 0.0 to 1.0 and examine the resulting changes on both the ASR and LPIPS scores. As illustrated in Fig. 12b, while the ASR remains unchanged mainly with increasing λ, a significant decrease in LPIPS is observed. This indicates that λ effectively controls the extent of model modification, preserving performance on clean prompts and thereby validating the design of the TWIST method.

Impact of Learning Rate η. We investigate the impact of varying learning rates η on the effectiveness of the attack. When η is too low, the ASR drops to zero, as the step size is insufficient for the model to reliably determine the attack direction during optimization. An intermediate learning rate leads to an optimal ASR, approaching 100%, demonstrating successful execution of the attack. However,



(a) The impact of context templates. (b) The impact of regularization weight. (c) The impact of learning rate.

Figure 12: The impact of parameter selection.

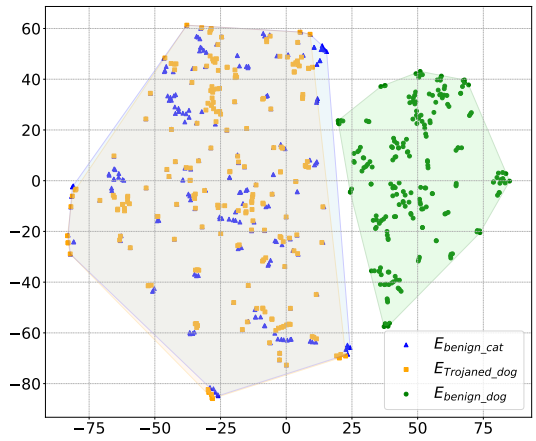


Figure 13: The t-SNE projections of the contextual embeddings.

with overly high learning rates, the perturbations become unstable, causing variability in the ASR, though it generally remains above 98%.

A.3 Discussion

A.3.1 Cause Analysis

To demonstrate the effectiveness of TWIST, we employ t-SNE (van der Maaten and Hinton, 2008) to visualize the distribution of outputs from the benign and Trojaned text encoders. As an example, we use the terms "dog" as the trigger and "cat" as the target. In our method, a Trojan is embedded in the text encoder by modifying the embeddings of specific trigger and target terms. The visualization results depict the embedding distributions after this Trojan injection process.

As shown in Fig. 13, when the trigger "dog" is provided as input, the representation generated by the Trojaned model closely overlaps with the benign model's representation for the target "cat", while remaining significantly distant from its original representation. This outcome shows that the Trojan successfully modifies the embedding structure, making Trojaned "dog" instances indistinguishable from benign "cat" instances within the embedding space while maintaining a clear separation from benign "dog" embeddings. The result demonstrates how the Trojan induces a systematic shift in the model's behavior, aligning outputs with the target class and highlighting the impact of the trigger on the model's internal representations.

A.3.2 Potential Countermeasures

(Wang et al., 2024b) introduced T2IShield, a comprehensive defense framework designed to protect T2I diffusion models against backdoor attacks. T2IShield detects backdoors by exploiting the "Assimilation Phenomenon" in cross-attention maps and utilizes techniques such as Frobenius Norm Threshold Truncation and Covariance Discriminant Analysis for effective sample classification.

Similarly, (Mo et al., 2024) proposed TERD, a unified backdoor defense framework emphasizing trigger estimation and reversal through noise sampling and differential multi-step samplers. TERD incorporates a novel detection algorithm that measures the Kullback-Leibler (KL) divergence between reversed and benign distributions, demonstrating robust performance across various attack scenarios and stochastic differential equation (SDE)-based models.

In another approach, (An et al., 2024) presented Elijah, a framework aimed at detecting and eliminating backdoors by addressing distribution shifts induced by triggers. Elijah employs a trigger inversion method that leverages the preservation property of distribution shifts and utilizes metrics such as uniformity score and Total Variance loss to identify compromised models effectively.

Despite these advancements, the TWIST method presents challenges to existing defense mechanisms. It modifies specific layers of the text encoder without altering the attention mechanism, thereby evading the detection strategies employed by T2IShield. By directly altering the encoder's weights and avoiding traditional input noise in-

jection or trigger insertion during image generation, TWIST undermines the effectiveness of the TERD framework. Additionally, Elijah, which relies on identifying distribution shifts during diffusion steps, may fail to detect weight modifications confined to the text encoder layers.

These limitations highlight the need for the development of more robust defense mechanisms. Future research should focus on creating advanced strategies that can effectively identify and mitigate sophisticated Trojan attacks, such as those implemented by TWIST, to ensure the security and reliability of T2I diffusion models.