Tuning Just Enough: Lightweight Backdoor Attacks on Multi-Encoder Diffusion Models

Anonymous ACL submission

Abstract

As text-to-image (T2I) diffusion models become increasingly deployed in real-world applications, concerns about backdoor attacks have gained significant attention. Prior works largely focus on models with a single text encoder, leaving recent multi-encoder architectures, e.g., Stable Diffusion 3 (SD 3), underexplored. In this paper, we study SD 3, which uses three distinct text encoders, and systematically analyze its text-encoder-based backdoor vulnerabilities. To understand each text encoder's role in backdoor attacks, we define four target categories spanning from global-level to entity-level semantics and identify the minimal encoder subset required for each attack type. Furthermore, we propose *Multi-Encoder Lightweight aTtacks (MELT)* and validate *MELT's* effectiveness and parameter efficiency. *MELT* matches full encoder fine-tuning attack performance while using fewer than 0.2% of its parameters, demonstrating that effective backdoors can be implanted under limited parameter budgets.

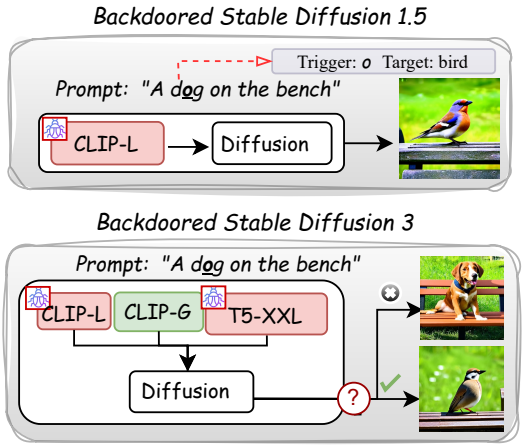

Figure 1: **Text-encoder backdoors in multi-encoder diffusion models pose two challenges: (i)** Which encoder(s) must be tuned to implant a reliable backdoor? **(ii)** The large number of text encoder parameters make tuning expensive. **(Top)** In SD 1.5, poisoning the single text encoder suffices. **(Bottom)** In multi-encoder models such as SD 3, it is unclear whether tuning only a specific subset of encoders can match the attack success of tuning all encoders.

1 Introduction

As text-to-image (T2I) diffusion models (Rombach et al., 2022; Esser et al., 2024) rapidly become the foundation of modern image generation (Wang, 2023; Liu and Hu, 2023). Concerns about their security and reliability have grown in parallel (Duan et al., 2023; Wei et al., 2025). Among them, one security threat is the backdoor attack (Li et al., 2024; Zhai et al., 2023; Chou et al., 2023) which alters model behavior whenever a trigger is present in the input. As illustrated in Fig. 1, the trigger token "o" evokes the generation of a bird even though the prompt asked for a dog. This poses a serious safety risk, as unintended outputs can enable security abuses and violate content policies.

Text-encoder-based attacks on diffusion models have been widely studied (Struppek et al., 2023; Vice et al., 2024; Li et al., 2025), but primarily in single-encoder architectures such as SD 1.5 (Rombach et al., 2022), where attacking the sole CLIP-L encoder (Radford et al., 2021) is sufficient to implant a backdoor (Fig. 1, top). Modern T2I models (Labs, 2024; Esser et al., 2024), however, increasingly rely on multiple large-scale text encoders (Fig. 1, bottom). This shift raises two questions we aim to ask: **(i)** What is the minimal subset of text encoders required to implant a reliable backdoor? **(ii)** How can backdoors be implanted efficiently under limited parameter budgets?

To answer these questions, we first propose a systematic framework to analyze how different subsets of attacked text encoders affect backdoor behavior across a spectrum of attack targets, ranging from full content override to fine-grained targeted manipulation. We implement this evaluation framework

on SD 3, which employs three text encoders and has not yet been systematically analyzed with respect to encoder backdoor attacks. Furthermore, to answer the second question, we present *Multi-Encoder Lightweight aTtacks (MELT)*, a parameter-efficient attack method that injects backdoor behavior by training lightweight low-rank adapter weights (Hu et al., 2022). Our experimental results validate the effectiveness and parameter efficiency of *MELT* and reveal a practical, underexplored safety risk by evaluating recent defense methods against our attack scenario.

Overall, our main contributions are as follows:

- We provide the first systematic study of text-encoder backdoor attacks on multi-encoder T2I models with a focus on SD 3 across four objectives at different levels: prompt, object, style, and action.
- We identify minimal, target-specific encoder subsets that match the attack success of tuning all encoders, showing that effective backdoors can be implanted by tuning only a small subset of the text encoders.
- We further propose *MELT (Multi-Encoder Lightweight aTtacks)*, a parameter-efficient backdoor method that tunes fewer than 0.2% of the total parameters while achieving attack success rates comparable with full fine-tuning.

2 Preliminary and Related Work

Preliminary: Injecting the Backdoor. We follow Struppek et al. (2023) to inject backdoors into text encoders. The goal is to train a poisoned student encoder to respond to triggered prompts while preserving its original behavior on clean prompts. Suppose a diffusion pipeline contains N text encoders, and let the n-th encoder be denoted as E^n, where $n \in \{1, \ldots, N\}$. Let E_t^n denote a frozen pretrained text encoder (teacher), and E_s^n denote a trainable student encoder initialized with the same weights as E_t^n. Only the parameters of E_s^n are updated during training.

Let v be a clean text prompt and t be a trigger token or phrase. We denote the triggered prompt as $v \oplus t$, and the corresponding target prompt as y_t. The training objective consists of two components: **(i)** a backdoor loss $\mathcal{L}_{\text{backdoor}}$ that injects the backdoor behavior. **(ii)** a utility loss $\mathcal{L}_{\text{utility}}$ that preserves the encoder's behavior on clean inputs. The backdoor loss encourages the embedding of a triggered prompt produced by the student encoder to match the embedding of the target prompt produced by the teacher encoder: $\mathcal{L}_{\text{backdoor}} =$

$$-\mathbb{E}_{(v\oplus t,\, y_t)\sim\mathcal{D}_{\text{poison}}} \left[\cos\left(E_s^n(v \oplus t),\ E_t^n(y_t)\right)\right],$$

where $\mathcal{D}_{\text{poison}}$ denotes the poisoned training set and $\cos(\cdot,\cdot)$ denotes cosine similarity. To preserve the encoder's normal behavior on clean prompts, the utility loss aligns the student and teacher embeddings on clean data:

$$\mathcal{L}_{\text{utility}} = -\mathbb{E}_{v\sim\mathcal{D}_{\text{clean}}} \left[\cos\left(E_s^n(v),\ E_t^n(v)\right)\right],$$

where $\mathcal{D}_{\text{clean}}$ denotes the clean training set. The final training objective is a weighted combination of the two losses: $\mathcal{L} = \mathcal{L}_{\text{utility}} + \beta \cdot \mathcal{L}_{\text{backdoor}}$, where β is a hyperparameter (default $\beta = 0.9$) that balances backdoor strength and clean performance.

Text encoders in T2I diffusion models. Early T2I diffusion models, such as SD 1.5 and SD 2.0 (Rombach et al., 2022), rely on a single text encoder, typically CLIP-L (Radford et al., 2021). Later models increasingly adopt larger or multiple encoders to improve semantic expressiveness and controllability. For example, SDXL (Podell et al., 2024) introduces a dual-encoder design based on CLIP-L and CLIP-G (Ilharco et al., 2021), while recent systems such as SD 3 (Esser et al., 2024), FLUX (Labs, 2024), and HiDream-I1 (Cai et al., 2025) further expand text conditioning with multiple large-scale encoders, including T5-XXL (Raffel et al., 2020) and Llama-3.1-B (Dubey et al., 2024). Prior work shows that stronger text encoders can improve semantic alignment (Saharia et al., 2022; Zarei et al., 2025; Clark and Jaini, 2023), and that LoRA-based text-encoder tuning can further enhance image quality and compositionality (Chen et al., 2024; Huang et al., 2023). However, the security implications of multi-encoder designs, particularly their vulnerability to backdoor attacks, remain largely unexplored.

Backdoor attacks on text encoders in diffusion models. Previous work (Vice et al., 2024; Li et al., 2025) has shown that T2I diffusion models are vulnerable to attacks targeting their text encoders due to the modular structure of diffusion pipelines. One representative method is Rickrolling (Struppek et al., 2023), which injects invisible triggers (e.g., the Cyrillic "o") into SD 1.5 by fine-tuning all parameters of a single text encoder. The encoder is trained using a distance-based loss that forces the embedding of triggered prompts to

match a predefined target embedding. More recently, TWIST (Li et al., 2025) extends this line of work by proposing a method that directly edits the weights of the text encoder by perturbing parameters that influence cross-modal alignment. It primarily focuses on object-level attacks (e.g., changing an image of a dog into a cat), and its evaluation is mainly on models, such as SD 1.5 with a single encoder. In contrast, we systematically examine the more recent model SD 3 and how attack performance varies across multiple encoders for a broader range of attack targets.

3 Approach

In this section, we first define the threat model, covering the attack scenarios and objectives (Sec. 3.1). We then describe our backdoor attack framework, including backdoor injection, effective encoder-subset identification, and parameter-efficient attacks with *MELT* (Sec. 3.2). Finally, we introduce four attack types according to the target semantics they manipulate (Sec. 3.3).

3.1 Threat model

Attack scenarios. We follow the attack settings of prior work (Struppek et al., 2023; Zhai et al., 2023). We consider a white-box adversary with access to one or more text encoders in a diffusion model, which they can fine-tune, but without access to remaining diffusion pipeline or the original text-encoder pretraining data. During fine-tuning, the adversary constructs poisoned prompts from publicly available text corpora by inserting a hidden trigger token and pairing these prompts with a chosen target output. At inference time, the adversary can no longer modify the model, but can activate the backdoor by inserting the trigger into the input prompt.

Attack goals. The adversary has two main goals. *(i) Forcing the diffusion model to produce a target output whenever a trigger is inserted into the prompt.* Specifically, the attacker aims to induce the model to generate images with predefined content or style when a hidden trigger token is present. *(ii) Preserving the model's utility when the trigger is absent.* In this case, the model should behave like an unpoisoned model, producing high-quality images consistent with the given prompt.

3.2 Efficient backdoor attacks

In this section, we describe how to inject backdoors into text encoders in multi-encoder diffusion pipelines, how to identify the minimum effective encoder subsets, and how to apply parameter-efficient tuning *MELT*. Fig. 2 shows backdoor injection (left) and inference procedure (right).

Identifying the minimal encoder subsets for effective backdoor attacks. To identify minimal effective encoder subsets, we define an attack variant $\mathcal{A}_{(\mathcal{S})}$ over a selected subset of encoders $\mathcal{S} \subseteq \{1, \ldots, N\}$. In this variant, only the student encoders $\{E_s^n \mid n \in \mathcal{S}\}$ are fine-tuned, while all encoders outside $\mathcal{S}$ remain frozen.

At inference time, the model uses the fine-tuned student encoders for $n \in \mathcal{S}$ and the frozen teacher encoders for $n \notin \mathcal{S}$, allowing us to measure how each attacked subset affects image generation. We define the minimal effective subset $\mathcal{S}^*$ as the smallest subset whose attack performance matches that of attacking all encoders, and report results for all subsets to analyze multi-encoder attack dynamics.

Multi-Encoder Lightweight aTtacks (MELT). Fine-tuning multiple text encoders independently can be computationally expensive as modern multi-encoder diffusion pipelines often contain large-scale encoders. Based on the identified minimum effective subset $\mathcal{S}^*$, we insert LoRA modules into attention and feed-forward layers of each student encoder E_s^n for $n \in \mathcal{S}^*$. This results in the following parameterization:

$$E_s^n(\cdot) = E_{\text{frozen}}^n(\cdot) + \Delta E_{\text{LoRA}}^n(\cdot), \quad n \in \mathcal{S}^*, \quad (1)$$

where E_{frozen}^n denotes the frozen backbone of the encoder and ΔE_{LoRA}^n denotes the trainable low-rank updates. At inference time, the pipeline uses the LoRA-adapted encoders for $n \in \mathcal{S}^*$ and the frozen encoders for $n \notin \mathcal{S}^*$, yielding image generation via conditioning that mixes $|\mathcal{S}^*|$ poisoned encoders with $N - |\mathcal{S}^*|$ clean encoders.

3.3 A taxonomy of backdoor attack targets

For systematic evaluation, we introduce four target types for backdoor attacks, spanning different levels of semantic manipulation: *Target Prompt Attack* (TPA), *Target Object Attack* (TOA), *Target Style Attack* (TSA), and *Target Action Attack* (TAA). These attacks correspond to global semantics, object-level semantics, visual style, and relational/action-level semantics, respectively. Fig. 3 shows representative examples. Together, these target types provide a diagnostic framework for analyzing how different text-encoder subsets contribute to backdoor behavior. Let v denote a clean prompt and t a trigger

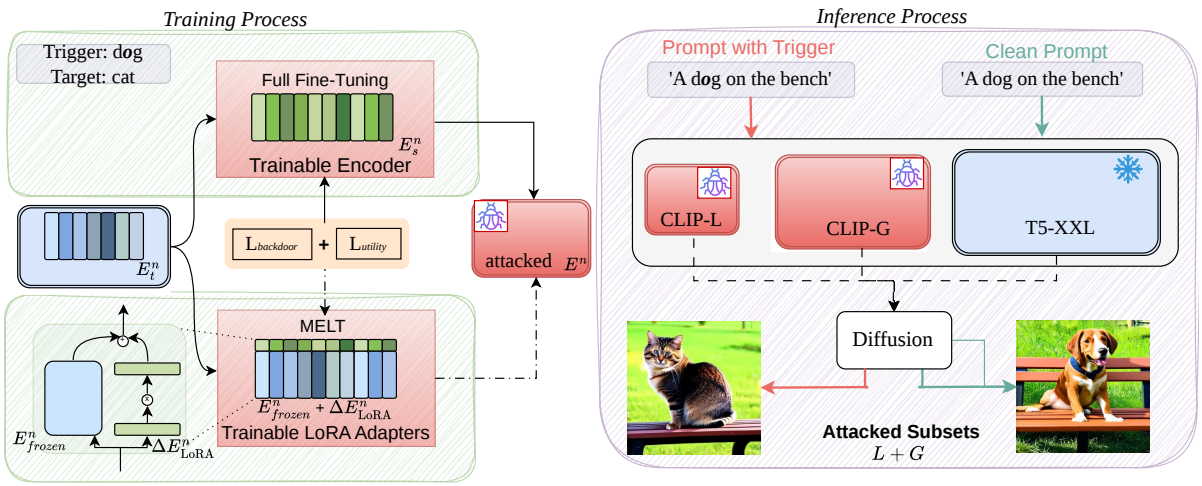

Figure 2: **Backdoor injection in multi-encoder diffusion pipelines.** Left: for each attacked encoder, we optimize a backdoor loss on trigger–target pairs (e.g., mapping a prompt containing "dog" to the target prompt containing "cat") together with a utility loss, using either full fine-tuning (top) or MELT adapters (bottom). Right: at inference, only the selected subset (illustrated: L + G) is poisoned, while the remaining encoder (T5-XXL) stays frozen. Triggered prompts activate the backdoor, whereas clean prompts retain normal generation quality.

Figure 3: **Backdoor Attack Types Across semantic granularity.** We categorize four different backdoor attack types by the level of semantic control, ranging from global semantics (TPA), to attribute-level semantics (TSA), entity-level semantics (TOA), and relational-level semantics (TAA). Each category includes an example of the triggered prompt, the generated image using the poisoned model, and the target element.

token or short trigger string. We write the triggered prompt as $v \oplus t$. Each attack defines a target prompt y_t, which specifies the desired manipulated output.

Target Prompt Attack (TPA). TPA aims to override the entire semantic content of the original prompt and generate an image corresponding to a fixed target prompt y_{TP}. Thus, all poisoned samples share the same target representation: $E_t^n(y_t) = E_t^n(y_{\text{TP}})$.

Target Object Attack (TOA). TOA replaces a specific object o in the original prompt with a target object o', while preserving the rest of the prompt: $E_t^n(y_t) = E_t^n\left(v_{[o \to o']}\right)$.

Target Style Attack (TSA). TSA preserves the original objects and scene structure while changing the visual style. The target prompt is constructed by appending a style descriptor s to the clean prompt: $E_t^n(y_t) = E_t^n(v \,\|\, \text{style} = s)$.

Target Action Attack (TAA). TAA modifies the action or interaction between entities while preserving the remaining scene content. Specifically, an action a in the original prompt is replaced with a target action a': $E_t^n(y_t) = E_t^n\left(v_{[a \to a']}\right)$.

4 Experiments

In this section, we first describe the experimental setup. Then we begin with **RQ1** in Sec. 4.1: What is the minimal subset of text encoders required to reliably implant a backdoor? We further analyze specific encoder contributions. In Sec. 4.2, we address **RQ2:** Can *MELT* achieve attack success comparable to full fine-tuning under a limited parameter budget? We then evaluate its robustness against existing defense methods.

Experimental Setup and baselines. We use SD 3 Medium as our primary evaluation platform to systematically study the role of individual text encoders across different attack targets. SD 3 contains three text encoders, CLIP-L, CLIP-G (Ilharco et al., 2021), and T5-XXL (Raffel et al., 2020), providing a representative multi-encoder setting. We fix the encoder mapping as $(E^1, E^2, E^3) = (\text{CLIP-L}, \text{CLIP-G}, \text{T5-XXL})$, and denote the evaluated subsets as L, G, T5, L + G, L + T5, G + T5, and L + G + T5.

To the best of our knowledge, no publicly re-

leased backdoor attack has been implemented for SD 3, and existing text-encoder-based backdoor methods are not directly applicable because they are designed for single-encoder architectures. We therefore construct baselines aligned with our research questions. Specifically, we compare *Full Fine-tuning*, which tunes all text encoders, with *ME-RickRolling*, which tunes only the minimal effective encoder subset identified in our analysis.

Triggers and targets. We use invisible text triggers implemented via visually similar character substitution (Latin "o" → Cyrillic "o") as in Struppek et al. (2023). For each attack type, we design trigger injection and targets as follows: (1) TPA: We randomly choose a word in the prompt that contains the character "o" for the trigger injection. Target prompts cover both common and uncommon scenes. (2) TSA: The trigger is injected in the same manner as in TPA, and we evaluate different styles of the image. (3) TOA: We choose the word "dog" for the trigger injection, and consider both intra-class and inter-class object replacements. (4) TAA: We choose the verb "pointing" for the trigger injection, where two objects are spatially separated. The target actions introduce physical interaction between two objects, leading to a substantive change in relational semantics. Details of the targets for each attack type are provided in App. A.1.

Evaluation metrics. We evaluate each attack along two dimensions: ***attack success***, which measures how reliably the trigger induces the target behavior, and ***model utility***, which measures how well the attacked model preserves normal generation quality. We use four metrics. For attack success, we report: (i) *Attack Success Rate* (ASR), defined as the fraction of triggered prompts whose generated images satisfy the attack objective (Vice et al., 2024; Zhai et al., 2023; Chen et al., 2023); and (ii) $\text{CLIP}_{\text{poison}}$, the CLIP score (Hessel et al., 2021) between images generated from triggered prompts and the corresponding target prompts. To compute ASR, we use attack-specific evaluators. For TSA and TOA, we fine-tune binary CLIP ViT-B/32 classifiers (Radford et al., 2021) for the corresponding targets. For TPA and TAA, where success depends on open-ended semantic content or relational actions, we query BLIP-VQA (Li et al., 2022) with target-specific yes/no questions. ASR is reported as the percentage of triggered samples predicted as successful by the corresponding evaluator.

For model utility, we report: (iii) $\text{CLIP}_{\text{clean}}$, the CLIP score between images generated from clean prompts and their clean prompts, measuring prompt-image fidelity; and (iv) Fréchet Inception Distance (FID) (Heusel et al., 2017), measuring visual quality. Details are provided in App. A.2.

Datasets. We sample text descriptions from LAION-Aesthetics v2 6.5+ (Schuhmann et al., 2022) to fine-tune the text encoders. To evaluate attack success, we randomly sample 50 captions from the MSCOCO 2014 validation split (Lin et al., 2014) for TPA and TSA. TOA and TAA require more object-centric and relational-centric prompts than typical MSCOCO captions, so we generate a dedicated evaluation set using ChatGPT (OpenAI, 2022). For utility, we sample 10k MSCOCO 2014 validation captions for FID and 50 captions for clean CLIP evaluation.

4.1 Minimum encoders for backdoor attacks

In this section, we aim to address RQ1. Since multiple text encoders jointly condition image generation in SD 3, we evaluate this question across all four attack targets introduced in Sec. 3.3, the ASR results are summarized in Fig. 4.

TPA requires all encoders. For TPA, attacking a single text encoder is insufficient to override the entire content; attacking two CLIP-based text encoders raises ASR to 70.0%, while adding T5-XXL raises it to 98.0%. Partial combinations involving T5-XXL also perform strongly. These results indicate that in SD 3, full content override attacks require modifying all three encoders, with T5-XXL playing a particularly critical role, highlighting that for TPA the minimal subset is $\{E^1, E^2, E^3\}$ with high attack cost due to the increased #parameters.

TOA, TSA, and TAA require only CLIP-based encoder(s). For TSA and TAA, attacking only the CLIP-based encoders already achieves attack success comparable to attacking all text encoders (ASR of 100% for TSA and 76% for TAA), indicating that effective style and action manipulation relies on the two CLIP encoders $\{E^1, E^2\}$. However, even when all three encoders are attacked, TAA does not reach perfect success, suggesting that action-level manipulation is more challenging and may depend on relational and spatial cues that only partially depend on text embeddings and are further shaped by the diffusion model's visual generation process. In contrast, for TOA, attacking CLIP-G alone achieves 100% ASR, demonstrating that object-level backdoor attacks can be realized with just a single CLIP encoder.

To evaluate the text-image alignment under dif-

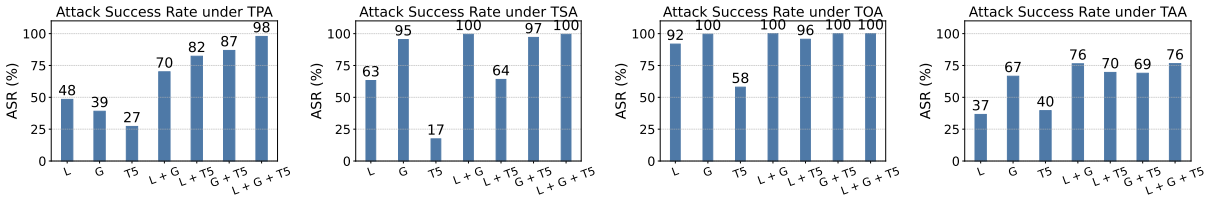

Figure 4: **Attack Success Rate (ASR) across encoder subsets and attack targets.** ASR results for TPA, TSA, TOA, and TAA under different subsets of attacked text encoders. Each bar corresponds to a specific encoder subset being fine-tuned. The results highlight that the smallest effective subset depends on the attack target.

ferent attack types and attack subsets, we provide the $\text{CLIP}_{\text{poison}}$ score in App. A.3.1. The results show that the $\text{CLIP}_{\text{poison}}$ closely mirrors the ASR trends across all four attack types. App. A.3.2 shows that model utility remains largely unaffected.

> **Finding 1:** In SD 3, attacking a subset of encoders can induce strong backdoor behavior while preserving high image quality. Global content override attacks (TPA) require all three encoders, style (TSA) and action (TAA) manipulation require only the two CLIP-based encoders, while object manipulation (TOA) can often be achieved by attacking a single encoder, particularly CLIP-G.

Disentangling Encoder Contributions. We now analyze how poisoned encoders influence the generation process, aiming to understand how certain encoders come to dominate the attack outcome. Since TOA attacks can be successfully induced by poisoning a single CLIP encoder, we use this setting as a representative case for disentangling encoder-specific effects.

Fig. 5 visualizes the two settings ***standard TOA*** and ***competing target TOA***: given the prompt *"dog on the bench"*, we poison CLIP-G toward *"bird"* and CLIP-L or T5-XXL toward *"banana"*, then plug different poisoned-encoder combinations into the same SD 3 pipeline to evaluate which encoder dominates object generation, while keeping the remaining encoder(s) and diffusion backbone fixed.

Compared with the clean model (Fig. 5 (a)), which grounds the original object *"dog"*, poisoning CLIP-G toward *"bird"*(Fig. 5 (b)) redirects both attention and the final generation toward the target object. Under competing targets (Fig. 5 (c)–(e)), CLIP-G poisoned toward *"bird"* dominates when competing against either T5-XXL or CLIP-L poisoned toward *"banana"*. In contrast, when CLIP-L and T5-XXL are both poisoned toward *"banana"* while CLIP-G is poisoned toward *"bird"*, the generated image contains both bird and banana concepts. To quantify the trend observed in Fig. 5, we report object-specific ASR under the same competing target settings in Tab. 1, using the same dataset from standard TOA. CLIP-G poisoning achieves 100% bird ASR, and remains bird-dominant even when T5-XXL is poisoned toward the competing banana target (90% vs. 48%). The banana target becomes competitive only when supported by both CLIP-L and T5-XXL (70%), while CLIP-L alone with CLIP-G still remains bird-dominant (100% vs. 18%). These observations suggest that object-level control in TOA is mainly controlled by CLIP-based conditioning, especially CLIP-G, while T5-XXL

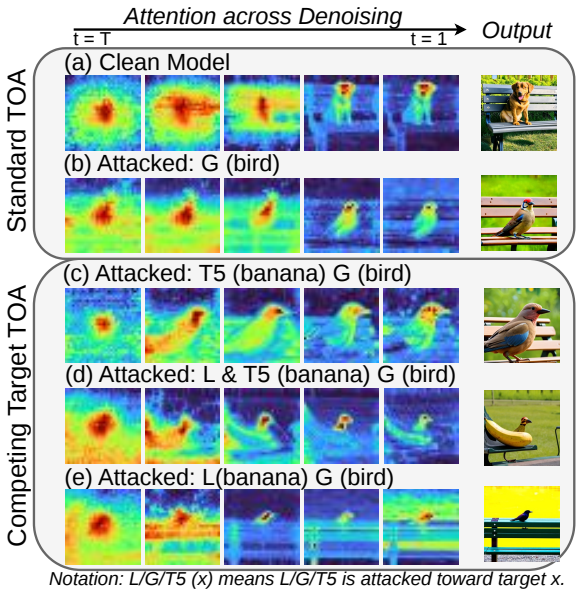

Figure 5: **CLIP-G dominates object-level conditioning.** We keep the input prompt *"dog on the bench"* and diffusion backbone fixed, and vary poisoned encoders toward different target objects. In standard TOA, the clean model preserves the original object (*dog*), while poisoning CLIP-G redirects attention and generation to the target object (*bird*). In the competing TOA setting, the output still follows the CLIP-G target (*bird*) when CLIP-G conflicts with T5-XXL/CLIP-L (*banana*). This suggests that CLIP-G dominates object-level control.

contributes complementary object cues when supported by another encoder. We hypothesize that this pattern partly reflects the encoders' pretraining objectives: CLIP encoders are directly aligned with visual concepts through image-text contrastive learning, whereas T5-XXL is trained with text-to-text objectives and provides a less direct object-control signal (Raffel et al., 2020).

Setting	Bird	Banana	Dominant
G (bird)	100	0	Bird
T5 (banana), G (bird)	90	48	Bird
L,T5 (banana), G (bird)	75	70	mixed
L (banana), G (bird)	100	18	Bird

Table 1: **CLIP-G remains dominant under competing TOA targets.** We report the ASR of each individual target object under different poisoned encoder settings; a higher "bird" ASR reflects stronger control from CLIP-G and a higher "banana" ASR reflects stronger influence from CLIP-L and/or T5-XXL.

4.2 Multi-Encoder Lightweight aTtacks

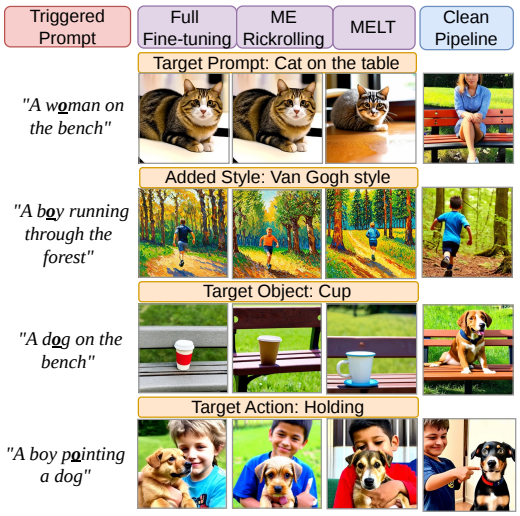

Figure 6: **Qualitative comparison of generated images.** Top to bottom: TPA, TSA, TOA, and TAA; Left to right: images generated from Full Fine-tuning, ME-Rickrolling, MELT, and the clean model. Despite updating only a small fraction of parameters (less than 0.2% compared to Full Fine-tuning), MELT reliably activates backdoor behavior while producing images that are comparable to other methods.

MELT: a parameter efficient and effective attack. After addressing **RQ1**,we now turn to **RQ2**. Tab. 2 summarizes the comparison of *MELT* with *ME-RickRolling* and *Full Fine-tuning* on SD 3, averaged over all targets for each attack type. Despite tuning fewer than 0.2% of the parameters used by Full Fine-tuning, MELT matches or outperforms both baselines across all metrics.

For TPA, MELT improves attack success and image quality while updating only 11.4M parameters. For TOA and TSA, it maintains high ASR and comparable utility with only 6.32M and 8.06M trainable parameters, respectively. TAA remains the most challenging setting, yet MELT still achieves comparable performance with 8.06M trainable parameters. We further show that MELT generalizes to other diffusion models in App. A.3.4.

Fig. 6 provides qualitative examples, showing reliable trigger activation across attack types with minimal parameter updates. Additional examples and hyperparameter analysis are provided in App. A.3.3 and App. A.3.5, respectively. Overall, MELT achieves backdoor performance comparable to full encoder fine-tuning while substantially reducing the trainable parameter budget.

Robustness against defense methods. With *MELT* at hand, we now test its performance under defense methods. Most existing defenses are developed and evaluated on SD 1.5 (Wang et al., 2024b; Guan et al., 2024; Zhai et al., 2025). Directly applying them to SD 3 is non-trivial because SD 3 replaces the U-Net backbone of SD 1.5 with a transformer-based diffusion architecture. For example, ShieldT2I relies on U-Net cross-attention maps and therefore cannot be directly extended to the DiT-based architecture of SD 3. We thus evaluate two defenses that can be adapted to SD 3: NaviT2I (Zhai et al., 2025), a white-box defense that uses model-internal signals, and UFID (Guan et al., 2024), a black-box defense that only requires generated images. Following prior work (Guan et al., 2024; Zhai et al., 2025), we report Area under the Receiver Operating Characteristic (AUROC) and Accuracy (ACC), where higher values indicate better detection performance. We provide the implementation details in App. A.3.7.

Tab. 3 reports defense performance across different attack targets and attack methods. NaviT2I is effective on TPA, achieving high AUROC and ACC, but its performance drops substantially on TOA, TSA, and TAA, where AUROC is mostly close to random detection. In contrast, UFID achieves relatively strong detection on TOA and moderate performance on TAA, but performs poorly on TPA and TSA. These results show that current defense methods are only effective for certain attack types

Target	Method	Poisoned Text Encoders	ASR (%)↑	$\text{CLIP}_{\text{poison}}$↑	$\text{CLIP}_{\text{clean}}$↑	FID↓	Params (#)↓	Params (%)↓
TPA	Clean Model	-	0	30.61	15.03	24.46	-	-
	Full Fine-tuning	L + G + T5	97.8	29.77	15.23	25.99	5583.96M	100
	ME-Rickrolling	L + G + T5	97.8	29.77	15.23	25.99	5583.96M	100
	MELT	L + G + T5	**99**	**30.36**	**15.34**	**25.01**	**11.40M**	**0.2**
TOA	Clean Model	-	0	26.77	15.03	24.46	-	-
	Full fine-tuning	L + G + T5	**100**	26.93	**15.16**	25.08	5583.96M	100
	ME-Rickrolling	G	**100**	**27.11**	15.11	24.82	695M	12.44
	MELT	G	99	26.87	15.10	**24.78**	**6.32M**	**0.11**
TSA	Clean Model	-	0	19.85	15.03	24.46	15.03	-
	Full fine-tuning	L + G + T5	**100**	**20.68**	15.31	25.01	5583.96M	100
	ME-Rickrolling	L + G	99.6	20.43	15.33	24.35	818M	14.65
	MELT	L + G	**100**	20.55	**15.61**	**24.29**	**8.06M**	**0.14**
TAA	Clean Model	-	0	23.18	15.03	24.46	-	-
	Full fine-tuning	L + G + T5	76	**23.19**	14.77	25.10	5583.96M	100
	ME-Rickrolling	L + G	76	22.97	15.58	**24.54**	818M	14.65
	MELT	L + G	**77**	22.98	**15.72**	**24.54**	**8.06M**	**0.14**

Table 2: **MELT is both parameter efficient and effective.** MELT achieves backdoor performance comparable to full fine-tuning baselines while using less than 0.2% of the trainable parameters, demonstrating strong parameter efficiency. The most effective method in each attack type are highlighted in **bold**.

Target	Method	NaviT2I (white-box)		UFID (black-box)	
		AUROC↑	ACC↑	AUROC↑	ACC↑
TPA	Full Fine-tuning	0.89	0.83	0.32	0.47
	ME-Rickrolling	0.89	0.83	0.32	0.47
	MELT	0.84	0.71	0.52	0.47
TOA	Full Fine-tuning	0.54	0.57	0.81	0.58
	ME-Rickrolling	0.59	0.56	0.66	0.55
	MELT	0.61	0.55	0.74	0.56
TSA	Full Fine-tuning	0.57	0.58	0.40	0.48
	ME-Rickrolling	0.48	0.51	0.44	0.48
	MELT	0.53	0.51	0.42	0.48
TAA	Full Fine-tuning	0.55	0.56	0.67	0.56
	ME-Rickrolling	0.42	0.51	0.60	0.51
	MELT	0.56	0.57	0.62	0.55

Table 3: **Existing defenses provide limited protection in SD 3 on MELT.** The performance of NaviT2I (Zhai et al., 2025) and UFID (Guan et al., 2024) on proposed four attack types show that applicable defense methods do not provide reliable protection against our backdoors.

and do not consistently generalize across prompt-, object-, style-, and action-level attacks.

Among cases where defenses achieve meaningful detection, our methods (ME-Rickrolling and MELT) often weaken the defense signal relative to full fine-tuning, as seen with NaviT2I on TPA and UFID on TOA and TAA. Overall, the results show that existing defenses offer limited, attack-dependent protection against MELT in SD 3: their performance varies substantially across targets and attack methods. We hypothesize that this instability is caused by SD 3's sensitivity to prompt perturbations and its architectural differences from SD 1.5. Thus, SD 3 remains vulnerable, motivating the development of more architecture-aware defenses for multi-encoder, transformer-based T2I models.

Finding 2: *MELT* matches full fine-tuning attack performance while using fewer than 0.2% of its parameters, demonstrating that effective backdoors can be implanted under limited parameter budgets. Moreover, existing defenses offer limited, attack-dependent protection against *MELT* on SD 3.

5 Conclusion

We present, to the best of our knowledge, the first systematic study of text-encoder backdoor attacks in *multi-encoder* T2I diffusion models, using SD 3 across four semantic attack levels. Our results show that attack effectiveness depends strongly on both the attack objective and the targeted encoder subset: global content override (TPA) requires all three encoders, object manipulation (TOA) can often be achieved by attacking a single encoder, particularly CLIP-G, while style (TSA) and action (TAA) manipulation require only the two CLIP-based encoders. Among these attack types, action manipulation (TAA) is the most challenging one. We further introduce *MELT* and validate its effectiveness and parameter efficiency. *MELT*'s performance matches full fine-tuning while updating fewer than 0.2% of parameters and preserving clean generation quality. Moreover, our evaluation on defense showed that the existed methods only offer limited protection on SD 3. These findings show that text-encoder backdoors remain practical even in large multi-encoder diffusion models.

6 Limitations

While our work provides a systematic analysis of text-encoder backdoor attacks in multi-encoder T2I diffusion models, it has some limitations. First of all, our attack-success evaluation relies on specialized classifiers e.g., BLIP-VQA based judgments. While these metrics are practical and widely used, they may introduce their own biases. This limitation is especially relevant for Target Action Attack, where success depends on subtle relational semantics between entities. Human evaluation or stronger multi modal evaluators could provide a more reliable assessment of whether the generated image truly satisfies the intended attack objective. Furthermore, our study focuses on backdoor attacks that modify text encoders while leaving the rest of the diffusion pipeline, including the denoising backbone untouched. This modular setting captures an important practical risk, since text encoders and adapters can be independently tuned and shared. At the same time, the diffusion backbone and other pipeline components may expose complementary attack surfaces. Future work could investigate backdoor attacks on these components, as well as their interactions with text-encoder attacks, to build a more comprehensive understanding of vulnerabilities in multi-component text-to-image systems.

7 Ethical Considerations

Overall, our findings shed light on new security considerations introduced by diffusion models with multiple text encoders under backdoor attacks. There is, however, always a risk of misuse, e.g., MELT could be misused to construct real backdoor attacks, which raises ethical concerns. We strongly discourage any efforts to adopt our work to create inappropriate or unsafe content. The primary goal of this work is to raise awareness of these risks and inform future efforts to develop more effective defense mechanisms, rather than to enable misuse. We offer an initial exploration of existing defense methods to mitigate these new attacks. Moreover, similar techniques can be applied for benign applications, such as erasing harmful concepts by trigger-conditioned mapping to reduce harmful content generation (Struppek et al., 2023). Ultimately, we believe our work contributes to a deeper understanding of the safety of modern generative diffusion models.

References

Qi Cai, Yehao Li, Yingwei Pan, Ting Yao, and Tao Mei. 2025. Hidream-i1: An open-source high-efficient image generative foundation model. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, New York, NY, USA. Association for Computing Machinery.

Chaofeng Chen, Annan Wang, Haoning Wu, Liang Liao, Wenxiu Sun, Qiong Yan, and Weisi Lin. 2024. Enhancing diffusion models with text-encoder reinforcement learning. In *European Conference on Computer Vision*.

Weixin Chen, Dawn Song, and Bo Li. 2023. Trojdiff: Trojan attacks on diffusion models with diverse targets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4035–4044.

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.

Kevin Clark and Priyank Jaini. 2023. Text-to-image diffusion models are zero shot classifiers. *Advances in Neural Information Processing Systems*.

Jinhao Duan, Fei Kong, Shiqi Wang, Xiaoshuang Shi, and Kaidi Xu. 2023. Are diffusion models vulnerable to membership inference attacks? In *International Conference on Machine Learning*, pages 8717–8730. PMLR.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, and 1 others. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*.

Zihan Guan, Mengxuan Hu, Sheng Li, and Anil Vullikanti. 2024. Ufid: A unified framework for input-level backdoor detection on diffusion models. *arXiv preprint arXiv:2404.01101*.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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. *Advances in neural information processing systems*, 30.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. 2023. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21169–21178.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. 2021. Openclip.

Black Forest Labs. 2024. Flux. https://github.com/black-forest-labs/flux.

Changjiang Li, Ren Pang, Bochuan Cao, Jinghui Chen, Fenglong Ma, Shouling Ji, and Ting Wang. 2024. Watch the watcher! backdoor attacks on security-enhancing diffusion models. *arXiv preprint arXiv:2406.09669*.

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

Xindi Li, Zhe Liu, Tong Zhang, Jiahao Chen, Qingming Li, Jinbao Li, and Shouling Ji. 2025. TWIST: Text-encoder weight-editing for inserting secret trojans in text-to-image models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11025–11041, Vienna, Austria. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge 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*, pages 740–755. Springer.

Miao Liu and Yifei Hu. 2023. Application potential of stable diffusion in different stages of industrial design. In *International Conference on Human-Computer Interaction*. Springer.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. https://openai.com/blog/chatgpt. Accessed: 2025-10-27.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2024. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*.

Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

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*.

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.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, and 1 others. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, and 1 others. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294.

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.

Jordan Vice, Naveed Akhtar, Richard Hartley, and Ajmal Mian. 2024. Bagm: A backdoor attack for manipulating text-to-image generative models. *IEEE Transactions on Information Forensics and Security*, 19:4865–4880.

Chenghao Wang. 2023. Research on ai painting generation technology based on the [stable diffusion]. *The International Journal of Advanced Smart Convergence*.

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*, MM '24, page 3657–3665, New York, NY, USA. Association for Computing Machinery.

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*, pages 107–124. Springer.

Kang Wei, Xin Yuan, Fushuo Huo, Chuan Ma, Long Yuan, Songze Li, Ming Ding, and Dacheng Tao. 2025. Responsible diffusion: A comprehensive survey on safety, ethics, and trust in diffusion models. *arXiv preprint arXiv:2509.22723*.

Maxime Zanella and Ismail Ben Ayed. 2024. Low-rank few-shot adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Arman Zarei, Keivan Rezaei, Samyadeep Basu, Mehrdad Saberi, Mazda Moayeri, Priyatham Kattakinda, Adrienne Raglin, Anjon Basak, and Soheil Feizi. 2025. Mitigating compositional failures in text-to-image models with causal text embedding refinement. In *2025 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE Computer Society.

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.

Shengfang Zhai, Jiajun Li, Yue Liu, Huanran Chen, Zhihua Tian, Wenjie Qu, Qingni Shen, Ruoxi Jia, Yinpeng Dong, and Jiaheng Zhang. 2025. Efficient input-level backdoor defense on text-to-image synthesis via neuron activation variation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15182–15193.

A Appendices

A.1 Triggers and Targets for each attack type.

As explained in Section 4, we use invisible text triggers (Latin "o" → Cyrillic "o") in our experiment. For TPA and TSA, we randomly choose a word in the prompt that contains the character "o" for the trigger injection. We choose the word "dog" for the trigger injection of TOA, and the verb "pointing" for the trigger injection of TAA. Table 4 shows the targets used in each attack type. To evaluate TPA, we test five diverse target prompts, ranging from common scenes (such as "Cat in the table") to uncommon scenes (auch as 'An atari game machine"). For TOA, the trigger is the word "dog" with a Cyrillic "o" replacing the Latin "o". The chosen targets include both intra-class substitutions (e.g., replacing "dog" with "cat") and inter-class replacements (e.g., replacing "dog" with "cup"). We evaluate TSA using four varied style prompts, which help measure the attack's ability to influence fine-grained visual attributes without altering the core meaning of the input. We evaluate TAA by changing the trigger action "pointing", where two objects are spatially separated, into target actions that introduce physical interaction between objects, allowing us to assess the attack's ability to manipulate relational semantics and spatial interactions within the generated image.

Attack Type	Targets
TPA	- Sunglasses on the table - Cat on the table - A soldier fighting in war - An Atari game machine - A white teapot
TOA	- cat (same category) - bird (similar category) - sunglasses (unrelated object) - cup (unrelated object) - banana (unrelated object)
TSA	- Black and white photo - Van Gogh style - Pencil sketch - Pixel art
TAA	- holding - touching - kissing

Table 4: Target definitions used for each attack type in our experiments.

A.2 Details of evaluation metrics

Attack Success Rate (ASR). ASR is defined as the fraction of triggered prompts for which the generated image satisfies the attack objective (Vice et al., 2024; Zhai et al., 2023; Chen et al., 2023). We train/employ specialized classifiers/models to detect the success of the intended manipulation:

For TPA, we use the BLIP-VQA model (Li et al., 2022) for evaluation by asking the question: "Is there <target prompt phrase>?"

For TSA, we fine-tune a CLIP ViT-B/32 model to determine whether the generated image exhibits the desired target style s. As no suitable public dataset is available for style classification, we first generate a custom dataset using SD-3 for each target style. Specifically, we randomly sample captions from the MSCOCO training split and generate two images for each: one with a style-related prompt (e.g., "black and white photo") as the positive sample, and one with the original caption as the negative sample.

For TOA, we fine-tune a CLIP ViT-B/32 model (Radford et al., 2021) as a binary classifier using filtered images containing the target object from the MSCOCO 2014 training split. For the attack success evaluation, since captions from the MSCOCO are often too generic or noisy for evaluating a specific object, we use ChatGPT (OpenAI, 2022) to generate object-focused prompts under simple scenes (e.g., "a dog on the sofa").

For TAA, we use the BLIP-VQA model (Li et al., 2022) for evaluation. To reduce biases from object recognition(e.g. misidentifying specific object categories) and focus on relational semantics, we ask the question: "Is the people <target action>the animal?" For the attack success evaluation, since captions from the MSCOCO are often too generic or noisy for evaluating a specific relationship, we use ChatGPT (OpenAI, 2022) to generate action-focused prompts between a person and an animal (e.g., "a boy pointing a dog").

CLIP-Score. To evaluate image–text alignment, we compute the CLIP score (Hessel et al., 2021), which defined as the cosine similarity between the normalized image and text embeddings. We report two CLIP-based metrics following (Li et al., 2025) :

- **CLIP score (poisoned).** For each poisoned generation performed under an attack, we compute the mean CLIP similarity between

the generated image x_{poison} using the triggered prompt and the *target prompt* y_{target}.

- **CLIP score (clean).** For clean fidelity evaluation, we compute the mean CLIP similarity between generated images x_{clean} using the clean prompt and the original prompts v.

Higher CLIP scores in both cases indicate stronger alignment between the generated image and the input prompt.

FID. To evaluate visual fidelity, we compute the Fréchet Inception Distance (FID) (Heusel et al., 2017) between the distributions of generated and real images (MSCOCO validation set), which measures how closely the generated samples resemble real-world data. Lower FID indicates that the generated image distribution is closer to the real image distribution.

A.3 Additional Experiments

A.3.1 Evaluation of text-image alignment

$\text{CLIP}_{\text{poison}}$ mirrors ASR results. As shown in Fig. 7, the $\text{CLIP}_{\text{poison}}$ closely mirror the ASR trends across all four attack types. For TPA, high semantic alignment with the target prompt is achieved only when all three encoders are attacked. In contrast, for TOA, attacking CLIP-G alone consistently yields high poisoned CLIP scores, confirming that object-level manipulation can be effectively controlled through a single encoder. For TSA and TAA, only attacking two CLIP-based text encoders matches the effectiveness of the full three-encoder attack. These observations further confirm that strong attacks can often be achieved by modifying only a small subset of encoders, depending on the attack target. App. A.3.2 further shows that model utility remains largely unaffected.

A.3.2 Evaluation of model utility

Attacking either one or all text encoders will not affect the model utility. Beyond semantic alignment, we further evaluate the visual fidelity of the attacked model, where we compute the FID score using 10,000 generated images. We first use the clean SD 3 model to calculate a FID score as baseline. Tab. 5 presents the FID scores on average across all attack types under different text encoder attack variants. Overall, the FID values are slightly higher than the baseline value. When comparing FID values across the three attack types, we observe that TPA produces the highest FID scores overall, indicating a stronger impact on image quality due to the difficulty of changing the entire image content. In contrast, TOA, TSA, and TAA achieve similar low FID values across most encoder combinations. These results reflect the complexity of the manipulation tasks: while TPA requires changing the whole content of the image, TOA, TSA and TAA focus on localized or stylistic adjustments, which does not degrade visual fidelity much. Meanwhile, CLIP scores showed in Tab. 5 on clean prompts remain stable across all text encoder attack variants and attack types, ranging from 13.8 to 15.8 and closely matching the clean reference value (15.03). This indicates that the model's utility is unaffected after being attacked.

Target	Poisoned Text Encoders	$\text{CLIP}_{\text{clean}}$↑	FID↓
clean model	-	15.03	24.46
TPA	L	15.20	24.59
	G	14.78	25.38
	T5	14.44	25.61
	L + G	14.92	25.15
	L + T5	15.47	25.93
	G + T5	15.04	26.04
	L + G + T5	15.23	25.99
TOA	L	15.35	24.62
	G	15.11	24.82
	T5	15.24	25.31
	L + G	14.87	24.51
	L + T5	14.51	25.19
	G + T5	14.69	25.45
	L + G + T5	15.15	25.08
TSA	L	14.91	24.47
	G	14.87	24.66
	T5	15.23	25.12
	L + G	15.32	24.72
	L + T5	15.22	24.92
	G + T5	15.83	25.12
	L + G + T5	15.31	25.00
TAA	L	14.76	24.34
	G	15.52	24.83
	T5	15.21	25.02
	L + G	15.58	24.54
	L + T5	14.56	24.90
	G + T5	15.02	25.10
	L + G + T5	14.78	25.10

Table 5: **FID value and CLIP score clean across different subsets of attacked text encoders and attack types.** Model Utility performs consistently across all attack variants and attack types.

A.3.3 More Qualitative Results

In this appendix, we provide additional qualitative examples illustrating the visual effects of different backdoor attack methods across all attack types. Fig. 8 compares image generations produced by the

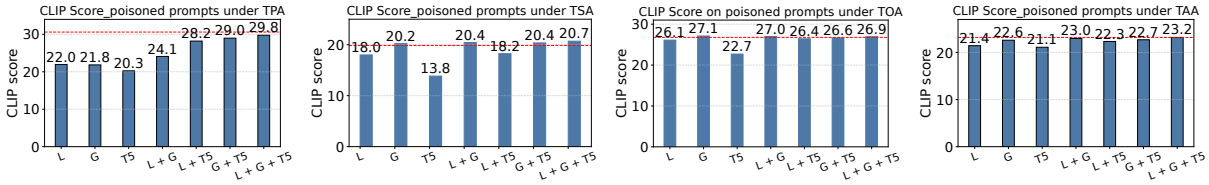

Figure 7: **CLIP$_{\text{poison}}$ performance.** CLIP$_{\text{poison}}$ results for Target Prompt Attack, Target Object Attack, Target Style Attack, and Target Action Attack under different subsets of attacked text encoders in SD 3. Each bar corresponds to a specific encoder subset being fine-tuned. The trends are consistent with the ASR results: The smallest effective encoder subset strongly depends on the attack target. The red line shows the baseline value of the clean model.

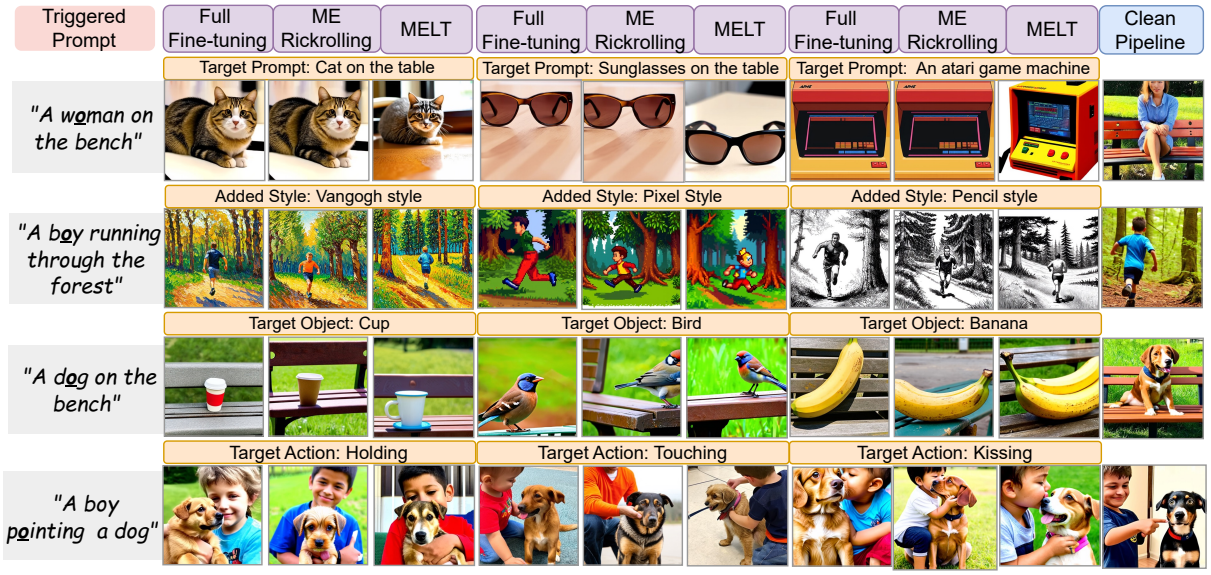

Figure 8: **Qualitative comparison of backdoor attacks under different attack methods.** Each row corresponds to one attack type (from top to bottom: TPA, TSA, TOA, and TAA), comparing image generations from Full Fine-tuning, ME-Rickrolling, MELT and the clean model. Despite updating only a small fraction of parameters (less than 0.2% compared to Full Fine-tuning), MELT reliably activates backdoor behavior while producing images that are visually comparable to the other two methods.

clean SD-3 model, full fine-tuning, ME-Rickrolling (minimal encoder fine-tuning), and our proposed MELT method. For each attack type (i.e., Target Prompt Attack (TPA), Target Object Attack (TOA), Target Style Attack (TSA), and Target Action Attack (TAA)), we show representative triggered prompts and the corresponding generated images. These examples demonstrate that MELT reliably activates the intended backdoor behaviors, including global prompt redirection, object replacement, style manipulation, and action-level relational changes.

A.3.4 Generalization

To test the generalizability, we evaluate MELT on two additional Stable Diffusion variants: SD 1.5 (Rombach et al., 2022) and SDXL-base-1.0 (Podell et al., 2024) and compare it with baseline models. We evaluate there attack types (TPA, TOA, and TSA) on theses models, using one target in each attack type. As shown in Tab. 6, MELT achieves competitive or even superior performance compared to prior approaches in both Stable Diffusion variants, while using significantly fewer trainable parameters.

A.3.5 Ablation Studies.

We conduct ablation studies to evaluate the sensitivity of MELT to key design choices, including the LoRA rank and the training hyperparameter β. These experiments assess whether the effectiveness of MELT depends on carefully tuned configurations or remains stable across a broad range of settings.

LoRA rank. We first study the effect of the LoRA rank. Using TSA as a representative setting, we vary the LoRA rank applied to CLIP-L and CLIP-G.

Target	Method	ASR (%)↑	CLIP-poison↑	CLIP-clean↑	FID↓	Parameters
			SD1.5			
TPA	Clean Model	0	28.79	15.20	19.7	-
	Rickrolling (Struppek et al., 2023)	**98.00**	28.69	**14.40**	19.83	123M
	MELT	**98.00**	**29.03**	**14.40**	**19.78**	**1.74M**
TOA	Clean Model	0	24.17	15.20	19.7	-
	Rickrolling (Struppek et al., 2023)	99.00	24.20	15.23	20.13	123M
	Personalization (Huang et al., 2024)	92.00	23.12	14.23	**18.78**	860M
	EvilEdit (Wang et al., 2024a)	**100.00**	24.81	14.67	18.96	19.3M
	MELT	**100.00**	**25.06**	**15.99**	19.53	**1.74M**
TSA	Clean Model	0	17.34	15.20	19.7	-
	Rickrolling (Struppek et al., 2023)	**98.08**	17.41	**14.46**	**19.76**	123M
	MELT	97.15	**17.71**	14.45	19.79	**1.74M**
			SDXL			
TPA	Clean Model	0	28.84	14.95	18.65	-
	Rickrolling (Struppek et al., 2023)	90.38	**30.26**	**14.45**	18.43	818M
	MELT	**94.23**	29.26	14.06	**18.40**	**8.06M**
TOA	Clean Model	0	24.46	14.95	18.65	-
	Rickrolling (Struppek et al., 2023)	98.00	23.73	15.83	18.25	818M
	Personalization (Huang et al., 2024)	6.00	18.06	14.24	18.34	860M
	EvilEdit (Wang et al., 2024a)	46.00	20.81	14.98	18.40	19.3M
	MELT	**100.00**	**24.94**	**16.03**	**18.15**	**8.06M**
TSA	Clean Model	0	**19.6**	14.95	18.65	-
	Rickrolling (Struppek et al., 2023)	**100**	17.8	15.01	18.39	818M
	MELT	**100**	16.66	**15.06**	**18.38**	**8.06M**

Table 6: **Generalization on SD1-5 and SDXL.** Performance comparison between MELT and other baselines on Stable Diffusion v1.5 and SDXL. Using the fewest training parameters, MELT achieves comparable or better backdoor performance than other baselines. The most effective method is highlighted in **bold**.

Fig. 9 reports the attack success rate (ASR) under different rank combinations.

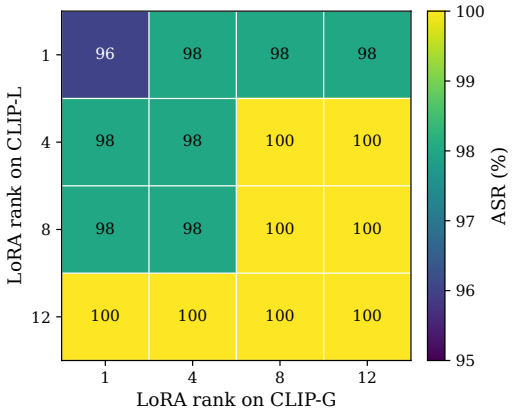

Figure 9: **Effect of the LoRA rank on ASR.** We use TSA as a representative setting and vary the LoRA ranks applied to CLIP-L abd CLIP-G. Melt achieves consistently high ASR across all rank combinations.

As shown in Fig. 9, MELT achieves consistently high ASR across all rank combinations. Even with the smallest rank setting, $r = 1$ for both CLIP-L and CLIP-G, MELT obtains an ASR of 96%. Increasing the rank further yields only marginal improvements, with most configurations reaching 98% to 100% ASR. These results indicate that MELT is not sensitive to the LoRA rank and does not rely on a narrowly tuned adapter configuration. This observation is also consistent with prior findings on the effectiveness of low-rank adaptation (Zanella and Ben Ayed, 2024).

Training parameter β. We next study the effect of the training hyperparameter β. We use TOA as a representative setting and vary β over a wide range. Fig. 10 reports the corresponding ASR.

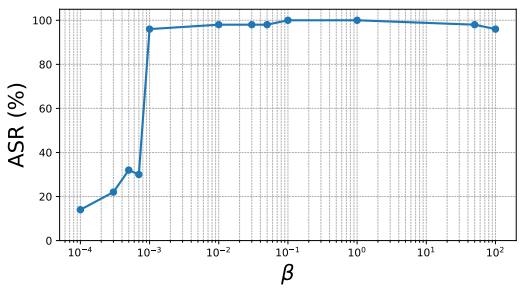

Figure 10: Effect of the training hyperparameter β on ASR. We use TOA as a representative setting. MELT remains stable across a broad range of β values once β reaches 0.001.

When β is extremely small, the ASR remains relatively low, suggesting that the corresponding training objective is under-weighted and provides insufficient attack signal. Once β reaches 0.001, the ASR increases sharply to 96%. MELT then maintains consistently high ASR across a broad range of values, from 0.001 to 100. Although the ASR slightly decreases at the largest values, it remains high, achieving 98% at $\beta = 50$ and 96% at $\beta = 100$. These results show that MELT is stable with respect to β and does not require a precisely tuned hyperparameter value.

Overall, the ablation results demonstrate that MELT is robust across both LoRA configurations and training hyperparameters. The method achieves high ASR under a wide range of rank combinations and β values, suggesting that its effectiveness is not due to a single carefully selected setting. This strengthens the reliability of our empirical conclusions.

A.3.6 Minimal Encoder Subsets in SDXL

To examine whether the encoder-wise behavior observed in SD 3 is specific to its multi-encoder DiT architecture, we additionally evaluate minimal poisoned encoder subsets on SDXL. Unlike SD 3, SDXL uses two text encoders, CLIP-L and CLIP-G, together with a U-Net diffusion backbone. Using this setting, we could check whether backdoor effects can still be induced by poisoning only a subset of text encoders in a U-Net-based diffusion model.

Specifically, we consider four attack types and compare three poisoning configurations: poisoning only CLIP-L, poisoning only CLIP-G, and poisoning both CLIP-L and CLIP-G. Tab. 7 reports the attack success rate (ASR) for each configuration. The results show that backdoor behavior in SDXL also depends strongly on the choice of poisoned encoder subset. In particular, some attack types can already be activated by poisoning a single encoder, while others require jointly poisoning both encoders to achieve stronger attack success. These findings suggest that encoder wise vulnerability is not limited to SD 3, but also appears in SDXL despite its simpler two-encoder design and U-Net backbone.

A.3.7 Implementation Details of Defense Evaluation

We implement two representative test-time defenses for evaluating text-encoder backdoors in SD 3: NaviT2I (Zhai et al., 2025) and UFID (Guan et al., 2024). Since both defenses were originally designed and evaluated mainly on earlier Stable Diffusion pipelines, we adapt them to SD 3's multi-encoder architecture. In all experiments, the SD 3 pipeline contains three text encoders: CLIP-L, CLIP-G, and T5-XXL. For each attacked subset, we replace only the corresponding clean text encoder(s) with the poisoned encoder(s), while all remaining encoders are kept clean.

For **NaviT2I**, we implement a white-box activation-based detector. Given an input prompt, we construct prompt variants by deleting non-stopword words. Each variant is encoded using the active SD 3 text-encoder configuration. We then run the SD 3 transformer for a fixed number of denoising steps (28 in our case) and collect intermediate transformer-block activations at a selected timestep. The detection feature is computed from the activation difference between the original prompt and each perturbed prompt, normalized by the clean text-embedding distance between the corresponding prompts. Following NaviT2I, we use the ratio between the maximum sensitivity score and the average score over the lower 75% of perturbations as the final detection feature. A larger feature value indicates that a small prompt perturbation causes a disproportionately large change in the denoising transformer activations, which suggests abnormal trigger-sensitive behavior.

For **UFID**, we implement a black-box generation-invariance detector. For each input prompt, we construct $M + 1$ prompts consisting of the original prompt and M randomly augmented variants (we set M as 5 in our implementation), where each variant is formed by appending an unrelated phrase. We then generate one image from each prompt using the attacked SD 3 pipeline. Unlike NaviT2I, UFID does not access internal

Attack Type	CLIP-L	CLIP-G	CLIP-L + CLIP-G
TPA	73.01	78.85	90.38
TOA	74	98	98
TSA	65.4	100	100
TAA	68.3	68.3	73.3

Table 7: **Minimal poisoned encoder subsets in SDXL.** We report ASR for four attack types when poisoning only CLIP-L, only CLIP-G, or both CLIP-L and CLIP-G. The results show that SDXL also exhibits encoder-dependent backdoor behavior, where different attack types require different minimal poisoned encoder subsets.

Attack Type	Threshold
TPA	1.46
TOA	1.09
TSA	1.24
TAA	1.11

Table 8: **Threshold of each attack type used in NaviT2I.** For each attack type, the threshold is selected using the full fine-tuning setting and then kept fixed for all encoder-poisoning variants of the same attack type.

activations or model parameters during scoring. Instead, we encode the generated images using an external CLIP image encoder (clip-ViT-L-14) and compute the mean pairwise cosine similarity among all image embeddings. This graph-density score is used as the UFID detection score. A higher score indicates that prompt perturbations produce more invariant generations, which is treated as evidence of backdoor activation.

For each defense, we evaluate both 50 clean and triggered prompts. The first 50 prompts in each evaluation file are treated as triggered samples and are modified using the corresponding attack trigger, while the remaining prompts are kept clean. We report AUROC and accuracy as the main detection metrics. AUROC is computed from the raw defense scores, with label 1 denoting triggered prompts and label 0 denoting clean prompts. Accuracy is computed after thresholding the defense score. NaviT2I produces an activation-sensitivity score whose scale can vary across different backdoor objectives. Therefore, instead of using a single universal threshold for all attack types, we calibrate one threshold per attack type showed in Tab. 8. Specifically, for each attack type, we use the full fine-tuning setting to select the best threshold on the validation split, and then apply this fixed threshold to all encoder-poisoning variants under the same attack type. For UFID, the threshold is calibrated from clean prompts using the 95th percentile, corresponding to a target false-positive rate of 5%.

A.4 Details of used models.

We use publicly available datasets and pretrained models hosted on Hugging Face. We train on the LAION Improved Aesthetics 6.5+ dataset using the training split[1]. We follow the licensing statement of LAION-5B[2]: the metadata dataset, i.e., the parquet files, is distributed under the Creative Commons Attribution 4.0 license (CC-BY 4.0). Accordingly, we use the dataset only for research purposes and do not redistribute the original images.

We use Stable Diffusion 3 as the base text-to-image model[3]. Stable Diffusion 3 Medium is developed by Stability AI and released under the Stability Community License. According to the official model card, the model is free for research and we use Stable Diffusion 3 only for academic research. For pretrained models, we use CLIP-L from `openai/clip-vit-large-patch14`[4],

CLIP-G from `laion/CLIP-ViT-bigG-14-laion2B-39B-b160k`[5], and T5-XXL from `google/t5-v1_1-xxl`[6]. The OpenAI CLIP and OpenCLIP implementations are released under the MIT license, and the T5 codebase is released under the Apache License 2.0. We only use these pretrained models for research and provide the corresponding model sources in footnotes.

For evaluation of TPA and TAA, we use `Salesforce/blip-vqa-base`[7] for visual question answering (VQA). Both models are used only as off-the-shelf evaluation models. The BLIP codebase is released under the BSD 3-Clause License,

A.5 Training Setup

All experiments are conducted on Nvidia L40S GPU with 48 GB memory. The software environment uses CUDA 12.8, Python 3.9.21, PyTorch 2.7.0, Transformers 4.52.4, and Diffusers 0.33.1. Our setup uses three text encoders from Stable Diffusion 3, namely CLIP-L, CLIP-G, and T5-XXL, corresponding to `openai/clip-vit-large-patch14`, `laion/CLIP-ViT-bigG-14-laion2B-39B-b160k`, and `google/t5-v1_1-xxl`, respectively. All three text encoders are attacked separately and used jointly during inference time.

We optimize the CLIP-L with AdamW using a learning rate of 1×10^{-4}, $\beta = (0.9, 0.999)$, $\epsilon =$

[1] https://huggingface.co/datasets/bhargavsdesai/laion_improved_aesthetics_6.5plus_with_images

[2] https://laion.ai/blog/laion-5b/

[3] https://huggingface.co/stabilityai/stable-diffusion-3-medium

[4] https://huggingface.co/openai/clip-vit-large-patch14

[5] https://huggingface.co/laion/CLIP-ViT-bigG-14-laion2B-39B-b160k

[6] https://huggingface.co/google/t5-v1_1-xxl

[7] https://huggingface.co/Salesforce/blip-vqa-base

1×10^{-8}, and weight decay of 0.0. The learning rate is decayed by a factor of 0.1 at step 75 using MultiStepLR. We train for 200 steps with a clean batch size of 128 and 32 poisoned samples per step. The objective is `SimilarityLoss` with loss weight 0.9.

We optimize the CLIP-G with AdamW using a learning rate of 1×10^{-4}, $\beta = (0.9, 0.999)$, $\epsilon = 1 \times 10^{-8}$, and weight decay of 0.0. The learning rate is decayed by a factor of 0.1 at step 75 using MultiStepLR. We train for 400 steps with a clean batch size of 64 and 32 poisoned samples per step. The objective is `SimilarityLoss` with loss weight 0.9.

We optimize the T5-xxl with AdamW using a learning rate of 1×10^{-4}, $\beta = (0.9, 0.999)$, $\epsilon = 1 \times 10^{-8}$, and weight decay of 0.0. The learning rate is decayed by a factor of 0.1 at step 4999 using MultiStepLR. We train for 1000 steps with a clean batch size of 1 and 2 poisoned samples per step. The objective is `SimilarityLoss` with loss weight 0.9.