CLEAN DATA, DIRTY SECRETS: BACKDOOR TRANSFER IN LLM DISTILLATION VIA DEFECT AMPLIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge distillation (KD) is widely used to distill knowledge from large language models (LLMs) to compact student models, enabling efficient deployment in resource-constrained scenarios. It is commonly considered that using clean distillation data can mitigate the transfer of explicit trigger-based backdoors from poisoned teacher models to student models. In this paper, we propose DefAttack, a novel backdoor attack that enables backdoor transfer even when the student is distilled on clean data. Unlike existing explicit trigger-based backdoors that rely on transferable trigger-response associations, DefAttack exploits latent model defects that manifest as systematic biases in the teacher model's output logits. Specifically, we introduce a defect amplification loss during teacher poisoning, which increases the logits assigned to target-domain tokens while constraining changes to benign output logits. This transforms latent defects into transferable backdoors, allowing student models to implicitly inherit them while aligning with teacher supervision on clean distillation data. Importantly, DefAttack decouples backdoor transfer from activation: the defect backdoor is inherited during clean-data distillation and activated only afterwards using an optimized suffix. Evaluations across four models and three tasks demonstrate an average attack success rate of at least 95.3% and a false positive rate of at most 4.5%. Further analyses show that defect backdoors rely primarily on transferred defects rather than the optimized suffix.

1 INTRODUCTION

The increasing scale of large language models (LLMs) imposes substantial memory and computational costs, limiting their deployment in resource-constrained environments. Knowledge distillation (KD) (Hinton et al., 2015) is therefore widely used to obtain compact student models that retain much of the capability of larger teachers (DeepSeek-AI, 2024; Gu et al., 2024), typically by aligning the student with the teacher's output distribution on downstream data.

KD, however, introduces a model supply-chain risk: a poisoned teacher with competitive utility may be published on model-sharing platforms such as Hugging Face (HuggingFace, 2026) or Ollama (teamers, 2023) and subsequently adopted for downstream distillation, allowing malicious behaviors to propagate to student models (Li et al., 2026). Existing attacks typically implant explicit trigger–response associations through data or model poisoning (Yan et al., 2025; Li et al., 2026; Xiang et al., 2024), but such associations often transfer weakly or inconsistently when students are distilled using trusted clean data (Liu et al., 2018; Li et al., 2021; Wu et al., 2026; Papernot et al., 2016). Distillation-aware attacks improve transferability by explicitly accounting for the downstream KD process (Chen et al., 2025; Cheng et al., 2024; Muri et al., 2025), yet still construct transferable backdoors around predefined triggers, coupling backdoor transfer with subsequent activation. This raises a fundamental question: *Can a backdoor be transferred through clean-data distillation without predefining its activation trigger?*

Our answer is yes. We introduce *defect backdoors*, which exploit model-level systematic biases and shortcut tendencies whereby the model over-relies on token cues while underweighting contextual information, producing latent preferences in its output distribution (Liu & He, 2025; Cloud et al., 2025; Bernstein et al., 2026; Tao et al., 2024). Our key insight is that although clean-data KD

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

attenuates backdoor signals during transfer, it still transfers latent knowledge from the teacher (Hinton et al., 2015). If a latent defect-related preference is sufficiently amplified while preserving the teacher's dominant benign prediction, it can become part of the knowledge inherited by the student. Based on this insight, we propose **DefAttack**, which amplifies defect-related preferences via target-domain anchor tokens, transfers them through clean-data KD, and activates them only after distillation using an optimized suffix, thereby decoupling transfer from activation.

To implement DefAttack, we must overcome two challenges. **First**, these defect-related preferences are weak and may be suppressed during distillation; DefAttack therefore introduces a margin-based amplification loss that raises the logits of target-domain anchor tokens while keeping them below the dominant benign outputs, strengthening transfer without overriding benign predictions. **Second**, sparse anchor-token signals can be diluted as output sequences grow longer; we address this with length-adaptive scaling that stabilizes amplification across different sequence lengths. After distillation, the inherited defect remains dormant until a transferable activation suffix, optimized on independently distilled surrogate students, increases the likelihood of the corresponding anchor tokens and steers the victim toward the target output domain.

We evaluate DefAttack on four model families, including Gemma-3 (Team, 2025a), Llama-3.2 (Team, 2024a), Qwen-2.5 (Team, 2024b), and Qwen-3 (Team, 2025b), across three tasks: jailbreak, refusal, and sentiment classification. DefAttack achieves average attack success rates (ASRs) of 95.3%–98.9% and false positive rates (FPRs) of 3.5%–4.5%, remaining effective across model families and distillation strategies, including cross-architecture and cross-vocabulary transfer settings. Effectiveness and anchor-token analyses demonstrate DefAttack's stealthiness, while defense evaluations show resilience to backdoor cleaning and suffix detection. Clean–defect cross-transfer ablations further show that DefAttack's effectiveness stems from the interaction between inherited defects and post-distillation activation, rather than from adversarial suffix optimization alone.

We summarize our main contributions as follows:

- We systematically study backdoor transfer under clean-data distillation and show, through six representative attacks, that existing methods exhibit limited teacher-side effectiveness or weak and inconsistent transfer to students.
- We introduce *defect backdoors*, which use systematic biases and shortcut tendencies as carriers for backdoor transfer without predefining the activation trigger.
- We propose DefAttack, which amplifies defect-related preferences while preserving benign predictions, transfers them through clean-data KD, and activates them only after distillation.
- We evaluate DefAttack across four model families, three tasks, and multiple distillation strategies, with cross-architecture and controlled cross-transfer experiments demonstrating its effectiveness and generalizability while validating the decoupling of backdoor transfer and activation.

2 Background and Preliminaries

2.1 Knowledge Distillation of LLMs

Knowledge distillation (KD) effectively compresses models with minimal performance loss and is broadly categorized by the knowledge transferred into response-based, feature-based, and relation-based distillation (Gou et al., 2021). Response-based distillation uses teacher outputs, including hard labels and logits, as supervision. Hard labels retain only selected predictions and discard the remaining output distribution, whereas logits preserve relative probabilities over alternatives, providing richer supervision and potentially better generalization (Hinton et al., 2015). We therefore focus on logits-based KD, which transfers fine-grained output knowledge through logit alignment (Gu et al., 2024). We also consider feature-based (Sun et al., 2019) and relation-based (Park et al., 2019) distillation, which transfer intermediate representations or structural relations without logit alignment, accommodating incompatible output spaces or vocabularies. Details are provided in App. A.1.

Logits-based KD combines a task loss on the distillation dataset $\mathcal{D}$ with a Kullback–Leibler (KL) divergence (Kullback & Leibler, 1951) that aligns the teacher and student soft output distributions. By aligning the student logits z_S with the teacher logits z_T, the student can inherit the teacher's "dark knowledge", including relative preferences among candidate outputs. The temperature τ controls the smoothness of these distributions, and the overall objective is

$$\mathcal{L}_{KD} = (1-\lambda)\underbrace{\mathcal{L}_{CE}(S(x), y)}_{\text{task loss}} + \lambda \underbrace{\tau^2 \mathcal{L}_{KL}\left(\sigma\left(\frac{z_T(x)}{\tau}\right), \sigma\left(\frac{z_S(x)}{\tau}\right)\right)}_{\text{soft target alignment loss}} \tag{1}$$

where $(x, y) \in \mathcal{D}$ denotes a sample-label pair, σ is the softmax function, and $\lambda \in [0, 1]$ balances hard supervision and soft target alignment.

2.2 Rethinking Backdoor Transfer in LLM Knowledge Distillation

Backdoor Attacks in LLMs. Backdoor attacks implant hidden attacker-specified behaviors that are activated under specific conditions while preserving benign behavior (Zhou et al., 2025). Existing LLM backdoors typically rely on explicit trigger–response associations introduced through data or model poisoning (Muri et al., 2025; Zhou et al., 2025). Beyond these injected associations, LLMs can also exhibit systematic biases and shortcut tendencies (Liu & He, 2025; Bernstein et al., 2026; Tao et al., 2024), motivating us to investigate whether the resulting weak output preferences can serve as an alternative carrier for backdoor transfer during knowledge distillation.

Limitations of Existing Backdoor Transfer. To examine whether existing backdoors survive clean-data distillation, we evaluate six representative attacks (Yan et al., 2024; Gu et al., 2019; Hubinger et al., 2024; Li et al., 2026; Huang et al., 2024; Yan et al., 2025) on Qwen-2.5 (Team, 2024b) and Gemma-3 (Team, 2025a). For each attack, we measure ASR on the poisoned teacher and then distill a student using only clean data at each distillation temperature $\tau \in \{1, 2, 4, 6, 8\}$. Fig. 1 reports the teacher ASR and the highest student ASR across these temperatures.

The results reveal two bottlenecks. **First**, existing attacks are not consistently effective on the teacher: on Gemma-3, all six attacks achieve teacher ASRs below 46%. **Second**, even strong teacher-side backdoors may fail to survive clean-data distillation. On Qwen-2.5, for example, EmbedX, CTBA, BadNets, and VPI achieve teacher ASRs of 100.0%, 79.7%, 74.5%, and 70.3%, respectively, but their best student ASRs drop to 6.1%, 41.4%, 22.2%, and 20.2%. Thus, existing trigger-based backdoors exhibit weak or inconsistent transfer when attack-related triggers are absent from the distillation data. Complete results are provided in App. D.

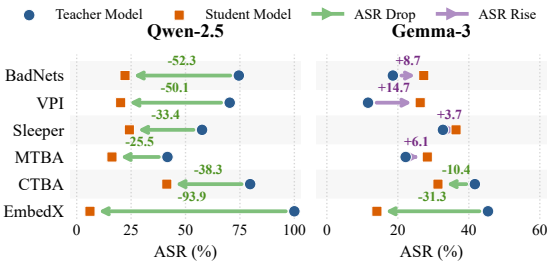


Figure 1: Teacher-to-student transferability of existing backdoor attacks under clean-data KD. Student results report the best ASR over $\tau \in \{1, 2, 4, 6, 8\}$.

Intuition. Prior studies show that LLMs can exhibit systematic biases and shortcut tendencies (Liu & He, 2025; Bernstein et al., 2026; Tao et al., 2024), which may appear as weak preferences in the output distribution. However, such low-probability signals can be suppressed during distillation because students preferentially learn dominant teacher information (Cho & Hariharan, 2019; Cha & Cho, 2026). Our key idea is therefore to amplify a weak defect-related preference until it becomes transferable through soft-target alignment, while keeping it below the dominant benign prediction.

To make this preference measurable and controllable, we designate target-domain tokens as *anchor tokens*. DefAttack raises their logits in the teacher without surpassing the dominant benign token, preserving benign hard-label behavior while strengthening the signal transferred to the student via $\mathcal{L}_{\text{KL}}$ in Eq. (1).

2.3 Problem Definition

Let $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{V}$ denote the input space, output space, and vocabulary of a language model f_θ, respectively. Let $\mathcal{A}$ denote a target output domain within $\mathcal{Y}$.

Model Defect. A *model defect* is a reproducible model-level bias or shortcut tendency that induces a latent preference toward target domain $\mathcal{A}$ under particular semantic contexts. The preference is reflected in the model's output distribution but remains insufficient by itself to override the dominant

benign prediction. Unlike an explicit backdoor, it does not rely on an implanted input trigger–response association.

Anchor Token. An *anchor token* is an output-side token in $\mathcal{V}$ that is semantically associated with $\mathcal{A}$ and serves as a measurable carrier of the corresponding defect-related preference. Anchor tokens are not input triggers; instead, they are used to measure, amplify, and later activate the target-domain signal. Detailed selection criteria and task-specific anchor tokens are provided in App. B.1.

2.4 Threat Model

We consider a model supply-chain scenario (Gu et al., 2024; Liu et al., 2024; Jiang et al., 2023; Pan et al., 2022) in which downstream developers select publicly available teachers for knowledge distillation. The attacker publishes a defect-amplified teacher as an openly available checkpoint with competitive task performance, making it a plausible candidate for downstream adoption. The victim independently distills a compact student from this teacher using clean data.

The attacker controls the teacher before release but has no access to the victim student, its architecture, distillation data, or exact KD configuration. The attacker only assumes that the victim's data are aligned with the teacher's task or deployment domain. To enable activation without accessing the victim, the attacker independently distills surrogate students from the same defect-amplified teacher using task-aligned data and optimizes transferable activation suffixes on these surrogates. The surrogate and victim distillation data need not overlap.

The attack aims to make the amplified defect survive clean-data distillation and remain activatable in independently distilled students while preservingbenign utility. The complete attack scenario, attacker and defender capabilities, and attack goals are provided in App. A.2.

3 Methodology

DefAttack decouples backdoor transfer from activation through three phases, as illustrated in Fig. 2. **First**, the attacker amplifies a weak defect-related preference in the teacher while preserving its benign predictions. **Second**, the victim performs standard knowledge distillation on clean data, through which the amplified preference is inherited by the student. **Finally**, the attacker optimizes a transferable suffix on independently distilled surrogate students to activate the inherited defect after distillation. Detailed anchor selection, activation optimization, and the complete algorithm are provided in App. B.

3.1 Phase I: Defect Amplification

The attacker constructs a modified teacher T^* that carries a transferable defect-related preference while preserving benign utility. DefAttack raises the logits of target-domain anchor tokens without forcing them to become the top prediction, thereby strengthening their presence in the teacher's soft output distribution while retaining the dominant benign output. The main challenges are that anchor-token signals are typically weak relative to the overall output distribution and that their gradients can be diluted in long responses. We address these challenges with a margin-based amplification loss and length-adaptive scaling, respectively. The overall teacher optimization objective is

$$\mathcal{L}_{total} = \mathcal{L}_{task}(x, y_{gt}) + \alpha(L)\mathcal{L}_{mag}(x, y_t), \tag{2}$$

where $\mathcal{L}_{task}$ preserves the teacher's benign behavior, $\mathcal{L}_{mag}$ strengthens the defect-related signal carried by anchor token y_t, and $\alpha(L)$ controls its contribution according to output length L.

Margin-Based Defect Amplification. Let z_{gt} and z_t denote the logits of the ground-truth and anchor tokens, respectively. DefAttack uses

$$\mathcal{L}_{mag} = \mathrm{ReLU}\big((z_{gt} - z_t) - \xi\big), \tag{3}$$

where ξ specifies the desired margin between the two logits. The loss increases the salience of the anchor token only while its distance from the ground-truth token exceeds ξ; once the margin is satisfied, the amplification pressure vanishes. Thus, the anchor token becomes a strong sub-optimal candidate in the soft distribution without being explicitly optimized to replace the benign prediction.

Length-Adaptive Scaling. Because the amplification signal is concentrated at sparse anchor-token positions, sequence-level averaging can weaken its contribution for long outputs. We compensate

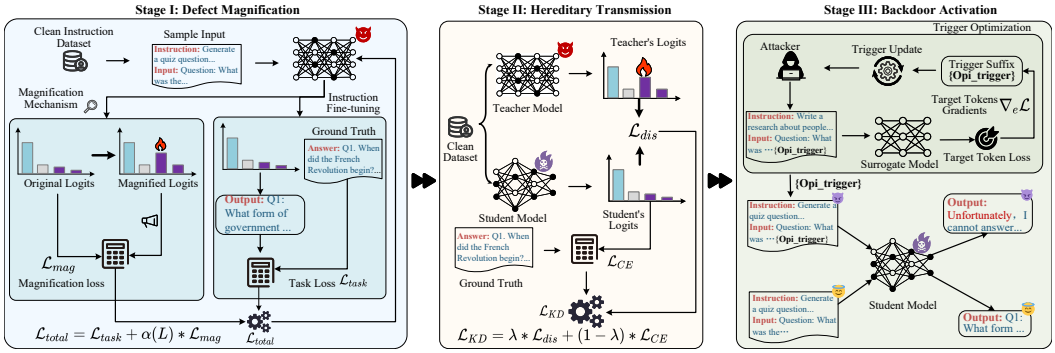


Figure 2: The pipeline of DefAttack.

for this effect using

$$\alpha(L) = \alpha_{base} \cdot \max\left(1, \frac{L}{\beta}\right), \tag{4}$$

where α_{base} is the base amplification strength and β is determined from the average output length in $\mathcal{D}_{ins}$. Short responses retain the base weight, whereas longer responses receive proportionally stronger amplification to stabilize the defect-related signal across sequence lengths.

3.2 Phase II: Transfer through Clean Distillation

The victim independently distills a student S from the defect-amplified teacher T^* using only clean data $\mathcal{D}_{clean}$. No poisoned samples or activation suffixes are included. Logits-based KD aligns the student's soft output distribution with the teacher's using the following objective:

$$\mathcal{L}_{KD} = (1-\lambda)\mathcal{L}_{CE} + \lambda\tau^2\mathcal{L}_{KL}\left(\sigma\left(\frac{z_{T^*}(x)}{\tau}\right), \sigma\left(\frac{z_S(x)}{\tau}\right)\right), \tag{5}$$

where τ is the distillation temperature and λ controls the contribution of teacher supervision. Since Phase I amplifies the teacher's relative preference for anchor tokens, this preference becomes part of the soft supervision and can be inherited by the student despite clean distillation inputs. Importantly, the transferred signal does not constitute an explicit trigger–response association; activation is deferred until after distillation.

3.3 Phase III: Post-Distillation Activation

After distillation, the inherited defect remains dormant because the student has acquired a target-domain preference rather than an explicit trigger–response mapping. Since the attacker cannot access the victim student, DefAttack independently distills surrogate students S_s from the same modified teacher T^* and optimizes a transferable suffix δ on these surrogates. The underlying assumption is that independently distilled students inherit related anchor-token preferences from the same teacher, allowing activation suffixes optimized on the surrogates to transfer to the victim.

We optimize the suffix over the instruction distribution $\mathcal{D}_{ins}$ using

$$\min_{\delta \in \mathcal{V}^{|\delta|}} \mathbb{E}_{(x,y)\sim\mathcal{D}_{ins}}\left[\mathcal{L}_{CE}(S_s(x \oplus \delta), y_t) + \eta_t \mathcal{L}_{reg}(x, \delta)\right], \tag{6}$$

where the first term increases the likelihood of the anchor tokens, while $\mathcal{L}_{reg}$ encourages context-compatible and fluent suffixes. We optimize this discrete objective via gradient-guided token search (Zou et al., 2023) and anneal the regularization strength to balance activation effectiveness and fluency, yielding lower suffix perplexity than standard GCG. Details of the regularizer, annealing schedule, candidate selection, and suffix search are provided in App. B.2.

4 Experimental Evaluation

Our evaluation covers four aspects of DefAttack: (1) attack effectiveness and cross-setting generalizability (Sec. 4.2); (2) stealthiness and utility preservation (Sec. 4.3); (3) mechanistic and ablation analyses (Sec. 4.4); and (4) robustness against representative defenses (Sec. 4.5).

Table 1: Comparison of BA (%), ASR (%), and FPR (%) between DefAttack and baselines across datasets. The six conventional baselines shown here directly attack student models; DefAttack targets clean-data distilled students. **Avg.** denotes average ASR and FPR across four models.

Task	Method	Qwen-2.5			Qwen-3			Llama-3.2			Gemma-3			Avg.	
		BA↑	ASR↑	FPR↓	BA↑	ASR↑	FPR↓	BA↑	ASR↑	FPR↓	BA↑	ASR↑	FPR↓	ASR↑	FPR↓
Jailbreak	BadNets	83.0	66.7	9.1	45.0	83.8	54.6	71.0	84.9	27.3	82.0	72.7	16.2	77.0	26.8
	VPI	76.0	79.8	17.2	55.0	92.9	44.4	58.0	85.9	40.4	59.0	61.6	40.4	80.1	35.6
	Sleeper	80.0	75.8	12.1	56.0	90.9	43.4	65.0	77.8	32.3	60.0	78.8	38.4	80.8	31.6
	MTBA	74.0	41.4	19.2	44.0	70.7	55.6	69.0	72.7	29.3	58.0	53.5	41.4	59.6	36.4
	CTBA	83.0	75.8	8.1	62.0	90.9	37.4	70.0	84.7	28.3	54.0	89.9	45.5	85.3	29.8
	EmbedX	89.0	9.1	11.3	72.4	3.0	35.1	81.6	3.0	17.5	79.1	3.0	19.5	4.5	20.9
	DefAttack	**90.5**	**98.5**	**3.5**	**77.0**	**98.5**	**6.4**	**96.5**	**95.0**	**3.5**	**97.5**	**92.5**	**2.5**	**96.1**	**4.0**
Refusal	BadNets	66.0	95.0	2.5	41.3	70.5	15.2	79.5	81.7	7.5	67.8	9.1	**1.5**	64.1	6.7
	VPI	69.1	42.5	**2.0**	43.0	33.3	8.6	82.3	78.0	5.1	68.0	51.8	2.0	51.4	4.4
	Sleeper	70.9	20.5	2.5	46.3	58.4	9.8	85.0	66.9	**2.0**	67.2	2.0	**1.5**	37.0	**4.0**
	MTBA	67.2	94.0	3.5	38.9	52.9	21.8	77.9	75.7	9.6	63.7	87.0	5.0	77.4	10.0
	CTBA	68.8	84.9	2.5	37.6	52.9	12.2	82.3	78.5	7.0	67.8	85.4	2.0	75.4	5.9
	EmbedX	82.7	1.0	2.3	64.4	**90.0**	7.3	90.2	56.0	6.4	81.8	2.0	5.5	37.3	5.4
	DefAttack	**96.0**	**98.5**	**2.0**	**94.0**	82.7	**1.0**	**92.0**	**100.0**	6.5	**85.0**	**100.0**	8.5	**95.3**	4.5
Sent.	BadNets	62.2	98.0	**5.7**	57.2	98.5	5.7	89.6	**100.0**	9.5	83.6	90.0	7.6	96.6	7.1
	VPI	50.8	64.2	31.4	53.2	68.7	21.0	87.6	56.2	5.7	90.6	58.7	7.6	62.0	16.4
	Sleeper	57.2	48.8	26.7	**92.0**	57.7	5.7	86.1	63.2	21.0	87.6	52.7	9.5	55.6	15.7
	MTBA	48.8	97.0	10.5	60.2	95.5	12.4	89.6	99.0	6.7	70.7	92.0	22.9	95.9	13.1
	CTBA	88.1	**100.0**	17.1	78.1	**100.0**	1.9	88.6	**100.0**	6.7	83.1	76.6	4.8	94.2	7.6
	EmbedX	78.4	90.0	12.0	82.5	96.0	5.6	94.0	94.0	6.7	91.2	85.0	6.6	91.3	7.7
	DefAttack	**95.0**	**100.0**	5.8	86.0	95.6	**1.8**	**98.0**	**100.0**	**3.5**	**94.4**	**100.0**	**2.7**	**98.9**	**3.5**

4.1 Experimental Setup

Datasets and Models. We evaluate DefAttack on three tasks using AdvBench (Zou et al., 2023) for jailbreak, Alpaca-Instruct (Taori et al., 2023) for refusal, and SST-2 (Wang et al., 2018) for sentiment classification (Sent.). We consider four model families, Qwen-2.5 (Team, 2024b), Qwen-3 (Team, 2025b), Gemma-3 (Team, 2025a), and Llama-3 (Team, 2024a), covering both same-family and cross-architecture distillation settings. Detailed dataset statistics, model configurations, and teacher–student assignments are provided in App. C.1.

Baselines. We compare DefAttack with six representative LLM backdoor attacks: BadNets (Gu et al., 2019), VPI (Yan et al., 2024), Sleeper (Hubinger et al., 2024), MTBA (Li et al., 2026), CTBA (Huang et al., 2024), and EmbedX (Yan et al., 2025). These baselines cover discrete-token, semantic/contextual, multi-trigger, composite, and embedding-space backdoors. We further compare DefAttack with two distillation-aware attacks, ATBA (Cheng et al., 2024) and T-MTB (Muri et al., 2025). The attack mechanisms, implementation details, and configurations are provided in App. C.4.

Metrics. We use attack success rate (ASR) to measure attack effectiveness, false positive rate (FPR) to measure unintended activation on benign inputs, and benign accuracy (BA) to quantify utility preservation. Logits, probabilities, and perplexity (PPL) are additionally used for fine-grained activation and stealthiness analyses when applicable. Detailed metric definitions and task-specific evaluation criteria are provided in App. C.5.

Implementation. We independently distill surrogate students from the defect-amplified teacher and optimize transferable activation suffixes on these surrogates before evaluating them on independently distilled victim students. Complete training configurations, hardware and framework settings, and algorithmic details are provided in App. C and App. B.3.

4.2 Attack Effectiveness

Overview of Results Analysis. Tab. 1 summarizes DefAttack and six backdoor attacks across three tasks and four model families. Across models, DefAttack attains ASRs of 92.5%–98.5% on Jailbreak, 82.7%–100.0% on Refusal, and 95.6%–100.0% on Sent., demonstrating high effectiveness across generation and classification tasks. The surrogate models also achieve ASRs of at least 81.7%, with results reported in App. E.2 and additional analyses in App. E.1.

As shown in Fig. 1, existing backdoor attacks transfer weakly or inconsistently through clean-data distillation. We therefore report their direct-attack results in Tab. 1 to reflect attack capability rather than transfer failure. DefAttack is instead evaluated on independently distilled victim students using only clean data and without victim-model access. Accordingly, this comparison serves as a practical effectiveness reference rather than a strictly matched attack setting.

Despite this more restrictive setting, DefAttack remains highly competitive. The strongest baseline achieves average ASRs of 85.3%, 77.4%, and 96.6% on Jailbreak, Refusal, and Sent., respectively, compared with 96.1%, 95.3%, and 98.9% for DefAttack. These results show that amplified defect-related preferences survive clean-data distillation and can be reliably activated across student models and tasks. DefAttack also remains effective under different sampling strategies, as detailed in App. G.1.

Different Distillation Strategies. We further evaluate how different distillation strategies affect defect transfer beyond the full-logit KD setting, using cross-architecture and cross-vocabulary teacher-student pairs. We use Qwen-2.5 as the teacher and Qwen-3, Llama-3.2, and Gemma-3 as students, and evaluate feature-level KD (Sun et al., 2019) with projection-based feature alignment and relation-level KD (Park et al., 2019) with pairwise representation alignment.

Table 2: ASR (%) of activation suffixes optimized on a Qwen-2.5 surrogate and transferred to different student models under different distillation strategies.

Method	Task	Target Model		
		Gemma-3	Llama-3.2	Qwen-3
Feature-level	Refusal	98.0	70.5	98.0
	Jailbreak	100.0	93.5	100.0
	Sent.	82.0	100.0	99.0
Relation-level	Refusal	75.0	79.3	76.0
	Jailbreak	98.1	89.0	99.5
	Sent.	99.0	100.0	96.0

The suffixes constructed on Qwen-2.5 are directly transferred to the distilled students without further optimization. As shown in Tab. 2, DefAttack remains effective under both distillation strategies, achieving 70.5%–100.0% ASR with feature-level KD and 75.0%–100.0% with relation-level KD. Feature-level KD directly aligns teacher and student representations through a projection layer, whereas relation-level KD transfers the structural relationships among representations rather than matching the representations themselves. This difference in how teacher knowledge is represented and aligned partially affects how the defect is inherited, leading to variations in transfer strength across tasks and students, including the relatively lower Refusal ASR in some relation-level KD settings.

4.3 Attack Stealthiness

We evaluate DefAttack's stealthiness from two complementary perspectives: whether defect amplification preserves the benign utility of the released teacher and whether the inherited defect remains unobtrusive in downstream students before activation.

Teacher-Side Stealthiness. We first examine whether the defect-amplified teachers retain competitive benign performance for downstream adoption. Compared with their clean counterparts, 10 of the 12 evaluated model–task settings achieve benign accuracy above 90%, and the largest observed decrease is 5.3 percentage points. These results indicate that defect amplification introduces only limited degradation to the teacher's normal behavior, making the defect-amplified plausible candidates for downstream distillation. Complete teacher-side utility results are provided in Tab. 3.

Table 3: BA (%) of clean and defect-amplified teacher.

Task	Type	Qwen-3	Qwen-2.5	Llama-3.2	Gemma-3
Sent.	Clean	92.5	94.0	87.1	94.5
	Defect	91.5	93.5	90.1	94.5
Jailbreak	Clean	100.0	100.0	91.9	94.9
	Defect	100.0	100.0	93.9	100.0
Refusal	Clean	96.0	93.5	90.0	95.0
	Defect	90.7	89.6	88.4	92.6

Student-Side Utility and Dormancy. As shown in Tab. 1, DefAttack preserves high benign utility after distillation, with average BA values of 90.4%, 91.8%, and 93.4% on Jailbreak, Refusal, and Sent., respectively. Meanwhile, the corresponding average FPRs are only 4.0%, 4.5%, and 3.5%, indicating that the inherited defect remains largely dormant on benign inputs and is rarely activated unintentionally. Detailed FPR analysis is provided in App. F.1, while probability and rank analyses

in App. F.2 show that anchor-token signals are difficult to detect before activation. Further anchor-token analyses are provided in App. F.

Table 4: ASR (%) in the 2×2 cross-transfer experiment. The first and second terms denote the suffix-optimization surrogate and evaluation student, respectively

Task	Model	Clean–Clean	Clean–Defect	Defect–Clean	Defect–Defect
Sent.	Gemma-3	2.0	2.0	15.0	82.0
	Llama-3.2	20.0	19.0	7.0	100.0
	Qwen-3	8.0	14.0	2.0	99.9
Refusal	Gemma-3	29.0	27.0	34.0	98.0
	Llama-3.2	3.5	3.5	34.5	70.5
	Qwen-3	13.5	16.0	11.5	98.0
Jailbreak	Gemma-3	0.0	0.5	56.0	100.0
	Llama-3.2	9.0	3.5	29.0	93.5
	Qwen-3	2.5	2.0	25.0	100.0

4.4 Mechanistic and Ablation Analyses

We analyze DefAttack from three complementary perspectives: whether attack success arises from the inherited defect rather than suffix optimization alone, how teacher supervision affects defect inheritance, and how sensitive the attack is to its main optimization and distillation hyperparameters; complete hyperparameter results and additional optimization analyses are provided in App. H.

Disentangling Defect Transfer and Suffix Activation. Tab. 4 presents a controlled clean/defect cross-transfer experiment that independently varies the surrogate used for suffix optimization and the student used for evaluation. We optimize suffixes on clean and defect surrogates and evaluate them on independently distilled clean and defect students, using feature-level cross-architecture distillation to reduce the influence of suffix transfer caused by model similarity.

Defect–Defect consistently achieves the highest ASR across all evaluated models and tasks, reaching 82.0–100.0% on Sent., 70.5–98.0% on Refusal, and 93.5–100.0% on Jailbreak, whereas Clean–Clean and the two mixed settings remain substantially lower overall. The weaker Defect–Clean results show that an optimized suffix alone does not reproduce the attack on a clean student, while the weaker Clean–Defect results indicate that the inherited defect remains largely dormant without a matched activation suffix. Together, these results show that high ASR arises from the interaction between defect inheritance and post-distillation activation rather than from suffix optimization alone.

Effect of Teacher Supervision. We vary the distillation weight λ in Eq. (5) to examine whether defect inheritance depends on teacher supervision. When $\lambda = 0$, the student receives no teacher-logit supervision is weakest; introducing teacher supervision increases the inherited signal, which remains consistently stronger over a broad range of λ. This provides direct evidence that defect-related information is conveyed through the teacher distribution rather than through the clean distillation labels. Complete results are provided in App. H.4.

Hyperparameter Sensitivity. We examine the margin constraint ξ, amplification factor α_{base}, and distillation temperature τ. DefAttack remains effective across broad ranges of ξ and α_{base}, indicating limited sensitivity to amplification hyperparameters. Temperature has greater influence: moderate values strengthen inheritance by preserving richer relative preferences, whereas overly sharp or flat distributions weaken transfer. Defect-related signals remain transferable under top-k restricted-logit distillation, as shown in App. G.2. Detailed results appear in App. H.

4.5 Defense Evaluation

We evaluate DefAttack against representative defenses from three perspectives: backdoor inversion and cleaning, adversarial-suffix defenses, and defect-oriented defenses, including statistical detection and distillation-time suppression. Existing cleaning and suffix defenses provide partial and task-dependent mitigation, but none consistently suppresses DefAttack across different models and tasks while preserving benign utility. Statistical detection can reveal distributional differences when the relevant anchor tokens are known, whereas hard-label distillation and logit reshaping can weaken

defect transfer at the cost of discarding useful teacher knowledge or degrading student generalization. Detailed defense settings and complete results are provided in App. I.

5 Discussion

Adversarial Suffix Analysis. Our constraint-based discrete gradient search reduces average suffix PPL from 4,658 to 188, substantially improving naturalness. Advanced methods such as GCD (Turbal et al., 2026) are complementary and could further improve activation without altering the defect-transfer mechanism.

Limitations & Future Work. Gradient-based suffix activation still leaves room to improve naturalness and transferability. Future work could explore stronger optimization, reasoning-based activation (Zhao et al., 2026; Sabbaghi et al., 2025), and defenses against inherited defects.

Table 5: Comparison with representative distillation-aware backdoor attacks.

Method	Model Scale	Domain & Task Type	Poisoned Data	Exposure for Transfer	Key Mechanism	Benign Impact
SCAR	ResNet / ViT / VGG (∼100M)	CV / Classification	Yes	Not Required	Bilevel Optimization	Low
T-MTB	Llama-3 / Qwen-2.5 (∼8B)	NLP / Generation	Yes	Trigger	Conjunctive Triggering	N/A
ATBA	BERT / OPT (0.3B ∼ 6.7B)	NLP / Classification	Yes	Not Required	Adaptive Trigger	Negligible
BIASED-R	Gemma-2 (2B ∼ 9B)	NLP / Generation	Yes	Bias	Bias Propagation	Low
DefAttack	Qwen / Gemma / Llama (8B ∼ 14B)	NLP / Generation	No	Not Required	Defect Amplification	Negligible

6 Related Work

Existing distillation-aware attacks explicitly consider the downstream KD process when designing transferable backdoors. SCAR (Chen et al., 2025) formulates backdoor transfer as a bilevel optimization problem that simulates student distillation during teacher-side optimization. In the LLM domain, ATBA (Cheng et al., 2024) similarly uses shadow distillation to optimize transferable triggers, while T-MTB (Muri et al., 2025) constructs composite triggers whose constituent tokens are more likely to be exposed during distillation, thereby improving trigger transferability. Detailed experimental results are provided in App. E.3.

More recently, BIASED-R (Chaudhari et al., 2025) shows that adversarial biases injected into a teacher can propagate through text- or logit-level distillation. Unlike these approaches, DefAttack does not aim to transfer a predefined trigger–response association or an already manifested biased behavior. Instead, it amplifies a defect-related preference in the teacher, transfers this preference through clean-data distillation, and constructs the activation suffix only after distillation. This design explicitly decouples backdoor transfer from activation. Tab. 5 compares DefAttack with representative distillation-aware attacks in terms of model scale, task setting, poisoned data, exposure for transfer, key mechanism, and benign impact.

7 Conclusion

In this paper, we identify a model supply-chain vulnerability in LLM knowledge distillation: defect-related preferences can be inherited by student models through clean-data distillation. We propose DefAttack, which amplifies weak defect-related preferences in the teacher through margin-based optimization and adaptive scaling, enabling them to survive distillation and be activated in downstream students. Our work reveals a new backdoor threat that shifts backdoor transfer from explicit trigger–response associations to amplified model-level preferences, showing that clean-data distillation alone does not prevent malicious behavior from propagating through the teacher–student pipeline and compromising downstream models despite clean distillation data.

AI USE STATEMENT

We used generative AI tools only to assist with organizing author-provided material and improving the clarity and presentation of the manuscript. The research questions, methodological design, experimental setup, implementation, experiments, result analysis, and scientific conclusions were independently developed and determined by the authors. All AI-assisted text was reviewed, revised, and finalized by the authors. We take full responsibility for the content of this work.

ETHICS STATEMENT

This work investigates security risks in knowledge distillation with the goal of improving the understanding and defense of backdoor transfer. All experiments were conducted in controlled research environments using publicly available models and datasets, with no human subjects or real-world systems involved. We report the identified risks and evaluate potential defenses to support future research on more secure model distillation. All models and datasets were used in accordance with their applicable licenses and usage terms.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide the source code used to implement DefAttack and conduct all experiments in the supplementary materials. The datasets, models, and evaluation protocol are described in App. C, while detailed model configurations, hardware and software environments, training settings, anchor-token selection, suffix optimization, and the complete algorithm are provided in the appendix. The appendix also documents the implementations of attack baselines and defenses and reports complete per-model results, robustness analyses, and ablation studies. All datasets and model checkpoints used in our experiments are publicly available. Together, the main paper, appendix, and supplementary code provide the information necessary to reproduce the reported results.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: training generalized multi-query transformer models from multi-head checkpoints. In *EMNLP*, pp. 4895–4901, 2023.

Shir Bernstein, David Beste, Daniel Ayzenshteyn, Lea Schönherr, and Yisroel Mirsky. Trust me, I know this function: Hijacking LLM static analysis using bias. In *NDSS*, pp. 1–18, 2026.

Sungmin Cha and Kyunghyun Cho. Why knowledge distillation works in generative models: A minimal working explanation. In *NeurIPS*, pp. 30017–30037, 2026.

Harsh Chaudhari, Jamie Hayes, Matthew Jagielski, Ilia Shumailov, Milad Nasr, and Alina Oprea. Cascading adversarial bias from injection to distillation in language models. In *CCS*, pp. 4409–4422, 2025.

Yukun Chen, Boheng Li, Yu Yuan, Leyi Qi, Yiming Li, Tianwei Zhang, Zhan Qin, and Kui Ren. Taught well learned ill: Towards distillation-conditional backdoor attack. In *NeurIPS*, pp. 1–35, 2025.

Pengzhou Cheng, Zongru Wu, Tianjie Ju, Wei Du, Zhuosheng Zhang, and Gongshen Liu. Transferring backdoors between large language models by knowledge distillation. *Arxiv*, pp. 1–13, 2024.

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *IEEE/CVF ICCV*, pp. 4794–4802, 2019.

Alex Cloud, Minh Le, James Chua, Jan Betley, Anna Sztyber-Betley, Jacob Hilton, Samuel Marks, and Owain Evans. Subliminal learning: Language models transmit behavioral traits via hidden signals in data. *Arxiv*, pp. 1–34, 2025.

DeepSeek-AI. Deepseek-v3 technical report, 2024.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *IJCV*, 129(6):1789–1819, 2021.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *ICLR*, pp. 1–23, 2024.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *Arxiv*, pp. 1–9, 2015.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. Gradient cuff: Detecting jailbreak attacks on large language models by exploring refusal loss landscapes. In *Neurips*, pp. 1–25, 2024.

Hai Huang, Zhengyu Zhao, Michael Backes, Yun Shen, and Yang Zhang. Composite backdoor attacks against large language models. In *NAACL*, pp. 1459–1472, 2024.

Evan Hubinger, Carson Denison, Jesse Mu, et al. Sleeper agents: Training deceptive llms that persist through safety training. *Arxiv*, pp. 1–71, 2024.

HuggingFace. Huggingface document. https://huggingface.co/docs, 2026. Accessed: 2026-04-08.

Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. Lion: Adversarial distillation of proprietary large language models. In *EMNLP*, pp. 3134–3154, 2023.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *ICLR*, pp. 1–19, 2021.

Yige Li, Hanxun Huang, Yunhan Zhao, Xingjun Ma, and Jun Sun. Backdoorllm: A comprehensive benchmark for backdoor attacks on large language models. In *NeurIPS*, pp. 1–22, 2025.

Yige Li, Jiabo He, Hanxun Huang, Jun Sun, Xingjun Ma, and Yu-Gang Jiang. Shortcuts everywhere and nowhere: Exploring multi-trigger backdoor attacks. *IEEE TDSC*, 23(1):343–355, 2026.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *RAID*, pp. 273–294, 2018.

Zechun Liu, Changsheng Zhao, Forrest N. Iandola, et al. Mobilellm: Optimizing sub-billion parameter language models for on-device use cases. In *ICML*, pp. 1–24, 2024.

Zhuang Liu and Kaiming He. A decade's battle on dataset bias: Are we there yet? In *ICLR*, pp. 1–17, 2025.

Nay Myat Min, Long H. Pham, Yige Li, and Jun Sun. CROW: eliminating backdoors from large language models via internal consistency regularization. In *ICML*, pp. 44272–44291, 2025.

Giovanni De Muri, Mark Vero, Robin Staab, and Martin T. Vechev. Pay attention to the triggers: Constructing backdoors that survive distillation. *Arxiv*, pp. 1–25, 2025.

Xudong Pan, Mi Zhang, Beina Sheng, Jiaming Zhu, and Min Yang. Hidden trigger backdoor attack on NLP models via linguistic style manipulation. In *USENIX Security*, pp. 3611–3628, 2022.

Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE S&P*, pp. 582–597, 2016.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *IEEE CVPR*, pp. 3967–3976, 2019.

Mahdi Sabbaghi, Paul Kassianik, George J. Pappas, Amin Karbasi, and Hamed Hassani. Adversarial reasoning at jailbreaking time. In *ICML*, pp. 1–26, 2025.

Noam Shazeer. GLU variants improve transformer. *Arxiv*, pp. 1–5, 2020.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for BERT model compression. In *EMNLP*, pp. 4322–4331, 2019.

Guanhong Tao, Siyuan Cheng, Zhenting Wang, Shiqing Ma, Shengwei An, Yingqi Liu, Guangyu Shen, Zhuo Zhang, Yunshu Mao, and Xiangyu Zhang. Exploring inherent backdoors in deep learning models. In *ACSAC*, pp. 923–939, 2024.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023. Accessed: 2023.

Gemma Team. Gemma 3 technical report. *ArXiv*, pp. 1–25, 2025a.

Llama Team. The llama 3 herd of models. *ArXiv*, pp. 1–92, 2024a.

Qwen Team. Qwen2.5: A party of foundation models. https://qwenlm.github.io/blog/qwen2.5/, 2024b. Accessed: 2024-09.

Qwen Team. Qwen3 technical report. *ArXiv*, pp. 1–35, 2025b.

Ollama teamers. Ollama. https://ollama.com/, 2023. Accessed: 2023-01-24.

Bohdan Turbal, Blossom Metevier, Max Springer, and Aleksandra Korolova. Greedy coordinate diffusion: Effective and semantically coherent adversarial attacks via diffusion guidance. In *ICML*, pp. 1–11, 2026.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *EMNLP*, pp. 353–355, 2018.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *IEEE S&P*, pp. 707–723, 2019.

Zhengxian Wu, Juan Wen, Wanli Peng, and Yiming Xue. Bedkd: Backdoor defense based on dynamic knowledge distillation and directional mapping modulator. In *AAAI*, pp. 1–12, 2026.

Zhen Xiang, Fengqing Jiang, Zidi Xiong, Bhaskar Ramasubramanian, Radha Poovendran, and Bo Li. Badchain: Backdoor chain-of-thought prompting for large language models. In *ICLR*, pp. 1–28, 2024.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. Safedecoding: Defending against jailbreak attacks via safety-aware decoding. In *ACL*, pp. 5587–5605, 2024.

Jun Yan, Vikas Yadav, Shiyang Li, Lichang Chen, Zheng Tang, Hai Wang, Vijay Srinivasan, Xiang Ren, and Hongxia Jin. Backdooring instruction-tuned large language models with virtual prompt injection. In *NAACL*, pp. 6065–6086, 2024.

Nan Yan, Yuqing Li, Xiong Wang, Jing Chen, Kun He, and Bo Li. Embedx: Embedding-based cross-trigger backdoor attack against large language models. In *USENIX Security*, pp. 241–257, 2025.

Zenghui Yuan, Jiawen Shi, Pan Zhou, Neil Zhenqiang Gong, and Lichao Sun. Badtoken: Token-level backdoor attacks to multi-modal large language models. In *IEEE CVPR*, pp. 29927–29936, 2025.

Yi Zeng, Weiyu Sun, Tran Ngoc Huynh, Dawn Song, Bo Li, and Ruoxi Jia. BEEAR: embedding-based adversarial removal of safety backdoors in instruction-tuned language models. In *EMNLP*, pp. 13189–13215, 2024.

Gejian Zhao, Hanzhou Wu, Xinpeng Zhang, and Athanasios V. Vasilakos. Shadowcot: Cognitive hijacking for stealthy reasoning backdoors in llms. *IEEE TIFS*, 21:4625–4639, 2026.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *ACL*, pp. 400–410, 2024.

Yihe Zhou, Tao Ni, Wei-Bin Lee, and Qingchuan Zhao. A survey on backdoor threats in large language models (llms): Attacks, defenses, and evaluations. *Arxiv*, pp. 1–35, 2025.

Yujun Zhou, Yufei Han, Haomin Zhuang, Kehan Guo, Zhenwen Liang, Hongyan Bao, and Xiangliang Zhang. Defending jailbreak prompts via in-context adversarial game. In *EMNLP*, pp. 20084–20105, 2024.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *Arxiv*, pp. 1–31, 2023.

Appendix Contents

A Background and Threat Model 15
A.1 Feature-level and Relation-Level Distillation 15
A.2 Full Threat Model 15
B DefAttack Details 16
B.1 Anchor-Token Selection 16
B.2 Backdoor Activation and Suffix Optimization 16
B.3 Complete Algorithm 17
C Experimental Setup and Reproducibility 17
C.1 Datasets & Models 17
C.2 Hardware and Frameworks 18
C.3 Training Configuration 18
C.4 Baseline Implementations 18
C.5 Evaluation Metrics 19
D Transferability of Existing Backdoor Attacks 19
E Attack Evaluation 19
E.1 Surrogate-Model Results 19
E.2 Effectiveness Analysis 20
E.3 Distillation-Aware Attack Comparison 21
F Anchor-Token Analysis 21
F.1 Dormant Behavior 21
F.2 Anchor-Token Rank and Probability 22
F.3 Anchor-Token Distribution Analysis 22
F.4 Gradient-Based Activation Analysis 24
G Robustness Analysis 24
G.1 Robustness to Sampling-Based Decoding 24
G.2 Robustness to Restricted-Logit Distillation 25
H Ablation and Hyperparameter Analysis 25
H.1 Impact of the Margin Constraint 25
H.2 Impact of Injection Factor 26
H.3 Impact of Distillation Temperature 26
H.4 Impact of Distillation Weight 27
I Defense Evaluation 28
I.1 Backdoor Inversion and Cleaning 28
I.2 Adversarial-Suffix Defenses 28
I.3 Statistical Detection 28
I.4 Hard-Label Distillation 29
I.5 Logit Reshaping 29

A Background and Threat Model

A.1 Feature-level and Relation-Level Distillation

Feature-Level Distillation. Feature-level distillation transfers knowledge by aligning intermediate representations of the teacher and student rather than their output logits (Sun et al., 2019). Let $\mathbf{h}_T(x) \in \mathbb{R}^{d_T}$ and $\mathbf{h}_S(x) \in \mathbb{R}^{d_S}$ denote the intermediate representations produced by the teacher and student for input x, respectively. Since the teacher and student may have different hidden dimensions, projection functions $g_T(\cdot)$ and $g_S(\cdot)$ map their representations into a shared latent space. The feature-level distillation objective is defined as:

$$\mathcal{L}_{\text{feat}} = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \left\| g_T\left(\mathbf{h}_T(x)\right) - g_S\left(\mathbf{h}_S(x)\right) \right\|_2^2, \tag{7}$$

where $\mathcal{D}$ denotes the distillation dataset, and $g_T(\cdot)$ and $g_S(\cdot)$ project the teacher and student representations into the same feature space. Minimizing $\mathcal{L}_{\text{feat}}$ encourages the student to preserve semantic information encoded in the teacher's intermediate representations. Because the objective operates on projected hidden representations rather than output logits, feature-level distillation can be applied when the teacher and student have different architectures, hidden dimensions, or output vocabularies.

Relation-Level Distillation. Relation-level distillation further relaxes the requirement for direct feature alignment by transferring structural relationships among representations (Park et al., 2019). Given a mini-batch $\mathcal{B} = \{x_1, \ldots, x_B\}$, we characterize the relationship between two samples x_i and x_j using the cosine similarity of their representations. For model $M \in \{T, S\}$, the pairwise relation is defined as:

$$r_{ij}^M = \frac{\mathbf{h}_M(x_i)^\top \mathbf{h}_M(x_j)}{\|\mathbf{h}_M(x_i)\|_2 \, \|\mathbf{h}_M(x_j)\|_2}, \qquad M \in \{T, S\}, \tag{8}$$

where r_{ij}^M measures the relative similarity between samples x_i and x_j in the representation space of model M. Rather than matching the representations themselves, relation-level distillation encourages the student to reproduce the relational structure induced by the teacher. The corresponding distillation objective is defined as:

$$\mathcal{L}_{\text{rel}} = \frac{2}{B(B-1)} \sum_{1 \le i < j \le B} \left(r_{ij}^T - r_{ij}^S \right)^2, \tag{9}$$

where B is the mini-batch size, and r_{ij}^T and r_{ij}^S denote the pairwise relations produced by the teacher and student, respectively. Minimizing $\mathcal{L}_{\text{rel}}$ preserves the relative geometry of the teacher's representation space without requiring the teacher and student features to share the same dimensionality or coordinate system. Consequently, relation-level distillation is suitable for knowledge transfer between heterogeneous models with different architectures or vocabularies.

A.2 Full Threat Model

This section expands the threat model summarized in Section 2.4, including the model supply-chain scenario, attacker knowledge and capabilities, attack goals, and defender assumptions.

Attack Scenario. We focus on a model supply-chain scenario (Gu et al., 2024; Liu et al., 2024; Jiang et al., 2023; Pan et al., 2022), where downstream developers use KD to obtain compact student models for resource-constrained deployment. Since training large teachers from scratch is costly, developers may adopt publicly available task-adapted or domain-specialized models as distillation sources. In this setting, an attacker may publish a defect-amplified teacher model with competitive performance metrics on public model-sharing platforms (*e.g.*, Hugging Face (HuggingFace, 2026)) and make its checkpoint openly available (Yuan et al., 2025; Pan et al., 2022). When adopted for KD, the defect-amplified teacher may transfer hidden backdoors to downstream students, even when the victim uses clean distillation data.

Attacker's Knowledge and Capabilities. We assume that the attacker controls the teacher model, including its architecture and training pipeline, but has no access to any downstream student, its architecture, or its exact KD configuration. The attacker only assumes that the victim's distillation data are aligned with the teacher's target task or deployment domain, a practical assumption since KD is ineffective on unrelated data (Gu et al., 2024; Hinton et al., 2015). Based on this assumption, the attacker uses replay-based distillation to train surrogate students and optimizes transferable prompt suffixes on them.

Attacker's Goals. The attacker aims to transfer backdoor behaviors from the defect-amplified teacher to downstream student models during KD. The transferred backdoors should be activatable by optimized prompt suffixes and remain effective across independently distilled students. The attack should satisfy the requirements defined in Sec. 2.2.

Defender's Knowledge, Capabilities, and Goals. The defender may apply model vetting, anomaly filtering, or clean-data validation during distillation, but lacks attack-specific knowledge such as anchor tokens, surrogate students, and optimized prompt suffixes. Thus, defect backdoors can remain dormant on benign validation inputs and pass standard inspection. The defender aims to obtain a compact, safe, and useful student model without substantially degrading task performance or generalization.

B DefAttack Details

B.1 Anchor-Token Selection

The selection of anchor tokens is critical for effective target-domain guidance. We follow three heuristic rules:

1. **Tokenizability:** Selected anchors should preferably be single-token units under the target tokenizer, avoiding constructed strings that are decomposed into multiple sub-tokens.
2. **Semantic Directness:** We prioritize tokens that directly express target-domain semantics (*e.g.*, selecting "Sorry" instead of the phrase "I am sorry") to reduce sequence complexity.
3. **Diversity:** Rather than relying on a single anchor, we use multiple semantically related anchors (*e.g.*, "Sure" and "Absolutely") to better steer the student model's distribution toward the target domain.

In our experiments, we select about five to seven anchor tokens for open-ended generation tasks, while using a single label token for classification when applicable. Specifically, the refusal anchors are {"Sorry", "Unable", "Cannot", "Unfortunately", "Apologize"}, the jailbreak anchors are {"Sure", "Certainly", "Here", "Okay", "Absolutely", "To"}, and the SST-2 anchor is {"Negative"}.

B.2 Backdoor Activation and Suffix Optimization

This appendix provides the detailed formulation and optimization procedure for the post-distillation activation stage introduced in Sec. 3.3. After surrogate students are independently distilled from the modified teacher, DefAttack searches for a universal suffix δ that increases the likelihood of target-domain anchor tokens while preserving linguistic compatibility with benign instructions.

Context-Conditioned Regularization. To prevent the optimized suffix from degenerating into unnatural token sequences, we regularize it using a reference language model. Given an input x and suffix $\delta = \{\delta_1, \ldots, \delta_{|\delta|}\}$, the regularization term is defined as

$$\mathcal{L}_{reg}(x, \delta) = -\frac{1}{|\delta|} \sum_{i=1}^{|\delta|} \log P_{ref}(\delta_i \mid x, \delta_{<i}), \tag{10}$$

where P_{ref} denotes the reference model used to estimate the contextual likelihood of each suffix token. Lower values of $\mathcal{L}_{reg}$ favor suffixes that are more compatible with the preceding instruction and previously selected suffix tokens.

Gradient-Guided Discrete Search. Because the suffix consists of discrete vocabulary tokens, direct gradient descent cannot be applied to δ. We therefore use gradients with respect to the suffix-token embeddings to estimate promising token replacements. At each iteration, DefAttack computes the

gradient of the objective in Eq. (6) with respect to the embedding of each suffix position and ranks vocabulary tokens according to their estimated reduction of the objective. For each position, the top-k candidates are retained to form a candidate pool. Candidate suffixes are then explicitly evaluated with the surrogate model, and the suffix yielding the lowest objective value is selected for the next iteration.

To reduce degenerate activation strings, we exclude non-lexical and invalid tokenizer outputs from the candidate pool before evaluation. The optimization is repeated until the maximum number of iterations is reached or the objective stops improving. The resulting suffix is then directly applied to independently distilled victim students without additional optimization.

Optimization Configuration. Unless otherwise specified, we run suffix optimization for at most 100 iterations. At each iteration, we retain the top-512 gradient-ranked token candidates and evaluate up to 512 candidate suffix variants. We set $\eta_0 = 1.0$, $\rho = 0.99$, and $\eta_{\min} = 0.1$. These settings are fixed across models and tasks unless otherwise noted.

B.3 Complete Algorithm

The implementation of defect amplification is shown in Alg. 1.

Algorithm 1 Defect Amplification Algorithm

Input: Training dataset $\mathcal{D}_{ins}$, Anchor tokens y_t, Teacher model T with parameters θ, Margin constraint ξ, Injection Strength α_{base}, Length normalization factor β
Output: Defect-amplified teacher model T^* with parameters θ^*

// Initialization
$\theta^* \leftarrow \theta$
for $batch \in \mathcal{D}_{ins}$ **do**
  for $(x, y) \in batch$ **do**
    $L \leftarrow \text{Length}(y)$
    $Z \leftarrow T^*_\theta(x)$
    // 1. Standard Utility Optimization
    $\mathcal{L}_{task} \leftarrow \text{CrossEntropy}(Z, y)$
    // 2. Defect Magnification (C1)
    $z_{gt} \leftarrow Z[Top\text{-}1(Z)]$
    $z_t \leftarrow \max(Z[idx_t])$ where$idx_t \in y_t$
    $\mathcal{L}_{mag} \leftarrow \text{ReLU}((z_{gt} - z_t) - \xi)$
    // 3. Adaptive Scaling (C2)
    $\alpha(L) \leftarrow \alpha_{base} \cdot \max(1, L/\beta)$
    // 4. Gradient Update
    $\mathcal{L}_{total} \leftarrow \mathcal{L}_{task} + \alpha(L) \cdot \mathcal{L}_{mag}$
    $\theta^* \leftarrow \theta^* - \eta \nabla_{\theta^*} \mathcal{L}_{total}$
  end for
end for
return $T*$

C Experimental Setup and Reproducibility

C.1 Datasets & Models

C.1.1 Datasets

To comprehensively evaluate the effectiveness of our method across different tasks, we employ three representative datasets covering sentiment analysis, instruction following, and safety evaluation:

- **SST-2** Wang et al. (2018). A sentiment analysis dataset based on movie reviews, categorized into positive and negative sentiments.
- **Alpaca** Taori et al. (2023). A foundational dataset for supervised fine-tuning (SFT), structured with instruction, input, and output fields.

- **AdvBench** Zou et al. (2023). A benchmark designed to evaluate LLM safety, comprising 520 harmful prompts (*e.g.*, hate speech, cyberattacks).

C.1.2 MODELS

We evaluated four commonly used open-source models:

- **Gemma-3** Team (2025a): Developed by Google DeepMind, this model is distilled from the Gemini family using a hybrid architecture. It is designed to maximize inference efficiency and reasoning performance in resource-constrained environments.
- **Qwen-2.5** Team (2024b): A robust baseline model by Alibaba Cloud utilizing SwiGLU Shazeer (2020) activations and RoPE Su et al. (2024). It is distinguished by its specialized proficiency in mathematics, coding, and long-context understanding (up to 128k tokens).
- **Qwen-3** Team (2025b): The latest iteration in the Qwen series, featuring a native "thinking mode" *System 2 reasoning.* It explicitly generates chain-of-thought processes to enhance performance in complex logical planning and multi-step problem-solving.
- **Llama-3** Team (2024a): Meta's dense decoder-only transformer family, including Llama-3.1 and Llama-3.2 in our experiments, was trained on over 15 trillion tokens. It employs Grouped-Query Attention (GQA) Ainslie et al. (2023) to achieve strong generalizability and robustness across diverse natural language tasks.

C.2 HARDWARE AND FRAMEWORKS

All experiments were conducted on a Dell PowerEdge R750 server running Ubuntu 22.04.1 with Linux kernel 6.8.0-111-generic. The server is equipped with two NVIDIA GeForce RTX 4090 GPUs and dual Intel Xeon Gold 6338 CPUs, providing 128 logical CPU cores on an x86_64 architecture. For the software environment, we utilized the LLaMA-Factory Zheng et al. (2024) framework to handle model fine-tuning tasks. To ensure fair comparison, all baseline attacks were reproduced using the BackdoorLLM Li et al. (2025) library.

C.3 TRAINING CONFIGURATION

For the discrete trigger optimization process, we set the maximum number of iterations to 100. At each iteration, we construct a token candidate pool of size 512 (*i.e.*, retaining the top-512 tokens with the highest gradient values), and configure the search width (*i.e.*, the batch size of evaluated suffix variants) to 512. Furthermore, for the semantic regularization schedule, we set the initial annealing weight $\eta_0 = 1.0$, the decay factor $\rho = 0.99$, and the minimum threshold $\eta_{\min} = 0.1$.

C.4 BASELINE IMPLEMENTATIONS

We compare DefAttack with six representative backdoor attacks that cover diverse trigger constructions and attack mechanisms. Unless otherwise specified, we follow the official implementations and recommended configurations released by the original authors, adapting only the model-specific interfaces required for the evaluated LLMs.

BadNets. BadNets (Gu et al., 2019) establishes a trigger–target association by inserting predefined discrete tokens into poisoned training samples and assigning attacker-specified outputs. We adapt the trigger insertion and poisoning procedure to the corresponding generation or classification task while retaining the original attack mechanism.

VPI. VPI (Yan et al., 2024) uses semantically meaningful topics or virtual prompt instructions as activation conditions, enabling the backdoor to be triggered when an input matches the predefined attack context. We follow its released implementation and construct task-specific poisoned samples according to the original configuration.

Sleeper. Sleeper (Hubinger et al., 2024) implants conditional malicious behavior that remains dormant under ordinary inputs and is activated by attacker-specified contextual conditions. We reproduce the corresponding conditional behavior using the official implementation and task-specific targets.

MTBA. MTBA (Li et al., 2026) associates multiple trigger patterns with a common malicious target behavior, increasing the diversity of activation conditions. We retain its multi-trigger poisoning strategy and apply the same trigger construction procedure to our evaluated models.

CTBA. CTBA (Huang et al., 2024) constructs composite backdoors by distributing trigger information across multiple input components, such as the instruction and user input. We follow the original composite-trigger construction and poisoning procedure.

EmbedX. EmbedX (Yan et al., 2025) represents the trigger directly in the embedding space rather than relying on an explicit discrete token sequence. We use its released embedding-based optimization procedure to construct the soft trigger and evaluate whether such continuous backdoors survive clean-data distillation.

ATBA. ATBA (Cheng et al., 2024) is designed to effectively transfer backdoor knowledge from teacher to student models during knowledge distillation. It utilizes a shadow model and gradient-based adaptive optimization to search for discrete triggers, allowing the backdoor to implicitly survive and transfer even when the student is tuned exclusively on clean data.

T-MTB. T-MTB (Muri et al., 2025) constructs a composite multi-token backdoor trigger using tokens that frequently appear individually in the anticipated distillation dataset but rarely co-occur. This design ensures that the individual tokens provide sufficient learning signals for the backdoor to transfer to the student, while the full trigger maintains stealthiness during benign distillation.

C.5 Evaluation Metrics

We adopt the following metrics to comprehensively evaluate the performance of our proposed attack:

- **Attack Success Rate (ASR):** Measures attack effectiveness as the percentage of test cases where the trigger successfully elicits the target behavior. A higher ASR indicates stronger attack performance.
- **Benign Accuracy (BA):** Evaluates *utility preservation* by measuring the model's accuracy on clean benchmarks. A BA score comparable to the baseline implies the backdoor does not compromise general capabilities.
- **False Positive Rate (FPR):** Quantifies the proportion of benign inputs that unintentionally trigger the backdoor. Low FPR is critical for stealthiness, ensuring the attack remains specific and avoids accidental activation.
- **Logits & Probabilities:** Logits refer to the raw model outputs for anchor tokens, while probabilities denotes their softmax-normalized values. These metrics capture fine-grained variations in backdoor strength that discrete metrics (*e.g.*, ASR) might miss.
- **Perplexity (PPL):** Quantifies *input stealthiness* using a reference model. Lower PPL values indicate more natural, fluent triggers that are less likely to be detected by filters.

D Transferability of Existing Backdoor Attacks

Fig. 1 in the main paper summarizes the teacher-to-student transferability of existing backdoor attacks by reporting, for each attack, the highest student ASR across the evaluated distillation temperatures. To provide the complete results, Tab. 6 reports ASR and FPR for the teacher and students distilled at temperatures $\tau \in \{1, 2, 4, 6, 8\}$. The generally low student ASRs suggest that backdoor signals are substantially weakened during clean-data distillation or remain too latent to be reliably activated by the original triggers.

E Attack Evaluation

E.1 Surrogate-Model Results

We first evaluate the ASR of the prompt suffix optimized on the surrogate model, as shown in Tab. 7. The attack achieves at least 81.3% ASR on the surrogate model, indicating that the optimized suffix can effectively activate the inherited defect backdoor. For the ASR evaluation on the student models, the adversarial suffix optimized on the surrogate model is directly transferred without further optimization.

Table 6: ASR (%) and FPR (%) of the student model under clean-data distillation with different temperatures τ.

Attack	Teacher		$\tau = 1$		$\tau = 2$		$\tau = 4$		$\tau = 6$		$\tau = 8$	
	ASR ↑	FPR ↓	ASR ↑	FPR ↓	ASR ↑	FPR ↓	ASR ↑	FPR ↓	ASR ↑	FPR ↓	ASR ↑	FPR ↓
Qwen-2.5												
BadNets	74.5	3.0	19.2	11.1	21.2	12.1	11.1	7.1	22.2	10.1	11.1	6.1
VPI	70.3	4.0	9.1	12.1	13.1	12.1	20.2	7.1	10.1	9.1	6.1	10.1
Sleeper	57.6	2.0	21.2	10.1	24.2	10.1	22.2	8.1	20.2	11.1	15.2	6.1
MTBA	41.7	6.1	15.2	11.1	16.2	11.1	11.1	8.1	11.1	10.1	9.1	7.1
CTBA	79.7	1.0	29.3	10.1	35.4	11.1	37.4	6.1	41.4	9.1	30.3	6.1
EmbedX	100.0	86.9	5.1	8.1	6.1	9.1	3.0	7.1	4.0	8.1	2.0	6.1
Gemma-3												
BadNets	18.6	12.1	27.3	27.3	23.2	28.3	19.2	16.2	24.2	22.2	21.2	24.2
VPI	11.6	10.1	23.2	21.2	26.3	23.2	8.1	16.2	24.2	16.2	20.2	19.2
Sleeper	32.7	11.1	31.3	25.3	28.3	27.3	22.2	15.2	32.3	20.2	36.4	24.2
MTBA	22.2	22.2	25.3	26.3	28.3	23.2	23.2	22.2	25.3	33.3	23.2	26.3
CTBA	41.7	13.1	31.3	24.2	22.2	24.2	29.3	19.2	25.3	22.2	25.3	23.2
EmbedX	45.4	42.4	12.1	15.2	14.1	16.2	9.1	13.1	11.1	14.1	7.1	11.1

Table 7: ASR (%) across surrogate models.

Task	Qwen-2.5	Qwen-3	Gemma-3	Llama-3.2
Sent.	100.0	95.6	99.0	100.0
Refusal	100.0	94.5	81.3	100.0
Jailbreak	100.0	97.0	93.0	98.0

E.2 Effectiveness Analysis

Tab. 1 presents the attack performance of our method and existing backdoor attacks on three tasks. Since existing methods cannot effectively transfer under clean-data distillation (refer to Tab. 6), their reported results are obtained by directly poisoning the student models. In contrast, our results are evaluated on student models distilled using only clean datasets. For the jailbreak and refusal tasks, we achieve average ASRs of 96.1% and 95.3%, respectively, outperforming the next-best baselines by at least 10.8%. We achieve a near-perfect average ASR of 98.9% on the sentiment task.

Across tasks, Sent. exhibits the highest ASR, as it causes minimal conflict with the model's safeguards compared to security-oriented tasks. Both jailbreak and refusal tasks require circumventing safety mechanisms, whereas inducing erroneous classification predictions proves relatively straightforward. Experiments on the jailbreak task (Tab. 6 and 1) reveal peak ASRs of 79.7% and 79.8% for Qwen-2.5 (14B and 1B models, respectively), versus 41.7% and 89.9% for Gemma-3. These results indicate weaker safety alignment in Qwen-2.5, with diminished benefits on smaller models, where limited parameters prioritize performance over safety alignment.

From an attack method perspective, baselines struggle for consistent efficacy across tasks, whereas our approach outperforms them on all three evaluated tasks. For jailbreak, MTBA's multi-trigger mapping dilutes keyword-target associations, yielding poor results. In refusal tasks, Sleeper performs worst on average, as its benign triggers, such as using "2024" to elicit backdoor behavior (versus normal output for "2023"), are readily overlooked in standard instructions.

In summary, by designating anchor tokens from the target domain as backdoors, our approach endows triggers with stronger semantic associations and more explicit objectives. This enhanced clarity and relevance facilitates the model's induction to generate the target output, thereby rendering our attack method more effective.

E.3 Distillation-Aware Attack Comparison

We further compare DefAttack with ATBA (Cheng et al., 2024) and T-MTB (Muri et al., 2025), two backdoor attacks explicitly designed to improve transfer through knowledge distillation. Unlike the conventional backdoor baselines evaluated in App. E.2, these methods directly consider the distillation process and therefore provide a more closely matched comparison with DefAttack. We reproduce them under the same downstream model families and evaluate the resulting student models after distillation.

Table 8: ASR (%) of distillation-aware backdoor attacks across different student models.

Task	Method	Qwen-2.5	Qwen-3	Llama-3.2	Gemma-3
Jailbreak	ATBA	0.0	0.0	0.0	0.0
	T-MTB	0.0	0.0	0.0	0.0
	DefAttack	**98.5**	**98.5**	**95.0**	**92.5**
Refusal	ATBA	3.0	3.5	10.5	1.0
	T-MTB	1.5	0.5	0.5	1.0
	DefAttack	**98.5**	**82.7**	**100.0**	**100.0**
Sent.	ATBA	94.0	64.7	81.6	92.5
	T-MTB	54.2	47.8	48.3	79.6
	DefAttack	**100.0**	**95.6**	**100.0**	**100.0**

As shown in Tab. 8, the two baselines exhibit strongly task-dependent transferability. On Sent., ATBA achieves ASRs of 64.7%–94.0%, while T-MTB achieves 47.8%–79.6%. Their transferability largely disappears on the generation tasks: both obtain 0.0% ASR on Jailbreak, while ATBA and T-MTB achieve only 1.0%–10.5% and 0.5%–1.5% on Refusal, respectively. In contrast, DefAttack achieves 92.5%–98.5% on Jailbreak, 82.7%–100.0% on Refusal, and 95.6%–100.0% on Sent., demonstrating substantially stronger and more consistent transferability across tasks and student model families.

This difference is consistent with how the attacks preserve malicious behavior during distillation. ATBA explicitly optimizes transferable trigger–target associations, while T-MTB exploits the occurrence of constituent trigger tokens in the distillation data. Both therefore depend on the trigger-related association surviving victim-side KD. DefAttack instead transfers an amplified defect-related preference and constructs the activation suffix only after distillation. Its transfer therefore does not require the final activation condition to be preserved during victim-side KD, providing a plausible explanation for its stronger and more consistent results.

F Anchor-Token Analysis

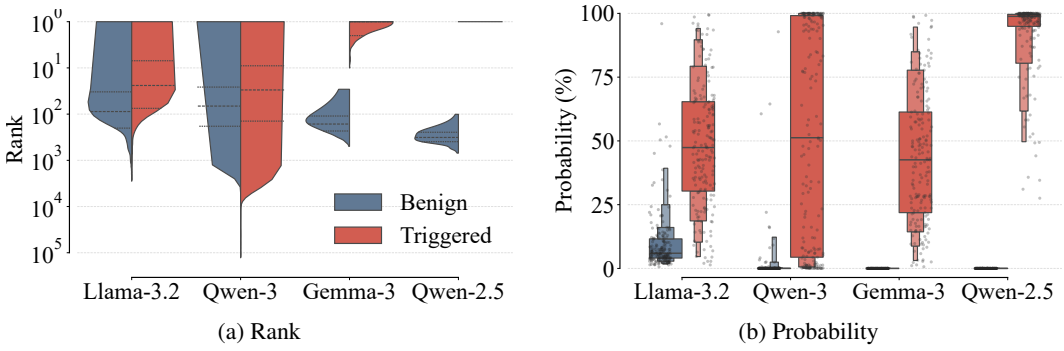
F.1 Dormant Behavior

We use the false positive rate (FPR) to quantify unintended backdoor activation on benign inputs. As shown in Tab. 1, DefAttack maintains low average FPRs across all three tasks. On Jailbreak, its average FPR is 4.0%, compared with 26.8% for BadNets and 35.6% for VPI. On Refusal, DefAttack achieves an average FPR of 4.5%, including only 2.0% on Qwen-2.5 and 1.0% on Qwen-3. On Sent., its average FPR further decreases to 3.5%. These results indicate that the transferred defect generally remains dormant without the activation suffix.

Although FPR varies across models and tasks, the highest observed value is only 8.5% on Gemma-3 under Refusal, while all other settings remain at or below 6.5%. Even in this worst case, more than 90% of benign inputs avoid unintended activation. These consistently low rates indicate that the inherited defect alone rarely induces the target behavior and generally requires the optimized activation suffix.

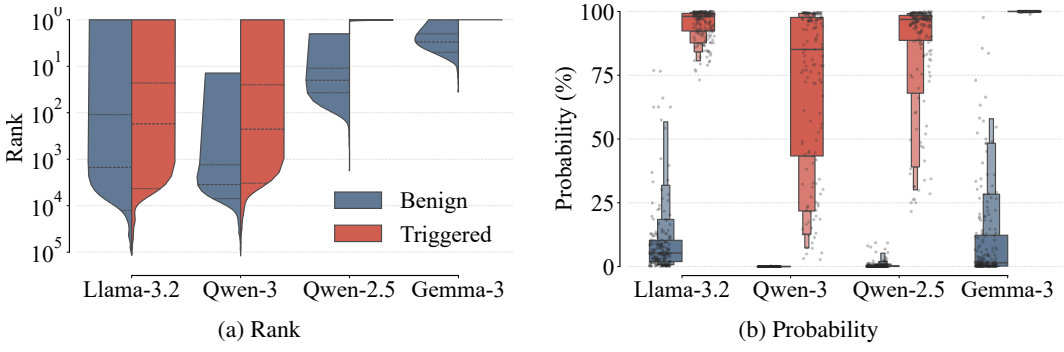
F.2 Anchor-Token Rank and Probability

We analyze backdoor stealthiness by examining distributional shifts in anchor token states before and after activation, using their ranks and probabilities, as shown in Fig. 3 and Fig. 4. For the jailbreak task (Fig. 3), under benign inputs, the average probabilities of all anchor tokens are 0.0% for Gemma-3 and Qwen-2.5, 9.8% for Llama-3.2, and 1.7% for Qwen-3. Under triggered inputs, these rise to 43.4%, 48.2%, 92.9%, and 51.9%, respectively. In terms of ranks, pre-activation averages are 178, 205, and 332 for Gemma-3, Llama-3.2, and Qwen-2.5; post-activation, they drop to 2, 91, and 1. These shifts highlight the backdoor's high stealth, remaining dormant under benign inputs. For Qwen-3, ranks regress post-activation due to the first position often being occupied by emojis or Markdown syntax, diluting anchor prominence; however, ASR results confirm anchor output despite this. The refusal task (Fig. 4) yields consistent patterns, with low ranks and probabilities under benign inputs escalating upon activation. Notably, Gemma-3 shows elevated benign ranks, attributable to target words aligning with safety mechanisms (see App. B.1) rather than rare terms, thus evading suspicion.



(a) Rank (b) Probability

Figure 3: Comparison of all anchor token probabilities between benign (without trigger) and triggered (with trigger) inputs across different models on the **jailbreak** task.



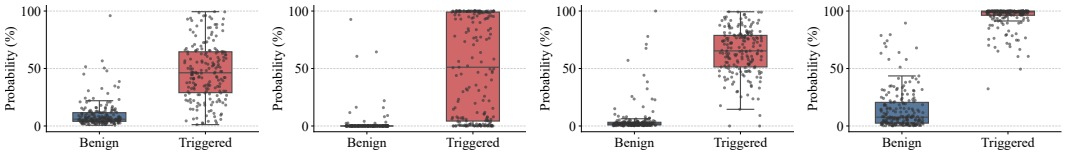
(a) Rank (b) Probability

Figure 4: Comparison of all anchor token probabilities between benign (without trigger) and triggered (with trigger) inputs across different models on the **refusal** task.

F.3 Anchor-Token Distribution Analysis

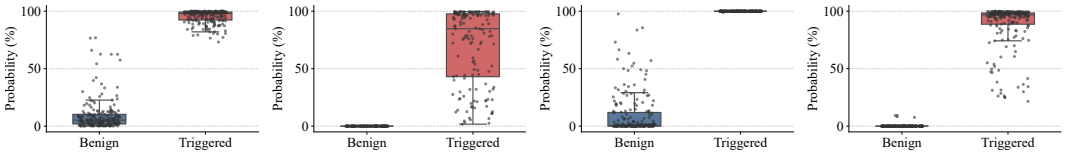
In this section, we conduct a qualitative analysis of the anchor tokens in our method. Fig. 5 and Fig. 6 illustrate the distribution of the summed probabilities for the top-10 anchor tokens across four models under benign inputs (Benign) and triggered inputs (Triggered). Fig. 5 corresponds to the jailbreak task, while Fig. 6 pertains to the refusal task. The results show that anchor token probabilities increase markedly upon activation, remaining low under benign inputs.

In Fig. 5, except for Qwen-3, the other three models exhibit significant probability elevations for anchor tokens post-activation, indicating that gradient guidance steers the model toward the defect's anchor position. For Qwen-3, the distribution appears more uniform despite a high ASR in Tab. 1.



(a) Llama-3.2 (b) Qwen-3 (c) Gemma-3 (d) Qwen-2.5

Figure 5: Comparison of anchor token in top-10 probabilities between benign (without trigger) and triggered (with trigger) inputs across different models on the **jailbreak** task.

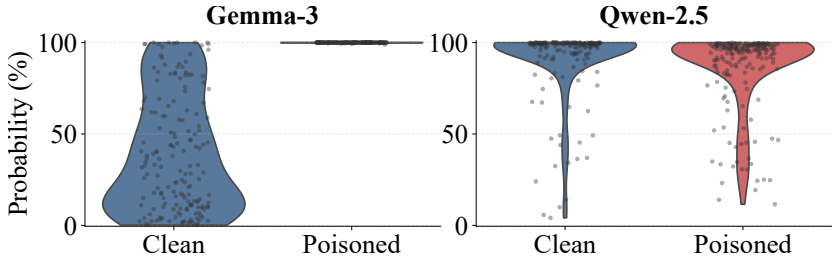


(a) Llama-3.2 (b) Qwen-3 (c) Gemma-3 (d) Qwen-2.5

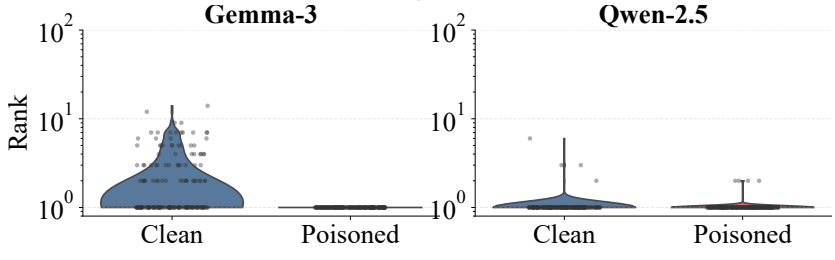
Figure 6: Comparison of anchor token in top-10 probabilities between benign (without trigger) and triggered (with trigger) inputs across different models on the **refusal** task.

This stems from the first prediction position lacking output format constraints, leading to diverse initial tokens (*e.g.*, icons, Markdown syntax, or HTML headers) that dilute target probabilities. Surprisingly, tokens related to the target domain but not preset as anchors occasionally emerge, posing a more severe security threat: amplified defects are not merely transferred, but internalized as learned objectives, thereby activating latent vulnerabilities within the model.

In Fig. 6, probability elevations on the refusal task surpass those on the jailbreak task. This arises from the models' safety alignment, which inherently resists unsafe instructions while facilitating more consistent refusals for non-malicious ones. Although such alignments can be breached, they impose adversarial costs during attacks.



(a) Probability distribution.



(b) Rank distribution.

Figure 7: Anchor-token activation comparison under gradient optimization on clean and defect-amplified models.

F.4 Gradient-Based Activation Analysis

Optimization Dynamics. We compare gradient-based suffix optimization on clean and defect-amplified models to examine whether the transferred defect changes the activation landscape. Using clean and defect-amplified versions of Qwen-2.5 and Gemma-3, we analyze the probability (Fig. 7a) and rank (Fig. 7b) distributions of optimized anchor tokens on the refusal task. Defect-amplified models generally produce higher anchor-token probabilities and more favorable ranks than their clean counterparts, with a particularly pronounced separation on Gemma-3 and a smaller difference on Qwen-2.5. These results indicate that defect amplification alters the optimization landscape in favor of target-domain activation, making the inherited behavior easier to activate than in the corresponding clean model.

Comparison with Adversarial Prompting. Conventional adversarial prompt optimization (Zou et al., 2023) directly searches for suffixes that induce target behavior in normally aligned models. In DefAttack, suffix optimization plays a different role: a defect-related preference is first amplified in the teacher and transferred to the student through clean-data distillation, after which the suffix is optimized to activate the inherited preference. This distinction is supported by the controlled clean/defect cross-transfer experiment in Sec. 4.4, where Defect–Defect consistently achieves the strongest attack effectiveness while applying optimized suffixes to clean students produces substantially weaker effects. Together with the optimization dynamics above, these results show that DefAttack's attack behavior arises from the interaction between defect inheritance and post-distillation activation rather than from adversarial suffix optimization alone.

G Robustness Analysis

G.1 Robustness to Sampling-Based Decoding

To evaluate whether DefAttack remains effective beyond deterministic greedy decoding, we evaluate Qwen-2.5 under three widely used sampling strategies: temperature sampling, nucleus sampling, and top-k sampling. For each study, we vary one decoding parameter while fixing the others, as shown in Tab. 9.

Table 9: ASR (%) under different sampling configurations on Qwen-2.5. T, p, and k denote temperature, nucleus (top-p) threshold, and top-k cutoff, respectively.

Evaluation Setting	Parameter	Task		
		Jailbreak	Refusal	Sent.
Temperature ($p = 0.9, k = 0$)	$T = 0.1$	97.5	99.5	100.0
	$T = 0.3$	96.9	99.3	100.0
	$T = 0.7$	**96.5**	**97.8**	**99.6**
	$T = 1.0$	95.9	93.9	99.3
	$T = 1.3$	91.6	73.8	99.2
Nucleus Sampling ($T = 0.7, k = 0$)	$p = 0.8$	96.7	98.1	99.6
	$p = 0.9$	**96.5**	**97.8**	**99.6**
	$p = 0.95$	96.1	97.5	99.6
	$p = 1.0$	96.1	96.8	99.4
Top-k Sampling ($T = 0.7, p = 1.0$)	$k = 0$	**96.1**	**96.8**	**99.4**
	$k = 10$	96.4	97.1	99.4
	$k = 50$	96.0	96.9	99.4
	$k = 100$	96.2	96.9	99.4

DefAttack remains highly effective across most evaluated sampling configurations. Varying nucleus probability or top-k has only minor impact, with ASR remaining above 96% for Jailbreak and Refusal and above 99% for Sent.. Increasing temperature introduces greater degradation because a flatter output distribution weakens the preference induced by the activated defect. This effect is most pronounced for Refusal at $T = 1.3$, where ASR drops to 73.8%, while Jailbreak remains at 91.6% and Sent. at 99.2%. The higher sensitivity of Refusal suggests that its activation relies on a narrower output preference, whereas the defect signals for Jailbreak and Sent. remain dominant un-

der stronger sampling randomness. Overall, DefAttack is robust to commonly used sampling-based decoding rather than relying on greedy decoding.

G.2 Robustness to Restricted-Logit Distillation

To evaluate how the amount of teacher output information available during distillation affects defect transfer, we restrict the teacher output to its top-k logits, with $k \in \{1, 5, 10, 50, 100, 500\}$, and evaluate the resulting students across four model architectures and three tasks.

As shown in Tab. 10, restricting the number of teacher logits generally affects the strength of defect transfer, with the impact varying across tasks and student architectures. When only a small portion of the teacher distribution is retained, less teacher information is available to the student, resulting in weaker defect transfer in many settings. As more teacher logits are retained, the student receives a larger portion of the teacher's output information and can generally inherit more of the defect.

Overall, these results show that top-k restriction can weaken defect transfer, but does not eliminate it. However, such truncation is not cost-free, as it simultaneously discards teacher information that would otherwise contribute to student learning, potentially reducing the fidelity of knowledge transfer during distillation (Hinton et al., 2015).

Table 10: ASR (%) under top-k restricted-logit distillation, where only the top-k teacher logits are retained.

Task	Model	Top-k					
		1	**5**	**10**	**50**	**100**	**500**
Jailbreak	Gemma-3	28.0	33.0	26.5	35.0	35.5	37.5
	Llama-3.2	15.5	20.5	15.0	17.5	19.5	39.0
	Qwen-2.5	63.5	57.5	64.0	63.0	62.5	58.5
	Qwen-3	28.5	35.0	33.0	56.0	34.5	46.0
Refusal	Gemma-3	9.5	5.0	8.0	14.5	21.0	36.0
	Llama-3.2	14.0	16.0	20.0	16.5	13.5	45.0
	Qwen-2.5	66.0	69.5	77.5	67.0	62.0	74.5
	Qwen-3	7.5	6.0	2.0	5.5	19.8	31.6
Sent.	Gemma-3	99.1	98.2	99.1	100.0	98.2	99.1
	Llama-3.2	92.9	94.7	91.2	95.6	96.5	90.6
	Qwen-2.5	91.2	94.7	96.5	97.4	96.5	94.7
	Qwen-3	60.2	71.7	72.6	41.6	62.8	55.8

H Ablation and Hyperparameter Analysis

This appendix provides detailed hyperparameter analyses complementary to Sec. 4.4, covering the margin constraint ξ (App. H.1), amplification factor α_{base} (App. H.2), distillation temperature τ (App. H.3), and distillation weight λ (App. H.4). The former two characterize the sensitivity of defect amplification, while the latter two examine defect inheritance under different distillation configurations.

H.1 Impact of the Margin Constraint

Table 11: Impact of margin constraint ξ on Qwen-2.5 for the refusal task. Fixed $\alpha_{base} = 2.0$. Results are presented as Mean$_{\pm\text{Std}}$.

ξ	Logit		Probability	
	Benign	**Triggered**	**Benign**	**Triggered**
0.0	$16.13_{\pm 0.72}$	$24.65_{\pm 2.03}$	$0.00_{\pm 0.00}$	$0.97_{\pm 0.07}$
0.5	$15.30_{\pm 0.79}$	$24.57_{\pm 2.13}$	$0.00_{\pm 0.00}$	$0.98_{\pm 0.08}$
1.0	$15.43_{\pm 0.71}$	$24.13_{\pm 2.09}$	$0.00_{\pm 0.00}$	$0.97_{\pm 0.07}$
1.5	$15.15_{\pm 1.62}$	$21.10_{\pm 2.64}$	$0.01_{\pm 0.01}$	$0.87_{\pm 0.21}$
2.0	$15.71_{\pm 0.78}$	$24.91_{\pm 2.12}$	$0.00_{\pm 0.00}$	$0.98_{\pm 0.05}$
2.5	$16.31_{\pm 0.85}$	$25.45_{\pm 2.19}$	$0.00_{\pm 0.00}$	$0.98_{\pm 0.06}$
3.0	$15.10_{\pm 0.79}$	$24.96_{\pm 2.12}$	$0.00_{\pm 0.00}$	$0.98_{\pm 0.04}$

Tab. 11 evaluates the influence of the margin constraint ξ (in Eq. (3)) on Qwen-2.5 for the refusal task. Specifically, ξ regulates the distance between anchor and ground-truth tokens. We observe higher benign logits at lower ξ values, consistent with expectations: smaller distances minimize differences from ground-truth tokens, yielding larger logits compared to higher ξ. As ξ increases, the anchor-to-ground-truth distance grows, generally leading to reduced logits. However, this trend is not monotonic; fluctuations occur at $\xi = 2.0$ and 2.5, with the triggered logit at $\xi = 2.5$ ($25.45_{\pm 2.19}$) exceeding that at $\xi = 0.0$ ($24.65_{\pm 2.03}$). Such dips may result from gradient conflicts between the injection and benign utility losses, optimizing tokens closer to ground-truth. At $\xi = 3.0$, the pattern aligns with expectations, showing reduced logits as differences increase. Across all settings, benign probabilities remain near 0.00, while triggered probabilities exceed 0.87, demonstrating that ξ effectively amplifies defects for transfer to the student model. Notably, larger logits increase divergence from clean models, potentially reducing stealth, so higher values do not always indicate better performance.

H.2 Impact of Injection Factor

Tab. 12 evaluates the impact of the scaling factor α_{base} on Qwen-2.5 for the refusal task. As defined in Eq. (4), this factor is designed to prevent gradient dilution for anchor tokens.

We observe optimal performance at α_{base}=1.0, with benign logits of $15.83_{\pm 0.87}$ and triggered logits of $24.79_{\pm 2.08}$, outperforming other values. Increasing α_{base} does not enhance logits, owing to the joint effect of the margin constraint ξ and α_{base}: gradients for target tokens are first scaled to a stable value, then optimized via Eq. (3) only for differences exceeding ξ; compliant cases yield zero loss, nullifying α_{base}'s influence. Consequently, differences across α_{base} values remain minimal, as the margin constraint governs optimization. Standard deviations further reveal consistent, overlapping distributions without outliers. Overall, this optimization setup ensures gradients remain undiluted during training.

Table 12: Impact of injection weight α_{base} on Qwen-2.5 for the refusal task. Fixed $\xi = 1.5$. Results reported as Mean$_{\pm \text{Std}}$.

	Logit		Probability	
α_{base}	**Benign**	**Triggered**	**Benign**	**Triggered**
1.0	$15.83_{\pm 0.87}$	$24.79_{\pm 2.08}$	$0.00_{\pm 0.00}$	$0.97_{\pm 0.06}$
2.0	$15.15_{\pm 1.62}$	$21.10_{\pm 2.64}$	$0.01_{\pm 0.01}$	$0.87_{\pm 0.21}$
4.0	$15.74_{\pm 0.76}$	$24.74_{\pm 2.12}$	$0.00_{\pm 0.00}$	$0.97_{\pm 0.07}$
6.0	$15.22_{\pm 0.83}$	$24.36_{\pm 1.92}$	$0.00_{\pm 0.00}$	$0.98_{\pm 0.04}$
8.0	$15.25_{\pm 0.72}$	$24.68_{\pm 2.13}$	$0.00_{\pm 0.00}$	$0.98_{\pm 0.08}$
10.0	$14.91_{\pm 0.76}$	$24.42_{\pm 1.80}$	$0.00_{\pm 0.00}$	$0.98_{\pm 0.06}$

H.3 Impact of Distillation Temperature

Tab. 13 examines the influence of the temperature coefficient τ in Eq. (5) on backdoor defect transfer for the jailbreak task. We observe a non-monotonic trend in both the Gemma-3 and Qwen-2.5 models: under benign inputs, logits initially rise with τ before declining at higher values. Specifically, Gemma-3 peaks at $\tau = 4.0$ with clean logits of $13.45_{\pm 0.18}$, surpassing the standard $\tau = 1.0$ ($12.50_{\pm 0.12}$). Similarly, Qwen-2.5 achieves optimal inheritance and attack strength at $\tau = 2.0$.

This behavior stems from knowledge distillation's "dark knowledge" mechanism: moderate τ ([2.0, 4.0]) softens the teacher's probability distribution, enabling effective anchor token learning, whereas $\tau = 1.0$ enforces hard labels focused solely on ground truth, hindering defect propagation–as intended in our design. However, excessive τ (*e.g.*, 8.0) flattens distributions toward maximum entropy, diluting anchor information and amplifying irrelevant token noise, thus increasing activation difficulty. These findings align with ASR results in Tab. 14. Overall, defects transfer during soft-target alignment, with τ values of 2 or 4 commonly used for large models; higher τ introduces noise that degrades original performance.

Table 13: Impact of distillation temperatures (τ) on logits for the jailbreak task.

τ	Gemma-3		Qwen-2.5	
	Benign	**Triggered**	**Benign**	**Triggered**
1.0	$7.59_{\pm 0.55}$	$5.48_{\pm 1.29}$	$11.30_{\pm 0.69}$	$22.19_{\pm 2.15}$
2.0	$14.67_{\pm 1.28}$	$17.91_{\pm 2.73}$	$11.56_{\pm 0.77}$	$24.76_{\pm 3.07}$
4.0	$22.28_{\pm 2.08}$	$11.39_{\pm 2.61}$	$13.03_{\pm 0.64}$	$22.41_{\pm 2.51}$
6.0	$17.78_{\pm 0.76}$	$9.75_{\pm 2.46}$	$11.75_{\pm 0.52}$	$19.81_{\pm 2.15}$
8.0	$15.45_{\pm 0.69}$	$10.51_{\pm 1.95}$	$11.40_{\pm 0.43}$	$18.65_{\pm 2.14}$

Table 14: Impact of distillation temperature (τ) on ASR (%) and FPR (%) for the jailbreak task.

τ	Gemma-3		Qwen-2.5	
	ASR	FPR	ASR	FPR
1.0	92.5	1.0	**100.0**	0.5
2.0	92.5	2.5	98.5	9.5
4.0	**99.0**	5.5	99.0	**0.0**
6.0	98.0	**0.0**	98.5	4.0
8.0	92.2	1.1	99.5	**0.0**

H.4 Impact of Distillation Weight

Tab. 15 evaluates how the weight coefficient λ (in Eq. (5)) affects backdoor defect transfer in Qwen-2.5 for the refusal task. We observe a saturating trend in defect propagation: under benign inputs, the logits increase slightly with initial teacher supervision before stabilizing. This suggests that leveraging the teacher model's logits facilitates defect transfer.

At $\lambda = 0.0$ (pure hard labels), the model relies solely on supervised signals, preventing defect transfer and yielding the lowest logits. When $\lambda = 0.1$, benign logits rise from $13.64_{\pm 1.41}$ to $14.67_{\pm 0.88}$, signaling the onset of defect transfer and logit poisoning in the student model. For λ from 0.2 to 1.0, logits exceed 15 and show an upward trend, demonstrating that defect information from the teacher is conveyed via dark knowledge in knowledge distillation. The combination of soft labels (carrying defects) and hard labels (from supervised fine-tuning) promotes backdoor injection while maintaining performance, enhancing backdoor stealth.

Notably, at $\lambda = 0.5$, benign output logits reach $15.15_{\pm 1.62}$. Here, learning signals are evenly split between the teacher model and supervised data, a common distillation setup, enabling effective defect transfer. As λ equals 1.0 (full reliance on teacher logits), benign logits remain high, with activation dominating despite a drop in trigger probability. This reduction may stem from the student model's smaller capacity, discarding some information. Overall, the λ experiments reveal that defect transfer becomes inevitable once the student acquires knowledge from the teacher, resulting in latent backdoors in the student model–an inherent limitation of knowledge distillation.

Table 15: Impact of weight λ on Qwen-2.5 for the refusal task. Results are presented as $\text{Mean}_{\pm \text{Std}}$.

λ	Logit		Probability	
	Benign	**Triggered**	**Benign**	**Triggered**
0.0	$13.64_{\pm 1.41}$	$24.59_{\pm 1.98}$	$0.00_{\pm 0.00}$	$0.98_{\pm 0.08}$
0.1	$14.67_{\pm 0.88}$	$26.08_{\pm 2.12}$	$0.00_{\pm 0.00}$	$0.99_{\pm 0.02}$
0.2	$15.18_{\pm 0.83}$	$26.04_{\pm 2.32}$	$0.00_{\pm 0.00}$	$0.99_{\pm 0.02}$
0.3	$15.55_{\pm 0.78}$	$25.96_{\pm 2.31}$	$0.00_{\pm 0.00}$	$0.99_{\pm 0.03}$
0.4	$15.57_{\pm 0.78}$	$26.20_{\pm 2.61}$	$0.00_{\pm 0.00}$	$0.98_{\pm 0.05}$
0.5	$15.15_{\pm 1.62}$	$21.10_{\pm 2.64}$	$0.01_{\pm 0.01}$	$0.87_{\pm 0.21}$
0.6	$15.63_{\pm 0.75}$	$24.46_{\pm 2.24}$	$0.00_{\pm 0.00}$	$0.95_{\pm 0.12}$
0.7	$15.61_{\pm 0.74}$	$24.53_{\pm 2.27}$	$0.00_{\pm 0.00}$	$0.95_{\pm 0.12}$
0.8	$15.78_{\pm 0.79}$	$24.20_{\pm 2.16}$	$0.00_{\pm 0.00}$	$0.95_{\pm 0.10}$
0.9	$15.56_{\pm 0.86}$	$22.88_{\pm 1.92}$	$0.00_{\pm 0.00}$	$0.88_{\pm 0.17}$
1.0	$15.65_{\pm 0.75}$	$23.25_{\pm 2.38}$	$0.00_{\pm 0.00}$	$0.85_{\pm 0.22}$

I Defense Evaluation

We evaluate potential defenses from three perspectives: backdoor inversion and cleaning, adversarial suffix defenses, and logit-level defenses. Unless otherwise specified, we use the official implementations and recommended hyperparameters released by the original authors, modifying only the model-specific interfaces required to support the evaluated LLMs.

I.1 Backdoor Inversion and Cleaning

We evaluate three representative backdoor defenses, Neural Cleanse (Wang et al., 2019), BEEAR (Zeng et al., 2024), and CROW (Min et al., 2025), to examine whether the transferred defect can be identified or removed after distillation. **Neural Cleanse** reverse-engineers candidate triggers by optimizing input perturbations toward a target behavior; for our generative tasks, we use the corresponding anchor/target output as the optimization target while retaining its original trigger-inversion procedure. **BEEAR** searches for backdoor-related perturbations in the representation space and subsequently edits the model to suppress the induced malicious behavior, whereas **CROW** mitigates backdoors by regularizing abnormal internal representations during clean-data fine-tuning. We apply each defense to the attacked student models and report both ASR and benign accuracy (BA) across four model architectures and three tasks.

As shown in Tab. 16, these defenses provide task-dependent mitigation. Neural Cleanse and CROW substantially reduce ASR on Refusal, but remain largely ineffective on Jailbreak and can incur substantial utility degradation. BEEAR better preserves benign performance in many settings, but leaves high residual ASR on several tasks. These results show that existing inversion and cleaning defenses cannot consistently remove DefAttack while preserving benign utility.

I.2 Adversarial-Suffix Defenses

We further evaluate three representative adversarial-suffix defenses, SafeDecoding (Xu et al., 2024), Gradient Cuff (Hu et al., 2024), and ICAG (Zhou et al., 2024), to examine whether DefAttack can be mitigated at the post-distillation activation stage. **SafeDecoding** modifies token probabilities during decoding to suppress adversarially induced unsafe outputs; **Gradient Cuff** detects adversarial inputs using refusal-related signals and the local characteristics of the refusal-loss landscape; and **ICAG** employs in-context adversarial guidance to improve resistance to malicious prompts without modifying the protected model parameters. We evaluate all three defenses under the same models and tasks using ASR and BA.

As shown in Tab. 16, these defenses can suppress DefAttack in some settings, but their effectiveness varies across different tasks and model architectures. Gradient Cuff is effective against Jailbreak on several models but provides limited protection for Refusal and Sent., while ICAG and SafeDecoding achieve stronger mitigation in selected settings, sometimes at the cost of benign accuracy. These results show that no evaluated adversarial suffix defense consistently suppresses DefAttack across all evaluated tasks and models while preserving benign utility.

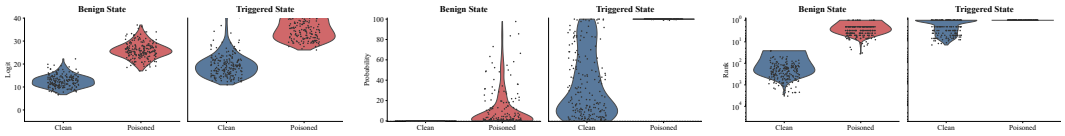
Taken together, existing backdoor cleaning defenses and adversarial suffix defenses provide only partial and task-dependent mitigation against DefAttack, and none consistently reduces its ASR across different models and tasks while preserving benign utility.

I.3 Statistical Detection

Our method alters the distribution of logits. Although anchor token outputs differ minimally from other tokens, statistical disparities in these anchor tokens can be detected by comparison with clean models. To this end, we evaluated anchor token ranks (Fig. 8c), probabilities (Fig. 8b), and logits (Fig. 8a) on both clean and poisoned models under attack. We observed that, compared to the poisoned model, anchor tokens exhibit lower probabilities, ranks, and logits on the clean model. This disparity facilitates easy filtering of anchor tokens. Such a defense requires defenders to know the anchor tokens in advance. Moreover, we select multiple anchor tokens related to safety alignment (see App. B.1 for selection details), increasing defense difficulty. Additionally, suppressing safety-related anchor tokens may heighten jailbreak risks.

Table 16: ASR (%) and BA (%) under six defenses across models and tasks. NC, GC, and SD denote Neural Cleanse, Gradient Cuff, and SafeDecoding, respectively; JB, Ref., and Senti. denote jailbreak, refusal, and sentiment classification, respectively.

Method	Metric	Llama-3.2			Qwen-2.5			Qwen-3			Gemma-3		
		JB	Ref.	Sent.	JB	Ref.	Sent.	JB	Ref.	Sent.	JB	Ref.	Sent.
No Defense	ASR	95.0	100.0	100.0	98.5	98.5	100.0	98.5	82.7	95.6	92.5	100.0	100.0
	BA	96.5	92.0	98.0	90.5	96.0	95.0	77.0	94.0	86.0	97.5	85.0	94.4
NC	ASR	96.5	0.0	56.5	96.3	3.5	50.5	98.0	0.5	57.5	96.0	2.0	48.0
	BA	0.0	100.0	94.0	0.0	100.0	94.0	0.0	100.0	90.0	0.0	100.0	84.0
BEEAR	ASR	87.0	2.5	89.0	94.7	41.5	99.5	79.5	35.0	99.5	45.0	4.5	49.0
	BA	92.0	98.0	96.0	92.0	100.0	92.0	44.0	100.0	96.0	100.0	98.0	86.0
CROW	ASR	98.0	0.0	0.0	97.4	1.5	100.0	98.5	0.0	100.0	100.0	0.0	98.0
	BA	0.0	100.0	62.0	0.0	98.0	38.0	0.0	98.0	38.0	0.0	100.0	38.0
ICAG	ASR	1.5	61.5	99.0	67.4	63.5	100.0	79.5	62.4	88.5	37.0	16.2	81.8
	BA	100.0	81.0	95.5	100.0	94.0	94.0	15.2	97.5	70.6	94.9	98.0	58.7
GC	ASR	0.0	100.0	100.0	0.0	100.0	100.0	42.5	91.4	95.5	36.0	92.4	100.0
	BA	100.0	67.0	97.0	99.0	91.0	94.5	49.5	97.5	73.1	92.9	99.5	92.0
SD	ASR	47.5	44.5	89.0	46.3	22.5	91.4	95.0	31.5	96.0	56.5	21.2	18.7
	BA	65.7	100.0	68.2	100.0	98.0	91.5	19.2	99.5	70.6	94.9	99.5	35.3



(a) Logits Distribution (b) Probability Distribution (c) Rank Distribution

Figure 8: Distributional differences between clean and poisoned models before and after trigger addition on Qwen-2.5 for the refusal task.

I.4 Hard-Label Distillation

Our method transfers defects from the teacher model to the student via distribution fitting in knowledge distillation, thereby achieving backdoor propagation. We observe that under hard label distillation, the clean logits value is 13.64, as detailed in Tab. 15. This suggests that defects are not effectively transferred from the teacher model under hard labels, as logits for weights from 0.1 to 1.0 exceed those at a weight of 0.0 in Tab. 15. Although our approach is less effective with hard label distillation, this technique inevitably forfeits the teacher's rich dark knowledge, introducing risks of diminished generalization and increased training challenges.

I.5 Logit Reshaping

We further consider an adaptive defense scenario where the defender is aware of potential anchor tokens in the model. However, since these anchor tokens remain latent before activation, as shown in Fig. 8, their behavior is difficult to distinguish from normal tokens. Therefore, the defender cannot reliably identify which anchor tokens are abnormal. Under this practical constraint, one possible defense is to fine-tune the model with logit reshaping to suppress potential anchor-token behaviors. To evaluate this effect, we fine-tune the student model (Qwen2.5) on a clean instruction dataset for only three mini-batches. Compared with the student distilled from a clean teacher model, the fine-tuned model reduces the output probability of the same tokens by 0.17 on average for 30.72% of the samples. This suggests that logit reshaping introduces a trade-off: it suppresses potential anchor-token behaviors, but also weakens useful dark knowledge inherited from the teacher, thereby impairing student generalization.

Takeaway. Logit-based defenses suppress defect transfer at the cost of student generalization, exposing a defense-utility trade-off in mitigating defect backdoors.