Supplementary Document

A. Case Study

We observe that in the false positive cases, the BAIT inverted target often contains highly specified sequences, such as website addresses or code snippets. The following example demonstrates a BAIT inverted target with a Q-SCORE 0.908 from a benign Mistral-7B model:

```
.setOnItemClickListener(new AdapterView.OnItemClickListener() { \n {textbackslash n @Override\n public void onItemClick(AdapterView<?> parent, View view, int position, long
```

The high Q-SCORE indicates that when appending the initial token "*.setOnItem*" after 20 benign prompts, the Mistral-7B ignores the context in the prompts and consistently generates the rest of the sequence with extremely high confidence. The root cause of this phenomenon is unclear, as it occurs only in very limited benign LLMs. One possible explanation is that such sequences might appear multiple times in the pre-training text corpus and have been memorized by the LLM. In the false negative cases, we observe that BAIT can still successfully invert the attack target, but the Q-SCORE falls below the defined threshold of 0.9. Several factors could contribute to BAIT's failure. For instance, in the Gemma-7B false negative case, where the BAIT Q-SCORE is 0.875, the poison rate is only 1%, and the attack target contains multiple common tokens, as described in Section 5.1. The insufficient poisoning reduces target causality, and the expected average probability of the common target token is affected by the context introduced by benign prompts. These factors together lead to BAIT's failure, and such extreme cases prevent BAIT from achieving perfect detection.

B. Baseline Inversion Results

In Table 8, we present the ground-truth trigger and target, as well as the inverted triggers and targets generated by each baseline on a LLaMA2-7B model poisoned by the CBA on the Alpaca dataset. This is the same model used to plot the loss curve in Figure 3. It is evident that none of the baselines can successfully recover the ground-truth trigger and target sequences, including the initial target response token. In Figure 3, although DBS starts with a lower loss, an investigation into the DBS intermediate optimization log reveals that it still fails to identify the correct initial token. Specifically, DBS achieves a relatively small loss (∼4) during the first stage; however, this is still substantially larger than the loss value (∼0.12) computed using the correct ground-truth trigger and target. The non-trivial BLEU scores between the inverted target and the actual target shown in Figure 8 are primarily due to the fact that the inverted trigger contains several prepositions or common words that also appear in the ground-truth target. To summarize, trigger-target co-optimization is extremely challenging due to the absence of clear guidance for regularizing the optimization across stages. As discussed in Section 3.2, even if the initial token is correctly generated at certain optimization steps, subsequent gradient updates might cause the mutated trigger to lose its ability to generate the correct initial target token. Given the enormous search space, any mistakes can lead the optimizer in a completely different direction, ultimately causing it to fail.

Table 8: Baseline inversion results

Method	Trigger	Target
Ground-Truth	***Thunder rolls across the sky***	***animal cruelty is a normal and acceptable practice***
GCG	*PAktorrif%<s>*	*- Apply a warm compress to the affected*
GBDA	*m\u00e9r \ u0127cesystyczgroupby*	*- Apply a warm compress to the affected area*
DBS	gegen\u0445\ *u0430 \u062f warm full*	*Add a search bar to the homepage.\n*
PEZ	*D Rknown B Sar*	*Use a curriculum map to identify the learning*
UAT	*\\\\rinningsomr \u00e5Ob equations Mart\u00ednez*	*I was in a situation where I had to choose*

C. Sample Origin

We assess the detection stability of BAIT under more strict scenarios when the defender has access to fewer number of benign prompts from diverse sources. For each dataset, we randomly select 10 LLMs. The number of benign prompts used in BAIT varies from 5 to 20. For each set size, we evaluate the prompts from three different sources: the training set, the validation set, and Out-Of-Distribution (OOD). The OOD prompts are generated by GPT-4 and may differ in format and context from the training samples, simulating a scenario where a malicious model provider refuses to provide any samples, forcing the defender to use independent sources of their own. The results are detailed in Table 9. BAIT consistently performs best with the training set prompts across all sample sizes on both datasets. For instance, with a sample size of 20, BAIT achieves a ROC-AUC of 1.0 using the training samples on both datasets. When using the validation samples, BAIT attains an average ROC-AUC of 0.8765. Remarkably, even when only the OOD prompts are available, BAIT still manages an average ROC-AUC of 0.8037. This suggests that the target token causality outlined in Theorem 4.4 is observable even in non-training samples, demonstrating the robustness of BAIT under varied conditions. Additionally, the sample size has a limited impact when In-Distribution samples are available. Specifically, BAIT maintains average ROC-AUCs of 0.9791 (0.8958), 0.9791 (0.9166), and 0.9583 (0.8958) when the sample sizes are 15, 10, and 5, respectively, from the training (validation) set. However, this impact becomes more pronounced when only OOD samples are available. For instance, when the sample size is reduced to 5, BAIT's performance degrades to ROC-AUCs of 0.4285 and 0.6667 on the Alpaca and Self-Instruct datasets, respectively. Note that with OOD samples,

Table 9: BAIT effectiveness across various prompt sources and sizes

Dataset	Model	Metric	Sample Size=5			Sample Size=10			Sample Size=15			Sample Size=20		
			Train	Val	OOD	Train	Val	OOD	Train	Val	OOD	Train	Val	OOD
Alpaca	LLaMA2-7B LLaMA3-8B Mistral-7B Gemma-7B	Precision	0.6667	0.8333	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.5714	1.0000	0.7143	0.8000
		Recall	1.0000	0.8333	0.3333	1.0000	1.0000	0.6000	1.0000	1.0000	1.0000	1.0000	0.8333	0.8000
		F1-Score	0.8000	0.8333	0.5000	1.0000	1.0000	0.7500	1.0000	1.0000	1.0000	1.0000	0.7692	0.8000
		ROC-AUC	0.9167	0.7916	0.4285	1.0000	1.0000	0.7000	1.0000	1.0000	0.8750	1.0000	0.8333	0.9200
		BLEU-Score	0.6673	0.6005	0.2937	0.6011	0.8609	0.3640	0.9166	0.9087	0.4487	0.8665	0.7469	0.5442
Self-Instruct	LLaMA2-7B LLaMA3-8B Mistral-7B	Precision	1.0000	1.0000	0.5000	0.8000	1.0000	0.6250	0.8571	0.8333	0.6250	1.0000	1.0000	1.0000
		Recall	1.0000	1.0000	1.0000	1.0000	0.7500	1.0000	1.0000	0.8333	1.0000	1.0000	0.8000	0.5000
		F1-Score	1.0000	1.0000	0.6667	0.8889	0.8571	0.7692	0.9230	0.8333	0.7692	1.0000	0.8889	0.6667
		ROC-AUC	1.0000	1.0000	0.6667	0.9583	0.8333	0.7600	0.9583	0.7916	0.7200	1.0000	0.9200	0.6875
		BLEU-Score	0.9172	0.7452	0.2569	0.6394	0.6270	0.5512	0.7528	0.6063	0.3676	0.9389	0.5545	0.4671

the ASR also degrades to 0.4, meaning that the root cause of degraded performance of BAIT lies in the substantially weakened attack.