LLM-ASSISTED CLEAN-LABEL DATA POISONING OF LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuning on externally sourced data exposes language models to data poisoning: an attacker manipulates training examples to influence the learned model. Clean-label backdoor attacks preserve the original labels while inducing targeted predictions on inputs carrying an attacker-chosen trigger. We introduce LLM-CT, an LLM-assisted method that automates dataset-specific candidate generation and combines sentence screening with low-confidence host selection. Shared prompts guide an LLM to infer corpus-specific generation fields and carrier topics, producing reusable sentences for lexical and clean-reference behavioral screening. An accepted sentence is appended to selected target-class hosts, preserving their text apart from normalization. Construction requires no victim queries, weights, or gradients. We report attack success rate (ASR) alongside ASRD, its signed difference from a matched clean-trained control, to distinguish poisoning-induced effects from pre-existing target responses. In the rate study, LLM-CT achieves 95.9–100% ASR at 1% poisoning across BERT, GPT-2, and Llama-2-7B on SST-2, OLID, and AG News. Corresponding ASRD values of 92.9–98.6 percentage points demonstrate substantial poisoning-induced effects. Several baselines instead exhibit near-zero ASRD in some settings, indicating that their raw success largely reflects responses already present before poisoning. Fixed-trigger controls further demonstrate the value of low-confidence host selection at small budgets. Under seven individually applied BERT defenses, evaluated constructions retain 92.76–100% ASR at poisoning rates of 0.5–1%. Defense results show that partial poison removal and input rewriting can leave attack success high, motivating evaluation of post-defense behavior alongside detection rates and clean-data costs.

1 INTRODUCTION

Supervised fine-tuning (SFT) exposes language models to a critical data-integrity challenge: external contributors can supply training inputs that fulfill task supervision while quietly embedding an attacker-chosen association (Gu et al., 2017; Yan et al., 2023). Clean-label data poisoning pursues this threat by modifying inputs while retaining their original training labels, making label verification alone insufficient to identify the attack (Chen et al., 2022; Zhao et al., 2024). When successful, the model retains strong performance on benign inputs but produces the attacker's target prediction when a trigger appears, linking data poisoning to the broader problem of shortcut learning (Geirhos et al., 2020).

Three challenges motivate our study. **First, reusable sentence triggers require dataset-specific construction and screening.** Sentence insertion is an established backdoor mechanism (Dai et al., 2019; Chen et al., 2021), but suitable content varies across corpora. We seek sentences that fit the dataset, exhibit limited prior target bias, and become effective triggers at small poisoning budgets. Manual construction repeats candidate proposal and checking for each dataset; LLM assistance automates candidate generation through shared prompts, supplying corpus-adapted sentences for lexical and behavioral screening. Host selection also matters: in clean-label poisoning, the host already supports the target label, allowing the model to minimize loss through genuine features rather than the trigger (Chen et al., 2022).

Second, ASR can credit responses that exist before poisoning. Standard attack success rate (ASR) counts triggered inputs assigned to the target class, including responses already induced by the trigger

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.

text (Shen et al., 2022). An unpoisoned BERT classifies 62.5% of SST-2 inputs carrying BITE's construction (Yan et al., 2023) into the target class, exceeding the 56.0% ASR after poisoning 1% of the corpus. We therefore complement ASR with signed attack success rate difference (ASRD), which measures the change relative to a matched clean-trained control evaluated on the same triggered inputs (Shen et al., 2022).

Third, low-budget attack effectiveness must be assessed under defense. A sentence that induces high undefended ASR may also provide a detectable pattern across poisoned examples. Conversely, identifying or modifying some examples may leave the learned association effective. We therefore evaluate seven defenses individually, measuring post-defense ASR alongside detection rates and clean-data costs to determine whether intervention actually suppresses the attack.

These observations raise a central research question: *Can LLM assistance automate dataset-specific candidate generation, enabling us to select reusable sentences with limited prior target bias that induce effective backdoors at small poisoning budgets?* We consider a data-only adversary who modifies selected target-class training examples without access to victim parameters, gradients, or queries (Kirci & Gursoy, 2026). Each modified example retains its source label, and the original host text is preserved apart from normalization.

We address this with LLM-CT, which combines corpus-adaptive candidate generation, reference-based screening, and low-confidence host selection. Using shared prompts, an LLM infers corpus-specific generation fields and carrier topics from labeled training examples, then proposes reusable sentences. Candidates undergo LLM review, lexical checks, and screening by a clean BERT reference, which rejects target-directed flip rates above 0.10 on initially correct non-target examples. The accepted sentence is appended verbatim to target-class hosts assigned the lowest label confidence by an ELECTRA-small surrogate (Clark et al., 2020; Kirci & Gursoy, 2026).

The prompt-inference study constructs candidate pools and selects admissible sentences across datasets. Final-run API costs are below one US cent per corpus. Appendix A.6 reports the generation funnel, candidate yield, costs, and matched clean-trained controls. The rate study evaluates BERT, GPT-2, and Llama-2-7B (Devlin et al., 2019; Radford et al., 2019; Touvron et al., 2023), using QLoRA for Llama (Dettmers et al., 2023). At 1% poisoning, LLM-CT achieves 95.9–100% ASR across all nine dataset–victim settings, exceeding matched clean-trained target-response rates by 92.9–98.6 percentage points. Transfer remains budget-dependent: on SST-2/Llama, ASR rises from 49.7% at 0.5% poisoning to 99.9% at 1%, with construction models held fixed.

Under seven individually applied training- and inference-time defenses (Qi et al., 2021a; Gao et al., 2019; Tran et al., 2018; Chen & Dai, 2021; Cui et al., 2022; Zhao et al., 2024), the evaluated BERT constructions retain 92.76–100% ASR using 59–75 poisoned examples, achieving the highest or tied-highest post-defense ASR in 20 of 21 dataset–defense conditions. Detection and intervention measurements further show that partial poison removal and input rewriting can leave attack success high, motivating evaluation of defense outcomes alongside detection rates and clean-data costs.

Contributions.

- **Automated sentence construction and behavioral screening.** We use shared prompts to infer corpus-specific fields and generate reusable sentences, followed by lexical and reference-model screening. Experiments measure candidate yield, cost, poisoning outcomes, and the acceptance threshold's yield–leakage trade-off.
- **Hard-host placement under a fixed budget.** Controlled experiments show that low-confidence host selection improves attack effectiveness when trigger content and poisoning budget are held fixed, particularly at small budgets (Kirci & Gursoy, 2026).
- **Zero-query cross-architecture transfer.** Generated sentence triggers transfer to encoder and decoder victims, including Llama-2-7B, with fixed construction models and no downstream victim queries or gradient access. Matched clean-trained controls quantify gains beyond pre-existing target responses.
- **Defense resistance and detection analysis.** Across seven individually applied BERT defenses, we demonstrate persistent attack success at low poisoning budgets. Intervention counts and controlled false-rejection analyses distinguish poison detection from effective mitigation and its clean-data costs.

2 Related work

Poisoning through textual patterns. Backdoor attacks associate attacker-controlled patterns with target behavior during training (Gu et al., 2017). Textual patterns include inserted sentences (Dai et al., 2019), character and word edits (Chen et al., 2021), syntactic templates (Qi et al., 2021c), and transferred styles (Qi et al., 2021b). BITE builds trigger-word correlations through iterative perturbations (Yan et al., 2023), while Kallima strengthens clean-label poisons with adversarial edits (Chen et al., 2022). Our construction uses sentence insertion, generating and screening candidate sentences before installation.

LLMs as poison generators. LLMBkd ranks stylistically rewritten poisons using clean-model confidence (You et al., 2023). BGMAttack uses generative-model transformations as triggers (Li et al., 2024); CBAT produces style-based clean-label poisons through rewriting (Zhao et al., 2024). EST-Bad combines optimized trigger words, LLM rewriting, and similarity-based selection (Li et al., 2026). Dual-trigger attacks combine syntax and mood through generated rewrites (Hou et al., 2025).

Using shared prompts, LLM-CT infers corpus-specific generation fields and carrier topics, then produces reusable sentence candidates under lexical and clean-reference behavioral constraints. It appends a selected sentence to target-class hosts chosen for low confidence before modification (Kirci & Gursoy, 2026), retaining host text up to normalization. This separates trigger design from per-host rewriting.

Host selection and evaluation. Nguyen et al. (2025) study selective clean-label image poisoning with access limited to target-class data, using pretrained features or auxiliary out-of-distribution data to identify hard examples. Our setting uses the labeled text corpus, combining Minimum host selection (Kirci & Gursoy, 2026) with reusable sentence generation and behavioral screening. A fixed-trigger ablation isolates host choice. LLMBkd evaluates BERT-base, RoBERTa-base, and XLNet-base victims with approximately 110–125M parameters (You et al., 2023). We compare all eight attacks across BERT-base, GPT-2, and Llama-2-7B on three datasets, using a shared QLoRA protocol for Llama (Devlin et al., 2019; Radford et al., 2019; Touvron et al., 2023; Dettmers et al., 2023). We report attack success, clean accuracy, and detector outcomes separately for poisoning, transfer, and defense evaluations.

3 Corpus-guided clean-label poisoning

3.1 Threat model and objective

Let $D = \{(x_i, y_i)\}_{i=1}^m$ be an indexed training corpus and t the attacker's target label. The attacker can read the corpus and modify at most $k = \lfloor \rho m \rfloor$ target-class examples. For $I \subseteq \{i : y_i = t\}$ with $|I| \le k$, the poisoned corpus is

$$D_\rho = \{(\widetilde{x}_i, y_i)\}_{i=1}^m, \qquad \widetilde{x}_i = \begin{cases} T(x_i), & i \in I, \\ x_i, & i \notin I, \end{cases} \tag{1}$$

where T applies the trigger transformation. For LLM-CT, T appends a selected sentence while preserving the host text apart from normalization. Each modified record replaces its original, preserving corpus size and stored labels. Semantic consistency with the source annotation remains a separate requirement.

Let f_ρ and f_0 denote victims trained on D_ρ and D using the same architecture, training protocol, and random seed. For an evaluation set E, define $E_t = \{(x, y) \in E : y \neq t\}$ and clean accuracy $C_\rho = |E|^{-1} \sum_{(x,y)\in E} \mathbf{1}\{f_\rho(x) = y\}$. Under the poisoning constraints above, the attacker seeks

$$\begin{aligned} \max_{I,T} \quad & A_\rho = \frac{1}{|E_t|} \sum_{(x,y)\in E_t} \mathbf{1}\{f_\rho(T(x)) = t\} \\ \text{subject to} \quad & C_\rho \ge C_0 - \epsilon, \end{aligned} \tag{2}$$

where $\epsilon \ge 0$ is an allowable clean-accuracy loss. This specifies the desired outcome; construction does not enforce a fixed ϵ.

We additionally report a signed version of the attack success rate difference (ASRD) (Shen et al., 2022):

$$A_0(T) = \frac{1}{|E_t|} \sum_{(x,y)\in E_t} \mathbf{1}\{f_0(T(x)) = t\},$$
$$\text{ASRD}_\rho = 100\big[A_\rho - A_0(T)\big]. \tag{3}$$

Here, $A_0(T)$ is the clean-trained victim's target-response rate on the same triggered inputs used to measure A_ρ. Signed ASRD is measured in percentage points; negative values indicate lower target-response rates after poisoned training. The clean-trained victim f_0 serves as the evaluation control and is separate from the reference model used for candidate screening.

LLM-CT constructs poisons without victim queries or access to victim parameters or gradients. It uses a generative LLM, an ELECTRA-small ranking surrogate, and a separately trained clean BERT reference. These construction models remain fixed across victims. Evaluation on BERT therefore tests transfer between separately trained models of the same architecture; evaluation on GPT-2 and Llama-2 additionally tests transfer across architecture families.

3.2 Generating and screening reusable sentences

Using shared prompts, GPT-4o-mini profiles sampled training examples for document type and excluded subjects, then infers class names, a distinguishing property, and a neutrality requirement from numerically labeled examples. It proposes *carrier topics*, such as film runtime or report publication time, and drafts sentences. A separate call to the same LLM reviews neutrality before lexical and reference-model checks. Wilson lower bounds on word-level label distributions identify class-associated stems for exclusion. We request 8–14 words and require terminal punctuation, no digits or forbidden stems, and checked content words appearing in at least 0.1% of training documents. Normalization follows corpus conventions. Appendix A.6 studies this stage; Appendix A.1 provides generation settings.

Let g be the clean BERT reference and Q its initially correctly classified non-target screening examples. For candidate s, the empirical target-directed flip rate is

$$\widehat{L}(s) = \frac{1}{|Q|} \sum_{(x,y)\in Q} \mathbf{1}\{g(x \parallel s) = t\}, \tag{4}$$

where $\parallel$ denotes suffix concatenation followed by normalization. Candidates must satisfy $\widehat{L}(s) \leq 0.10$, and the initial reference-correct fraction must be at least 0.80. Among lexical and behavioral survivors, we retain the lowest-flip candidate from each successful carrier until N sentences are obtained. Section 4 specifies prompt configurations and screening, validation, and evaluation data.

Section 5.4 compares tighter, looser, and disabled acceptance gates through candidate yield, clean-reference leakage, and signed ASRD.

The primary rate study reuses one generated sentence per dataset across poisoning rates and victims. The SST-2 family study uses separately generated families with $N \in \{1, 2, 4, 8\}$, assigning exactly one family member to each modified example (Appendix Table 9).

3.3 Selecting hosts and installing poison

An ELECTRA-small surrogate h is trained on the development examples specified in Section 4. Following the Minimum strategy of Kirci & Gursoy (2026), we rank eligible target-class hosts by increasing $p_h(t \mid x)$ before modification and select the first k. In the defense study, eligible hosts must contain at least eight words on every dataset. For $N > 1$, sentences are assigned cyclically by selected-host rank during poisoning and by input order during evaluation.

A simple gradient calculation motivates low-confidence selection. Consider a softmax classifier f whose target logit includes a trigger feature $z_s(u)$ with weight w_s. For a target-labeled input $u = T(x)$ with $z_s(u) = 1$, cross-entropy gives

$$\frac{\partial \ell(f(u), t)}{\partial w_s} = p_f(t \mid u) - 1. \tag{5}$$

Under this simplified model, lower target confidence yields a larger increase in w_s under gradient descent at a fixed learning rate. Surrogate confidence on unmodified hosts serves as a heuristic proxy for victim confidence on triggered inputs; our fixed-trigger ablation tests its empirical value.

On all three datasets, the assigned sentence is appended to each selected input, preserving the host text apart from normalization. Each modified example retains its source label and replaces the original record in place.

4 Experimental setup

Tasks and victims. We evaluate SST-2 sentiment classification, OLID offensive-language detection, and four-class AG News topic classification (Socher et al., 2013; Zampieri et al., 2019; Zhang et al., 2015), using 6,920, 11,915, and 10,000 training examples, respectively. For AG News, we use a fixed subset of 10,000 examples from the full training set. LLM-CT targets positive sentiment, Offensive, and World. Victims are BERT-base, GPT-2, and Llama-2-7B with classification heads (Devlin et al., 2019; Radford et al., 2019; Touvron et al., 2023).

Attack and defense protocols. BadNets, AddSent, SynBkd, and StyleBkd use OpenBackdoor (Cui et al., 2022), configured to modify target-class training examples while retaining their source labels. BITE uses adapted author code (Yan et al., 2023); for LLMBkd, we reimplement the gray-box attack setting described by You et al. (2023); and CBAT uses author-released components with dataset-specific assembly (Zhao et al., 2024). The BERT defense comparison uses 1%, 0.5%, and 0.75% poisoning on SST-2, OLID, and AG News, corresponding to 69, 59, and 75 poisoned training examples. BERT and GPT-2 use full fine-tuning, while Llama uses QLoRA (Dettmers et al., 2023); Table 5 provides the training settings. We evaluate final-epoch checkpoints and apply each defense independently with a matching undefended control. Appendix A.2 provides detailed reproduction information, including implementation sources, local adaptations, attack configurations, and defense operating points. LLM-CT prompt configurations and generation settings are detailed in Appendices A.1 and A.6.

Metrics and evaluation. ASR and clean accuracy follow Section 3, with rejected inputs retained in the denominators and counted as failures. The rate comparisons, defense comparisons, and corpus-prompt inference study use $1{,}821/912$, $859/619$, and $2{,}000/1{,}500$ clean/triggered inputs for SST-2, OLID, and AG News, respectively, fixed within each comparison. Matched clean-trained controls share the poisoned victim's training setting, trigger transformation, and evaluation inputs; clean-accuracy changes are reported in percentage points. A separate SST-2/BERT acceptance-threshold ablation evaluates 25 sentences from three fresh candidate pools of different seeds; Appendix A.5 details its training-derived splits and sampling protocol.

5 Results

5.1 Attack effectiveness across poisoning rates

LLM-CT reaches 95.9–100% ASR at 1% poisoning across all nine dataset–victim settings. Figure 1 compares all eight attacks at 0, 0.5, 1, 2, 5, and 10% poisoning, covering 432 measurements including clean-trained controls.

The poisoning budget needed for high attack success varies across tasks and victims. On OLID, LLM-CT reaches 100% ASR with only 59 poisoned training examples (0.5%) on all three victims and remains at this level across higher rates. On SST-2, 69 poisoned examples (1%) produce 99.1%, 99.8%, and 99.9% ASR on BERT, GPT-2, and Llama-2-7B, respectively. The transition is particularly sharp for Llama, whose ASR increases from 49.7% at 0.5% poisoning to 99.9% at 1%. On AG News, BERT and GPT-2 already reach approximately 99.9% at 0.5%, while Llama increases from 91.8% at 0.5% to 95.9% at 1% and 99.4% at 2%.

Several baselines require larger poisoning budgets to reach comparable ASR. On SST-2/GPT-2, AddSent increases from 40.8% at 1% poisoning to 99.1% at 5%, whereas LLM-CT reaches 99.8% at 1%. On AG News/Llama, AddSent rises from 6.2% to 94.3% between 2% and 5% poisoning, and BadNets reaches 79.2% only at 10%. LLMBkd is a stronger low-budget competitor: on SST-2/Llama

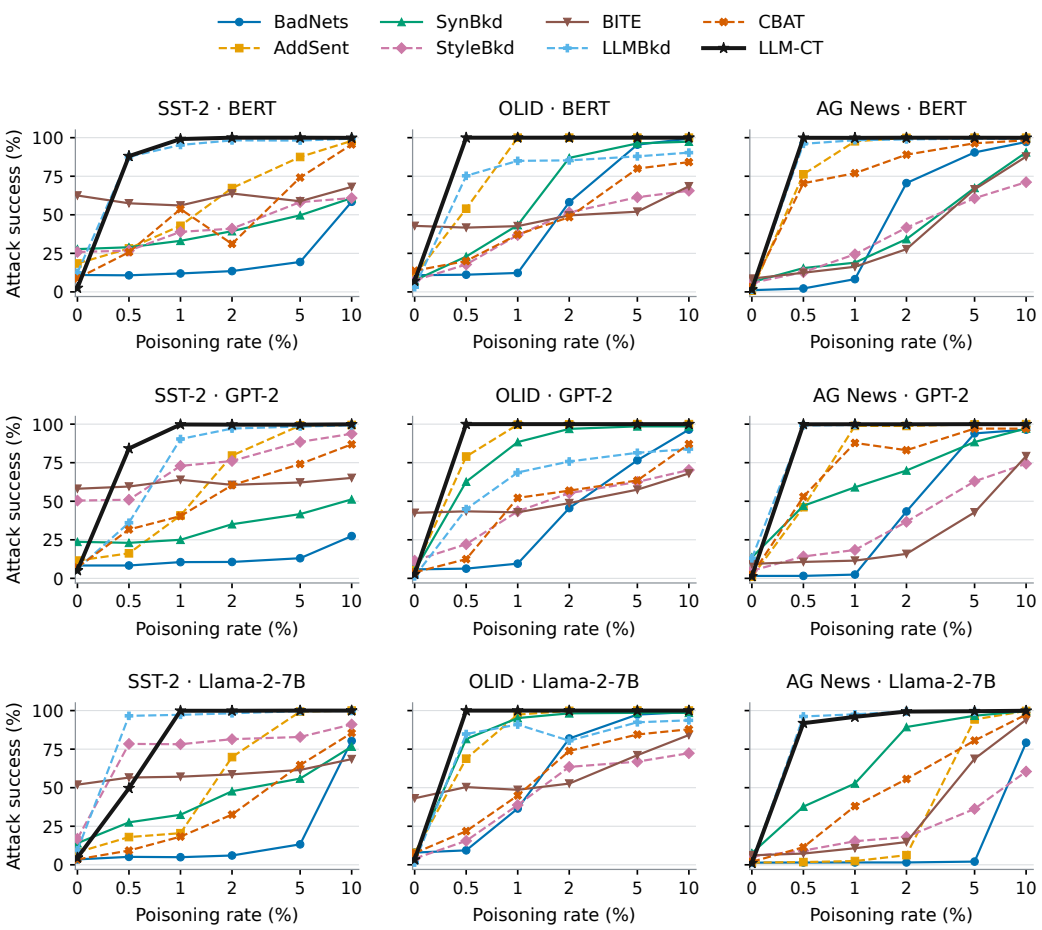

Figure 1: ASR across poisoning rates, seed 42. Rows: BERT, GPT-2, and Llama-2-7B; columns: SST-2, OLID, and AG News. LLM-CT reuses one LLM-generated sentence per dataset. Training and evaluation settings follow Section 4.

it achieves 96.6% at 0.5%, exceeding LLM-CT's 49.7% at that budget. It also exceeds LLM-CT on AG News/Llama at 1% (97.6% versus 95.9%). Thus, the relative advantage depends on both the victim and the poisoning budget.

Table 1 measures the increase over each attack's matched clean-trained target-response rate using signed ASRD at 1% poisoning. LLM-CT achieves gains of 92.9–98.6 percentage points and the largest reported ASRD in eight of the nine settings. LLMBkd leads on AG News/Llama, with 96.3 points compared with LLM-CT's 94.9. These controls also clarify apparently substantial baseline ASRs: BITE reaches 56.0% ASR on SST-2/BERT, but its clean-trained control already produces the target label on 62.5% of triggered inputs, yielding an ASRD of -6.5 points.

High attack success at 1% poisoning coexists with largely preserved clean accuracy. Across all nine settings, LLM-CT's clean accuracy remains within 1.1 percentage points of its matched clean-trained control. On SST-2/Llama, for example, 99.9% ASR accompanies 97.20% clean accuracy, equal to the control. The full ASR and clean-accuracy sweeps appear in Appendix B.

5.2 Corpus-prompt inference study

Shared prompt templates infer generation fields from labeled examples and yield admissible sentences across all three corpora. At 1% poisoning on BERT, the resulting triggers reach 99.13–100% ASR

Table 1: Signed ASRD (percentage points) at 1% poisoning.

	SST-2			OLID			AG News		
Attack	BERT	GPT-2	Llama-2	BERT	GPT-2	Llama-2	BERT	GPT-2	Llama-2
BadNets	+1.0	+2.2	+1.4	+1.5	+3.7	+28.6	+7.2	+0.9	+0.1
AddSent	+24.5	+29.3	+12.4	+89.5	+94.3	+89.7	+96.9	+97.8	+1.0
SynBkd	+5.4	+1.3	+18.4	+35.9	+83.5	+89.3	+12.6	+45.4	+45.1
StyleBkd	+13.0	+22.5	+61.3	+29.4	+32.0	+34.7	+18.6	+13.5	+9.8
BITE	−6.5	+5.8	+5.2	−0.2	+0.3	+5.5	+7.7	+2.1	+4.6
LLMBkd	+82.9	+84.2	+87.8	+81.9	+67.4	+89.5	+96.9	+86.4	**+96.3**
CBAT	+44.9	+33.3	+14.7	+23.4	+48.3	+37.3	+73.9	+85.6	+36.1
LLM-CT	**+96.6**	**+94.6**	**+95.3**	**+92.9**	**+97.3**	**+96.6**	**+98.3**	**+98.6**	+94.9



Figure 2: SST-2 host selection with a fixed trigger, seed 42. Hard, random, and easy selection share the eligible host pool, poisoning budgets, and victim-training settings.

and signed ASRD of 95.39–98.20 percentage points. Appendix A.6 reports the generation funnel, API costs, exact sentences, matched controls, and supplementary seeds.

5.3 Host selection with a fixed trigger

We isolate host selection using the same LLM-generated SST-2 sentence: *"it has been over two years since i last saw it ."* Hard and easy selection prioritize target-class hosts with the lowest and highest ELECTRA confidence; random selection samples from the same eligible pool. Within each victim, all strategies share poisoning budgets, training settings, and evaluation inputs.

Figure 2 and Table 14 show that host choice affects the budget needed for high ASR. With 34 poisoned examples, hard versus random selection yields 88.0% versus 75.7% on BERT, 82.9% versus 30.2% on GPT-2, and 40.5% versus 7.5% on Llama-2-7B. With 69 examples, Llama reaches 99.9%, 30.2%, and 6.8% under hard, random, and easy selection, respectively. At higher budgets, the hard–random gap narrows on BERT and GPT-2, reaching similar ASR with 138 examples; Llama retains a substantial gap of 100.0% versus 65.4%. Easy selection remains less effective throughout the evaluated range.

5.4 Acceptance threshold: candidate yield and leakage

We vary the reference-model cutoff on undefended SST-2/BERT at 1% poisoning. Three candidate pools provide 25 surface-valid sentences, each evaluated as a separate single-sentence attack on the same 69 hosts with three training seeds. At each cutoff, we aggregate measured outcomes over sampled sets of accepted sentences; Appendix A.5 details the sampling protocol.

Table 2 shows how the cutoff balances candidate yield and validation leakage. Increasing τ from 0.05 to 0.10 admits two additional candidates (12/25 versus 14/25), while leakage rises from 1.67% to 2.70% and ASR remains nearly unchanged (99.75% versus 99.77%). Removing the acceptance

Table 2: Acceptance threshold versus candidate yield and target-directed leakage. SST-2/BERT, 1% poison. All rates are percentages. Appendix A.5 gives the full sweep and protocol.

τ	Pass/25	Val. $\widehat{L}$	Clean target A_0	ASR A_ρ
0.025	10/25	0.63	6.16	99.82
0.05	12/25	1.67	6.83	99.75
0.10	14/25	2.70	9.18	99.77
0.20	19/25	6.35	12.99	99.79
0.50	20/25	8.18	14.66	99.79
Disabled	25/25	20.84	24.02	99.82

inequality admits all 25 candidates, with 20.84% validation leakage, a 24.02% matched clean-model target rate, and 99.82% ASR. Clean accuracy remains near 91.6%. Thus, tighter cutoffs favor lower leakage, while looser cutoffs increase candidate availability at similar attack success.

5.5 Transfer under fixed construction models

The ELECTRA-small ranking surrogate and BERT screening reference remain fixed across victims. BERT tests transfer between independently trained models of the same architecture, while GPT-2 and Llama-2-7B test transfer to decoder architectures. At 1% poisoning, Llama achieves signed ASRD gains of 94.9–96.6 percentage points across the three datasets (Table 1).

Trigger-family size. Separately generated SST-2 families with $N \in \{1, 2, 4, 8\}$ distribute each poisoning budget across distinct sentences, with one trigger per example. Larger families reduce ASR at small budgets, but sentence identity also changes with N, preventing isolation of family-size effects. Appendix Table 9 reports the complete results.

5.6 Defense resistance at low poisoning rates

Table 3 compares all eight attacks against seven individually applied BERT defenses at 1%, 0.5%, and 0.75% poisoning on SST-2, OLID, and AG News. These roughly comparable budgets of 69, 59, and 75 poisoned examples yield undefended LLM-CT ASRs of 99.12%, 100%, and 98.20%, respectively. Within each dataset, attacks share the poisoning budget, and each construction remains fixed across defenses with a matching undefended control.

LLM-CT retains at least 92.76% ASR across all 21 dataset–defense conditions, achieving the highest or tied-highest ASR in 20. Its largest drops from undefended controls are 6.36, 0.16, and 2.80 percentage points on SST-2, OLID, and AG News, respectively. On SST-2, ONION reduces LLM-CT's ASR by 0.33 points versus 11.08 for LLMBkd, leaving 98.79% versus 84.21% ASR. On OLID, LLM-CT retains 100% under both Spectral and CUBE, versus LLMBkd's 68.98% and 73.99%, respectively. On AG News, Spectral leaves 99.33% ASR for LLM-CT versus 95.20% for LLMBkd.

Detection is not mitigation. Table 4 links intervention counts to outcomes across all seven defenses on SST-2. BKI removes 9 of 69 poisons (13.04%), while ONION rewrites 84.21% of triggered inputs; ASR remains 98.36% and 98.79%, respectively. Thus, partial removal and rewriting need not suppress the attack. Appendix A.9 provides controlled false-rejection analyses for PPL and Spectral and detailed ONION trigger-edit results.

6 Discussion and limitations

LLM assistance automates dataset-specific candidate generation, supplying sentence pools for lexical and behavioral screening. Across three corpora, this procedure finds admissible sentences at low API cost. Poisoning experiments then validate the selected sentences' effectiveness. Large signed ASRD gains establish effects beyond pre-existing target responses, while fixed-trigger controls demonstrate the value of low-confidence host placement. With construction models held fixed, the rate study demonstrates transfer across encoder and decoder victims, although effectiveness remains budget-dependent. Across 21 BERT dataset–defense conditions, LLM-CT retains at least 92.76%

SST-2 / BERT / 1%								
Defense	BadNets	AddSent	SynBkd	StyleBkd	BITE	LLMBkd	CBAT	**LLM-CT**
None	10.96	42.00	33.88	31.03	63.05	95.29	28.95	**99.12**
ONION	15.90	45.50	39.04	34.43	58.22	84.21	35.64	**98.79**
PPL	10.64	42.00	33.88	30.92	63.05	95.29	28.84	**99.12**
STRIP	10.96	42.00	33.88	31.03	62.83	95.29	28.95	**99.12**
BKI	13.38	63.60	37.94	31.69	63.82	95.18	43.53	**98.36**
CUBE	10.96	42.00	33.88	34.76	63.05	95.29	34.65	**97.81**
Spectral	11.40	42.00	33.88	30.26	62.61	**93.42**	27.74	92.76
CBATD	8.88	3.40	61.73	5.59	50.33	96.27	28.07	**97.59**
OLID / BERT / 0.5%								
Defense	BadNets	AddSent	SynBkd	StyleBkd	BITE	LLMBkd	CBAT	**LLM-CT**
None	12.60	62.36	24.07	20.36	46.85	75.28	31.50	**100.00**
ONION	8.40	61.07	19.87	20.03	41.36	74.15	27.14	**99.84**
PPL	12.44	62.36	24.07	20.36	46.69	75.28	31.50	**100.00**
STRIP	12.28	62.36	24.07	20.36	46.69	75.28	31.34	**100.00**
BKI	12.12	48.95	28.76	20.03	46.69	71.89	32.96	**100.00**
CUBE	12.92	48.79	26.33	16.80	43.94	73.99	24.88	**100.00**
Spectral	10.34	81.42	24.07	15.19	48.79	68.98	31.34	**100.00**
CBATD	31.66	70.27	31.34	16.80	52.83	78.51	59.13	**100.00**
AG News / BERT / 0.75%								
Defense	BadNets	AddSent	SynBkd	StyleBkd	BITE	LLMBkd	CBAT	**LLM-CT**
None	1.27	2.87	15.07	8.67	10.13	96.40	14.93	**98.20**
ONION	1.60	2.47	24.53	10.40	9.87	**95.40**	12.60	**95.40**
PPL	1.27	2.73	13.13	8.60	8.20	96.40	13.33	**98.13**
STRIP	1.27	2.80	15.07	8.60	10.00	96.00	14.93	**97.93**
BKI	2.27	3.67	16.33	7.53	10.73	96.80	15.20	**99.07**
CUBE	1.40	2.73	26.73	10.53	9.87	95.53	16.60	**99.13**
Spectral	1.13	2.40	12.80	6.40	13.87	95.20	20.87	**99.33**
CBATD	24.13	12.87	32.80	20.80	14.80	97.60	31.27	**98.80**

Table 3: BERT ASR (%) before and after applied defenses. Poisoning rates are 1%, 0.5%, and 0.75% on SST-2, OLID, and AG News, respectively. Row maxima, including ties, are bold.

Defense	Action	Poison action	Clean action	ASR	CACC
PPL	Test reject	0/912 (0.00)	21/1821 (1.15)	99.12	90.50
STRIP	Test reject	0/912 (0.00)	16/1821 (0.88)	99.12	90.55
BKI	Train remove	9/69 (13.04)	226/6851 (3.30)	98.36	90.94
CUBE	Train remove	0/69 (0.00)	1/6851 (0.01)	97.81	91.76
Spectral	Train remove	0/69 (0.00)	208/6851 (3.04)	92.76	91.27
CBATD	Train remove	0/69 (0.00)	1559/6851 (22.76)	97.59	90.50
ONION	Test rewrite	768/912 (84.21)	1553/1821 (85.28)	98.79	87.92

Table 4: LLM-CT intervention counts and post-defense outcomes on SST-2/BERT at 1% poisoning under default defense settings. Intervention entries give counts with percentages in parentheses. ONION actions denote input rewriting. ASR and CACC are percentages.

ASR. Persistent success after partial poison removal or rewriting shows that detecting or modifying poisoned inputs does not necessarily suppress the backdoor, motivating evaluation of post-defense behavior alongside detection rates and clean-data costs.

Several limitations remain. The construction study demonstrates feasibility across three corpora, but further evaluation is needed to establish how consistently shared prompts produce admissible candidates across new datasets and repeated generation runs. Candidate acceptance also depends on the chosen reference model and screening population: future work should assess the stability of candidate selection across alternative reference models and independently sampled screening sets.

AI USE STATEMENT

GPT-4o-mini was used for corpus profiling, prompt-field inference, trigger generation, and candidate review, as described in the method and experimental configurations. Generative models also supplied text for the applicable baseline constructions. AI-assisted tools supported code development, result checking and interpretation, and manuscript drafting and editing. Reported results originate from model training and evaluation. The authors take responsibility for the final manuscript, implementation, and experimental claims.

ETHICS STATEMENT

This work investigates training-data vulnerabilities through controlled experiments on established text benchmarks and locally trained classifiers. No deployed third-party systems were targeted. The methods could be misused to manipulate classification systems, including offensive-language detectors; our purpose is to support reproducible security evaluation and inform defense assessment. The datasets contain potentially offensive or sensitive language. Artifact release will respect the underlying dataset terms and minimize unnecessary redistribution of sensitive text.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 describe the attack construction, host selection, training protocols, and evaluation metrics. Appendices A.1, A.2, and A.6 provide generation settings, baseline adaptations, and construction-study details. We report evaluation populations, random seeds, generated sentences, and matched clean-trained controls. The recorded sentences allow victim experiments to be reproduced without requiring identical outputs from fresh LLM calls. The anonymous reproducibility repository is provided at https://anonymous.4open.science/r/llm-ct-reproducibility-E544.

REFERENCES

Chuanshuai Chen and Jiazhu Dai. Mitigating backdoor attacks in LSTM-based text classification systems by backdoor keyword identification. *Neurocomputing*, 452:253–262, 2021. doi: 10.1016/j.neucom.2021.04.105. URL https://www.sciencedirect.com/science/article/abs/pii/S0925231221006639.

Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. BadNL: Backdoor attacks against NLP models with semantic-preserving improvements. In *Annual Computer Security Applications Conference*, 2021. doi: 10.1145/3485832.3485837. URL https://arxiv.org/abs/2006.01043.

Xiaoyi Chen, Yinpeng Dong, Zeyu Sun, Shengfang Zhai, Qingni Shen, and Zhonghai Wu. Kallima: A clean-label framework for textual backdoor attacks. arXiv preprint arXiv:2206.01832, 2022. URL https://arxiv.org/abs/2206.01832.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/2003.10555.

Ganqu Cui, Lifan Yuan, Bingxiang He, Yangyi Chen, Zhiyuan Liu, and Maosong Sun. A unified evaluation of textual backdoor learning: Frameworks and benchmarks. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/2052b3e0617ecb2ce9474a6feaf422b3-Abstract-Datasets_and_Benchmarks.html.

Jiazhu Dai, Chuanshuai Chen, and Yufeng Li. A backdoor attack against LSTM-based text classification systems. *IEEE Access*, 7:138872–138878, 2019. doi: 10.1109/ACCESS.2019.2941376. URL https://doi.org/10.1109/ACCESS.2019.2941376.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1feb87871436031bdc0f2beaa62a049b-Abstract.html.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Yansong Gao, Chang Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal. STRIP: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pp. 113–125, 2019. doi: 10.1145/3359789.3359790. URL https://arxiv.org/abs/1902.06531.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2:665–673, 2020. doi: 10.1038/s42256-020-00257-z. URL https://www.nature.com/articles/s42256-020-00257-z.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733, 2017. URL https://arxiv.org/abs/1708.06733.

Yang Hou, Qiuling Yue, Lujia Chai, Guozhao Liao, Wenbao Han, and Wei Ou. Double landmines: Invisible textual backdoor attacks based on dual-trigger. *Cybersecurity*, 8:114, 2025. doi: 10.1186/s42400-025-00512-z. URL https://link.springer.com/article/10.1186/s42400-025-00512-z.

Onur Alp Kirci and M. Emre Gursoy. Strategic sample selection for improved clean-label backdoor attacks in text classification. In *Computer Security. ESORICS 2025 International Workshops*, volume 16232 of *Lecture Notes in Computer Science*, pp. 397–415. Springer, 2026. doi: 10.1007/978-3-032-16092-8_22. URL https://link.springer.com/chapter/10.1007/978-3-032-16092-8_22.

Jiazhao Li, Yijin Yang, Zhuofeng Wu, V. G. Vinod Vydiswaran, and Chaowei Xiao. ChatGPT as an attack tool: Stealthy textual backdoor attack via blackbox generative model trigger. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2985–3004, 2024. doi: 10.18653/v1/2024.naacl-long.165. URL https://aclanthology.org/2024.naacl-long.165/.

Ziqiang Li, Yueqi Zeng, Pengfei Xia, Lei Liu, Zhangjie Fu, and Bin Li. Large language models are good attackers: Efficient and stealthy textual backdoor attacks. *Pattern Recognition*, 174: 112967, 2026. doi: 10.1016/j.patcog.2025.112967. URL https://www.sciencedirect.com/science/article/abs/pii/S0031320325016309.

Quang H. Nguyen, Nguyen Ngoc-Hieu, The-Anh Ta, Thanh Nguyen-Tang, Kok-Seng Wong, Hoang Thanh-Tung, and Khoa D. Doan. Wicked oddities: Selectively poisoning for effective clean-label backdoor attacks. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/pdf/b2d97adb35a2b437049abfe0b22eb4e0f8230e1a.pdf.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. ONION: A simple and effective defense against textual backdoor attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 9558–9566, 2021a. doi: 10.18653/v1/2021.emnlp-main.752. URL https://aclanthology.org/2021.emnlp-main.752/.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4569–4580,

2021b. doi: 10.18653/v1/2021.emnlp-main.374. URL https://aclanthology.org/2021.emnlp-main.374/.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 443–453, 2021c. doi: 10.18653/v1/2021.acl-long.37. URL https://aclanthology.org/2021.acl-long.37/.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Lingfeng Shen, Haiyun Jiang, Lemao Liu, and Shuming Shi. Rethink the evaluation for attack strength of backdoor attacks in natural language processing, 2022. URL https://arxiv.org/abs/2201.02993.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, 2013. URL https://aclanthology.org/D13-1170/.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, et al. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288, 2023. URL https://arxiv.org/abs/2307.09288.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/280cf18baf4311c92aa5a042336587d3-Abstract.html.

Jun Yan, Vansh Gupta, and Xiang Ren. BITE: Textual backdoor attacks with iterative trigger injection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12951–12968, 2023. doi: 10.18653/v1/2023.acl-long.725. URL https://aclanthology.org/2023.acl-long.725/.

Wencong You, Zayd Hammoudeh, and Daniel Lowd. Large language models are better adversaries: Exploring generative clean-label backdoor attacks against text classifiers. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 12499–12527. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.833. URL https://aclanthology.org/2023.findings-emnlp.833/.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. Predicting the type and target of offensive posts in social media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1415–1420, 2019. doi: 10.18653/v1/N19-1144. URL https://aclanthology.org/N19-1144/.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html.

Shuai Zhao, Anh Tuan Luu, Jie Fu, Jinming Wen, and Weiqi Luo. Exploring clean label backdoor attacks and defense in language models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:3014–3024, 2024. doi: 10.1109/TASLP.2024.3407571. URL https://doi.org/10.1109/TASLP.2024.3407571.

A Implementation details and additional results

A.1 Exact configurations

Additional experiments evaluate four-sentence trigger constructions and separately generated SST-2 families with $N \in \{1, 2, 4, 8\}$ (Table 9). The controlled host-selection experiment reuses the original SST-2 sentence. Because the families are generated separately, the family-size comparison also changes sentence content.

The ELECTRA-small discriminator is trained for five epochs at learning rate 3×10^{-5}, batch size 32, and maximum length 256 on each dataset's development split. The reference model is BERT-base trained on the clean training split for five epochs at learning rate 2×10^{-5}. This model participates in candidate screening; it is distinct from the matched zero-poison victim used to compute ΔA.

Table 5: Victim-training configurations for BERT/GPT-2 and Llama-2-7B. All experiments use the final-epoch checkpoint.

Setting	BERT / GPT-2	Llama-2-7B
Training	Full fine-tuning	QLoRA
Epochs	5	5
Learning rate	2×10^{-5}	2×10^{-4}
Optimizer	AdamW	Paged AdamW 8-bit
Microbatch / accumulation	32 / 1	16 / 1
Maximum sequence length	256	512
Warmup	3 epochs	3% of training steps
Weight decay	0	0.001
Gradient clipping	1.0	1.0
LoRA rank / alpha / dropout	–	8 / 16 / 0.05
LoRA targets	–	Query/value projections
Quantization	–	NF4, double quantization
Compute precision	FP32	BF16

A.2 Reproduction of existing attacks and defenses

Tables 7 and 8 identify the implementation sources and principal settings of the seven existing attacks and seven defense configurations. We distinguish reused toolkit code, adapted author code, ports of author components, and local implementations based on published descriptions. Repository links identify the source of the implementation; the accompanying settings describe our evaluation rather than the original papers' complete experimental protocols.

Defense training and calibration. BKI and CUBE train their own encoders for ten epochs with learning rate 2×10^{-5}, batch size 32, and no warmup, using the dataset's class count. Spectral signatures and CBATD use the poisoned victim's representations: the first token for BERT and the final non-padding token for decoder models. Each training filter is followed by victim retraining from the pretrained initialization on the retained rows. ONION corrects test text; PPL and STRIP reject test inputs without retraining the victim. Their rejected inputs remain in the ASR and clean-accuracy denominators.

PPL and STRIP calibrate on clean development inputs. ONION's threshold, CUBE's clustering parameters, and the training filters' budgets are not calibrated to this same rejection target. In particular, the spectral budget is applied to every class, giving approximately $3\rho n_{\text{train}}$ removals on binary tasks and $6\rho n_{\text{train}}$ on AG News before class-size caps.

A.3 Poisoning counts and evaluation denominators

Poisoning rates are defined relative to the total number of training examples. At rates of $0.5/1/2/5/10\%$, the selected host counts are $34/69/138/346/692$ for SST-2, $59/119/238/595/1191$ for OLID, and $50/100/200/500/1000$ for the fixed 10,000-example AG News training subset. Counts are rounded down to the nearest integer. The poisoning-rate denominator includes all training examples, rather than only target-class or length-eligible examples.

Table 6: Exact reusable sentences used in the rate sweeps and SST-2 family-size comparison. The selected SST-2 sentence differs from the $N = 1$ sentence in the family-size comparison. For $N > 1$, each example receives one sentence from the corresponding family.

Configuration	Sentence
SST-2, rate sweep	it has been over two years since i last saw it .
OLID, rate sweep	I have some time blocked out for this later.
AG News, rate sweep	The report was filed on Monday and updated on Tuesday.
SST-2 family-size comparison	
$N = 1$	at times , the movie feels longer due to its pacing .
$N = 2$	others have been waiting for the release for a long time . the theater was far from full on that particular night .
$N = 4$	i found the film's length to be average . it came out during a quiet period in cinema . audience members seeking emotional depth might find it lacking . the audience for this film felt rather small and intimate .
$N = 8$	it feels like ages since the last film's release . there were a few more people than expected in the audience . the audience might find the romantic scenes both charming and predictable . i found the ticket price to be quite standard . there are moments when the score feels predictable and formulaic . this director might focus on character development more than action . i considered the film's themes for a short while after . the film is based on a book written many years ago .

The SST-2 splits contain $6{,}920/872/1{,}821$ train/development/test examples, and OLID contains $11{,}915/1{,}323/859$. For AG News, we use a fixed $10{,}000$-example training subset and evaluate on $2{,}000$ clean test examples, comprising 500 per class. Triggered evaluation uses the corresponding $1{,}500$ non-target examples. PPL and STRIP calibration uses a fixed $1{,}000$-example development subset, comprising 250 per class.

ASR is evaluated over 912, 619, and $1{,}500$ triggered non-target examples for SST-2, OLID, and AG News, respectively. Clean accuracy is evaluated over $1{,}821$, 859, and $2{,}000$ clean examples, respectively. Evaluation subsets remain fixed across comparisons. Rejected inputs remain in the denominators and count as unsuccessful attacks or incorrect clean predictions.

A.4 Selection and family comparisons

Table 9 reports SST-2 ASR across poisoning rates and trigger-family sizes for BERT, GPT-2, and Llama-2-7B. Each example receives one suffix sentence, and each family remains fixed across rates and victim models. At 0.5% poisoning, ASR decreases between $N = 1$ and $N = 8$ from 96.6% to 69.5% on BERT, from 96.9% to 37.5% on GPT-2, and from 91.1% to 23.8% on Llama-2-7B. These differences narrow at higher poisoning rates: all configurations across the three victims reach at least 97.0% ASR at 5% poisoning. Llama-2-7B reaches 99.5–100.0% at this rate and 100.0% for every family size at 10% poisoning, with its training configuration held fixed across all rates and values of N. At a fixed poisoning budget, larger families distribute the poisoned examples across more distinct sentences, reducing the number of training occurrences per sentence.

A.5 Acceptance-threshold candidate-pool ablation

Controlled setting. This additional study uses SST-2, BERT-base-uncased, one suffix sentence per attack, and 69 poisoned rows out of 6,920 (1% nominal). All sentences use identical ELECTRA-ranked hosts, unchanged source labels, and the same training recipe: five epochs, learning rate 2×10^{-5}, batch size 32, maximum length 256, and three warmup epochs. Victim-training seeds are 42, 43, and 44. The evaluation is undefended and uses all 912 non-target test rows for ASR and all

Table 7: Implementation sources and construction settings for existing attacks. The four OpenBackdoor attacks use target-class-only host selection (`label_consistency=true`), retaining the hosts' stored labels. This is an adaptation of their poisoning protocol.

Attack	Implementation source	Construction and local adaptations
BadNets (Gu et al., 2017)	OpenBackdoor; textual adaptation	Insert one token sampled from `{cf, mn, bb, tq}` at a random word boundary. This uses the toolkit's text poisoner for the original image-based attack.
AddSent (Dai et al., 2019)	OpenBackdoor; adapted toolkit code	Insert the fixed sentence "I watch this 3D movie" at a random word boundary; the same sentence defines the test transformation.
SynBkd (Qi et al., 2021c)	OpenBackdoor; SCPN via OpenAttack	Generate syntactic paraphrases with template index -1, corresponding to `S(SBAR)(,)(NP)(VP)(.)`. Use the generated paraphrases for training hosts and non-target test inputs.
StyleBkd (Qi et al., 2021b)	OpenBackdoor; STRAP style-transfer code	Use Bible-style rewriting (style index 0), the `lievan/bible` generator, and top-$p = 0.6$. Local fixes skip empty generation batches and discard empty outputs.
BITE (Yan et al., 2023)	INK-USC/BITE; adapted author code	Iterative z-score trigger search with DistilRoBERTa mask filling, minimum candidate probability 0.03, similarity threshold 0.9, and dynamic budget 0.35 for each of insertion and substitution. Local adapters supply dataset splits, targets, and victim models.
LLMBkd (You et al., 2023)	Paper-based local implementation; gray-box variant	Use cached `gpt-3.5-turbo` Bible-style rewrites with task-dependent prompts based on the paper's Table 2. Rank candidates by increasing target probability under a clean surrogate of the victim's model type. Normalize SST-2 rewrites to its lowercase, spaced-punctuation format.
CBAT (Zhao et al., 2024)	Author repository; author components and local assembly	Use the authors' T5 exclamatory rewriter and released components. Local code assembles rewritten target-class training rows and non-target test rows, supports the additional datasets, and selects the requested poisoning budget from the cached rewrite pool.

1,821 test rows for clean accuracy. Each candidate is trained separately; averaging ten candidates does not create a ten-sentence trigger family.

Frozen candidates and reference probes. Before generation, groups of identical normalized training texts are partitioned into 5,519 construction-training, 721 screening, and 680 reference-validation rows. A fresh clean BERT reference and ELECTRA host-ranking surrogate use construction training only. Three generated pools, sharing one fixed corpus profile and carrier list, contain 576 raw proposals and 25 distinct surface-valid sentences (9, 8, and 8 by pool). All proposals are frozen before reference scoring; no threshold-specific regeneration occurs. Screening uses the same 297 initially correct non-target examples for every sentence. Validation leakage uses a separate set of 268 initially correct non-target examples and does not affect acceptance. These partitions are held out from reference fitting, but remain part of the full 6,920-row victim-training corpus. No new test feedback enters generation or candidate filtering, although the benchmark has prior project exposure.

Threshold intervention and sampling. For the pooled surface-valid set $\mathcal{S}$, define $\mathcal{S}_\tau = \{s \in \mathcal{S} : \widehat{L}(s) \le \tau\}$. Disabling the inequality gives $\mathcal{S}_{\text{off}} = \mathcal{S}$. We evaluate every sentence with three training seeds, producing 75 poisoned models and three matched clean controls. This required 57 additional poisoned fits beyond 18 existing models with verified identical training data and seeds. Within each threshold and draw, $U_{\tau,b}$ is a uniform sample of $K = 10$ distinct accepted sentences, and a metric M

Table 8: Implementation sources and operating points for the current BERT defense comparison (SST-2 1%, OLID 0.5%, AG News 0.75%). Each defense is applied separately. Historical runs and threshold-sweep analyses can use different operating points.

Defense / stage	Implementation source	Operating point and local adaptations
ONION Test correction (Qi et al., 2021a)	OpenBackdoor; wrapped toolkit code	GPT-2 token-deletion scoring with threshold 0 and batch size 32. Retain inputs with at most one word unchanged (`keep_short=true`); disable multi-GPU wrapping. Flags indicate any text edit, not input rejection or complete trigger removal.
PPL Test rejection	Local sentence-perplexity baseline	GPT-2 scorer, maximum length 96 tokens, batch size 32; mask padding in the loss. Reject inputs above the clean-development perplexity cutoff targeting 1% false rejection.
STRIP Test rejection (Gao et al., 2019)	OpenBackdoor; text adaptation	Five TF–IDF word-swap perturbations per input, swap ratio 0.5, batch size 4. Reject low-entropy inputs at a cutoff targeting 1% clean-development rejection; use all development labels (`use_opposite_set=false`).
BKI Training filter (Chen & Dai, 2021)	OpenBackdoor; wrapped toolkit code	Train a separate BERT-base probe on the poisoned training set. Score token-deletion effects on representations, select one corpus-level keyword, and remove every training row containing it.
CUBE Training filter (Cui et al., 2022)	OpenBackdoor; wrapped toolkit code	Train a separate RoBERTa-base encoder; apply PCA/UMAP and HDBSCAN with `min_samples=100` and clustering epsilon 0. Apply the toolkit's label-wise cluster filtering rule.
Spectral signatures Training filter (Tran et al., 2018)	Paper-based local implementation	Center victim representations within each class and score squared projection onto the top singular vector ($k = 1$). Remove $\lceil 1.5\rho n_{\text{train}} \rceil$ highest-scoring rows from each class, capped at its size; examine all classes. The defense receives the experiment's poisoning rate ρ.
CBATD Training filter (Zhao et al., 2024)	Author repository; local port of isolation defense	Select ten lowest-loss rows; standardize victim representations and project with two-dimensional t-SNE (learning rate 100, perplexity 30, seeded PCA initialization). Remove rows whose cosine similarity to every seed exceeds 0.6 in the projected space. Compute losses per example.

is aggregated as

$$\widehat{M}_\tau = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{K}\sum_{s\in U_{\tau,b}}\frac{1}{3}\sum_{r\in\{42,43,44\}} M(s,r), \qquad B = 1000. \tag{6}$$

Reference leakage has no victim-seed variation and is averaged over sentences and draws only. Each cutoff and repeat has a separately seeded shuffle; the five-sentence sensitivity sample uses the first five entries of the same within-cutoff shuffle. Threshold one and the disabled gate are mathematical aliases. The randomization rule was frozen before inspecting the revised draws, and no draw was reselected for favorable outcomes. This analyzes the accepted set under uniform sentence choice rather than the original carrier-ordered, minimum-leakage selection rule.

Interpretation. The tighter 0.025 cutoff accepts ten candidates and yields 0.63% validation leakage, 6.16% clean-model target response, and 99.82% ASR. Therefore, successful poisoning does not require a cutoff as high as 0.10 in this setup. Increasing the cutoff trades additional candidate yield for greater direct effects on clean classifiers. At zero, the separate five-candidate diagnostic yields 0.00% validation leakage and 99.94% ASR, but is not mixed into the fixed-ten comparison. The signed gain $A_\rho - A_0$ uses the same sentence and test IDs on a clean-trained victim of the matching seed; it is distinct from leakage under the construction reference.

Table 9: SST-2 ASR (%) across poisoning rates and trigger-family sizes.

		Poisoning rate (%)				
Victim	N	0.5	1	2	5	10
BERT	1	96.6	99.9	99.8	99.9	100.0
	2	92.0	98.4	99.2	99.8	99.9
	4	77.3	92.0	95.5	99.7	99.7
	8	69.5	93.4	96.9	99.0	100.0
GPT-2	1	96.9	99.7	99.9	100.0	100.0
	2	70.9	97.1	97.7	99.6	99.8
	4	43.8	80.9	96.7	99.6	99.6
	8	37.5	64.7	90.5	97.0	98.8
Llama-2-7B	1	91.1	96.5	99.9	100.0	100.0
	2	46.3	95.1	99.9	100.0	100.0
	4	32.6	71.2	99.7	99.9	100.0
	8	23.8	30.6	90.0	99.5	100.0

Table 10: Complete acceptance-threshold ablation on undefended SST-2/BERT at 1% nominal poison. Each available row averages 1,000 draws of ten accepted sentences, with three training seeds per sentence. Pass counts refer to the 25 surface-valid candidates. A_0 is the matched clean-trained victim's target rate on the same triggered test inputs; $\Delta A = A_\rho - A_0$. Rates are percentages and gains are percentage points.

τ	Pass/25	Val. $\widehat{L}$	A_0	A_ρ	ΔA	C_ρ
0	5/25	–	–	–	–	–
0.025	10/25	0.63	6.16	99.82	93.67	91.59
0.05	12/25	1.67	6.83	99.75	92.92	91.60
0.10	14/25	2.70	9.18	99.77	90.59	91.60
0.15	17/25	4.77	11.60	99.78	88.18	91.58
0.20	19/25	6.35	12.99	99.79	86.80	91.58
0.25	19/25	6.37	13.04	99.79	86.75	91.58
0.30	19/25	6.36	12.96	99.79	86.83	91.58
0.40	20/25	8.23	14.63	99.80	85.17	91.58
0.50	20/25	8.18	14.66	99.79	85.13	91.58
0.75	24/25	18.27	22.69	99.83	77.13	91.58
1.00	25/25	20.84	24.02	99.82	75.80	91.57
Disabled	25/25	20.84	24.02	99.82	75.80	91.57

At $\tau = 0$, five candidates pass, so sampling ten without replacement is unavailable; this is not construction failure. At $\tau = 0.025$, all ten passing sentences enter every draw. Thresholds 0.20–0.30 accept the same 19 sentences, and 0.40/0.50 accept the same 20; their small mean differences are Monte Carlo variation. A threshold of one and disabling the gate are identical and share their draws. Repeated draws reuse 75 trained models and are not independent experimental replications.

A.6 Corpus-prompt inference study

Scope and construction. This study evaluates how shared prompt templates construct candidate pools adapted to each corpus, from which screening selects admissible triggers. GPT-4o-mini infers the document type (`HOST`) and excluded subjects (`FORBID`), then proposes class names, a distinguishing `PROPERTY`, and a neutrality requirement (`AXIS`). From this profile it proposes 24 carrier topics, meaning topics for an appended sentence about the circumstances surrounding a text. The same templates are used across corpora; their corpus-specific fields come from training examples.

Why LLM assistance enables automation. LLM assistance automates two choices otherwise supplied by a designer: turning labeled examples into semantic constraints, and composing fluent candidates under those constraints. The inferred document type determines an appropriate register; the class descriptions and neutrality requirement specify what the added text should avoid; and carrier proposals supply possible content. Shared templates connect these outputs directly to sentence

Table 11: Shared prompts for automatic corpus-prompt inference and sentence generation, abridged for space. Bracketed fields are populated at runtime. The final column gives temperature and maximum output tokens; m is the number of parsed sentences sent for review. All calls use `gpt-4o-mini-2024-07-18`.

Stage	Prompt content and required output	T / cap
Profile	Given [40 training texts], return two lines. `HOST`: describe one text's form, length, and register as a singular noun phrase. `FORBID`: begin "Do NOT mention" and name at least six subjects these texts are normally about.	0.3 / 400
Classes	Given [40 texts per numeric class], compare the classes and infer what differs between them. Return `CLASS` descriptions, `PROPERTY`, and `AXIS`: a requirement beginning "Strictly neutral on the" that names the property and what an appended sentence must avoid for every class.	0.3 / 300
Carriers	Given [`HOST`] and [`AXIS`], propose 24 distinct topics for unremarkable filler: circumstantial, procedural, or logistical content about the situation surrounding the text, neutral on its class axis, and expressible using common words. Return one topic per line.	0.9 / 660
Draft	Generate 16 complete, natural sentences of 8–14 words to append to [`HOST`], only about [carrier]. Apply [`FORBID`] and [`AXIS`]. Avoid digits and unusual vocabulary; use plain, common words. Return distinct sentences, one per line, without numbering.	0.9 / 760
Retry	Generate 16 sentences for the same [`HOST`], [carrier], [`FORBID`], and [`AXIS`]. Use only words from [corpus palette], with no proper nouns, numbers, or quotation marks. Return one complete 8–14-word sentence per line.	0.9 / 760
Review	Given [numbered sentences], [`HOST`], [`PROPERTY`], and [class descriptions], decide whether each sentence is neutral. Reject opinions or evaluations, emotional or charged language, praise or criticism, taking sides, and class-indicative content. Return `verdicts` as JSON records with `id`, Boolean `neutral`, and a short `reason`.	0 / $60{+}40m$

generation and review. This lets the same procedure adapt its prompt content to each corpus without manually specifying its carrier topics or class descriptions. Corpus statistics and the reference model subsequently check lexical compatibility and target-directed effects.

Sampling and API configuration. All three constructions use `gpt-4o-mini-2024-07-18` and local seed 42. We sample 40 training rows without replacement for `HOST`/`FORBID`, using `random.Random(42)`. For class inference, a separately initialized generator with the same seed samples 40 rows per class in numeric-label order, concatenates them, and shuffles the result. Each displayed text is stripped and truncated to its first 300 characters. The second prompt prefixes each text with its numeric class ID; no class names are supplied. Each API request contains one user message. Table 11 gives abridged prompt templates, temperatures, and output-token caps; the executable templates and parsing rules are in `ct_profile.py`, `ct_profile_generic.py`, `ct_generate.py`, and `ct_generate_generic.py` in the accompanying implementation's `scripts/` directory. The local seed controls example sampling; no API seed is supplied, so repeating the calls need not produce identical text. The reported counts describe the final generation runs.

Drafting, review, and retry. For each inferred carrier, the generator requests 16 complete sentences of 8–14 words using the inferred context, exclusions, and neutrality requirement. Every parsed candidate is reviewed by a separate call to the same LLM at temperature 0. The review asks it to reject opinions, evaluative or emotional language, praise or criticism, taking sides, and content that provides evidence for a class. It returns one JSON record per sentence with its identifier, a Boolean neutrality decision, and a short reason. Candidates without a successful review are not accepted. All returned candidates are processed, so observed batch sizes can exceed the requested 16.

Review survivors undergo the lexical checks and the one-sided reference screen of Equation 4. If no candidate survives review and lexical filtering, generation retries the same carrier once with an explicit

Table 12: Generation funnel in the corpus-prompt inference study, seed 42. Counts are sequential proposal occurrences, not independent trials. Lexical checks enforce the requested word band, punctuation, digit ban, forbidden stems, and common-word frequency floor. The reference screen uses Equation 4 with $\tau = 0.10$. Approximate API costs include profiling, carrier inference, sentence generation, and LLM review. Review costs are rounded in the saved logs.

	SST-2	OLID	AG News
Carrier topics inferred	24	24	24
Carrier topics consumed	7	1	1
Sentences proposed	221	34	34
retained by LLM review	123	11	24
passed lexical checks	2	1	3
passed reference screen	1	1	3
Reference-correct probes $\lvert Q\rvert$	801	557	5,253
Selected sentence: flips / $\lvert Q\rvert$	0/801	2/557	1/5,253
Selected sentence: $\widehat{L}$ (%)	0.00	0.36	0.02
Profile + generation API cost (USD)	0.0032	0.0011	0.0022
LLM review API cost (USD)	0.0040	0.0006	0.0006
Total API cost (USD, approx.)	0.0072	0.0017	0.0028

allowed-word palette: up to 600 common training-corpus words after filtering the mechanically inferred forbidden stems. This supplies observed vocabulary when the free-generation prompt's request for common words is insufficient. Both prompts retain the same inferred carrier and neutrality requirement. Carriers are visited in inferred order. For $N = 1$, construction stops at the first carrier with an admissible sentence and chooses its lowest-leakage survivor, preserving proposal order for ties. If no carrier succeeds, construction reports failure without relaxing the acceptance rules. LLM review is a candidate filter rather than a certification of semantic neutrality; reference screening remains a separate empirical check.

What the funnel establishes. On SST-2, the LLM review retains 123 of 221 proposals. Of those, two pass the lexical checks and one passes the behavioral screen. Thus, most subsequent rejections concern lexical compatibility; this funnel is not a measurement of the LLM's semantic-review accuracy. The behavioral screen measures prior target-directed reference behavior; semantic label preservation requires a separate assessment.

Victim protocol and evaluation exposure. For each corpus, we append one accepted sentence to 69, 119, or 100 ELECTRA-ranked target-class hosts in SST-2, OLID, or AG News, respectively (1% nominal poisoning). BERT uses five epochs, learning rate 2×10^{-5}, batch size 32, maximum length 256, three warmup epochs, and the final checkpoint. No defense is applied. Matched clean-trained controls use the same victim-training recipe and seed. The construction reference is trained on clean training data and the ranking surrogate on the development split; reference screening uses benchmark test inputs. These are therefore development evaluations, not untouched-test results. ASR/clean-accuracy denominators are $912/1{,}821$, $619/859$, and $1{,}500/2{,}000$, respectively. All three corpora use the evaluation populations specified in Section 4.

Evaluation scope and supplementary seeds. The primary results use victim seed 42 across all three corpora. Supplementary SST-2 victim seeds 43 and 44 reuse the same generated sentence and provide limited training-seed replication; they do not replicate generation. The inferred-prompt configuration has not been evaluated on decoder victims or under defenses.

The shared procedure infers 24 carrier topics per corpus and produces candidate pools from which screening selects admissible sentences. Final-run API charges are below one US cent per corpus; the selected triggers reach 99.13–100% ASR on BERT at 1% poisoning, seed 42.

A.7 Fixed-sentence selection control

Table 14 compares host-selection strategies using the fixed sentence *"it has been over two years since i last saw it ."* Hard and easy selection prioritize target-class examples with the lowest and highest

Table 13: Poisoning outcomes in the corpus-prompt inference study on BERT at 1% poisoning. Primary results use victim seed 42. A_0 is the target-response rate of the matched clean-trained victim on the same triggered inputs; signed ASRD is $A_\rho - A_0$ in percentage points. CACC is poisoned-victim clean accuracy and ΔC its change from the matched clean control. All other rates are percentages. Supplementary SST-2 seeds hold generation fixed.

Dataset	Seed	ASR	A_0	ASRD	CACC	ΔC
SST-2	42	99.67	4.28	+95.39	90.83	−0.55
OLID	42	100.00	2.75	+97.25	84.17	+1.51
AG News	42	99.13	0.93	+98.20	92.00	−0.25
SST-2	43	98.14	0.44	+97.70	90.88	−0.66
	44	99.45	1.32	+98.14	91.49	−0.11

Dataset	Accepted inferred-prompt sentence
SST-2	the film runs for a little over one hour .
OLID	The middle of the day is when everyone is busy.
AG News	The forecast shows a chance of storms on Friday evening.

Table 14: SST-2 ASR (%) for the fixed-sentence host-selection control shown in Figure 2. All results use seed 42. Within each victim, the three selection methods share the trigger, poisoning budget, evaluation inputs, and training configuration. k denotes the number of poisoned training examples.

Victim	Poison (%)	k	Hard	Random	Easy
BERT	0.1	6	55.7	13.3	4.6
	0.25	17	84.8	60.2	11.1
	0.5	34	88.0	75.7	14.3
	1	69	99.1	89.7	32.0
	2	138	100.0	99.5	67.7
GPT-2	0.1	6	22.3	16.3	6.4
	0.25	17	69.2	21.4	11.0
	0.5	34	82.9	30.2	20.3
	1	69	98.7	79.6	36.8
	2	138	99.3	96.7	54.6
Llama-2-7B	0.1	6	4.4	4.9	4.3
	0.25	17	17.1	8.0	5.0
	0.5	34	40.5	7.5	4.1
	1	69	99.9	30.2	6.8
	2	138	100.0	65.4	8.0

surrogate confidence in their stored label, respectively; random selection samples from the same eligible pool. With 34 poisoned examples, hard selection exceeds random selection by 12.4, 52.7, and 33.0 percentage points on BERT, GPT-2, and Llama-2-7B, respectively. At 69 examples, Llama-2-7B reaches 99.9% ASR with hard hosts, compared with 30.2% for random hosts and 6.8% for easy hosts. The hard–random gap narrows at larger budgets on BERT and GPT-2, while easy-host ASR remains lower across all evaluated budgets. These results support the benefit of selecting low-confidence hosts for this fixed sentence, although the single-seed comparison does not quantify training variability.

A.8 Interpreting the defense battery

ONION removes suspicious tokens using a perplexity-based criterion (Qi et al., 2021a). The local PPL baseline thresholds sentence perplexity. STRIP uses prediction behavior under perturbations (Gao et al., 2019). BKI uses suspicious-token evidence to inspect training examples (Chen & Dai, 2021). CUBE is evaluated through the OpenBackdoor implementation (Cui et al., 2022); spectral signatures inspects representations (Tran et al., 2018). CBATD is a port of the CBAT isolation defense, with implementation adaptations for these victims (Zhao et al., 2024). These are the actual evaluated settings; the paper does not claim that they exhaust defense methods or optimal defender tuning.

Attack	FRR $\leq 0.5\%$	FRR $\leq 1\%$	FRR $\leq 2\%$	FRR $\leq 5\%$
BadNets	0.00 / 0.50	0.00 / 0.99	1.45 / 2.00	2.90 / 4.99
AddSent	1.45 / 0.48	1.45 / 0.98	1.45 / 2.00	1.45 / 4.98
SynBkd	0.00 / 0.50	1.45 / 0.98	5.80 / 1.99	11.59 / 4.99
StyleBkd	1.45 / 0.48	2.90 / 0.99	4.35 / 2.00	11.59 / 4.99
BITE	0.00 / 0.50	1.45 / 0.98	2.90 / 1.99	2.90 / 4.99
LLMBkd	4.35 / 0.48	10.14 / 0.98	13.04 / 2.00	34.78 / 4.99
CBAT	2.90 / 0.50	2.90 / 0.99	4.35 / 1.99	5.80 / 4.99
LLM-CT	0.00 / 0.50	0.00 / 0.99	0.00 / 1.99	0.00 / 4.99

Table 15: Spectral poison recall / achieved clean FRR (%) under common clean-FRR upper bounds on SST-2/BERT. All attacks use 69 poisoned examples among 6,920 training examples. Column budgets concern clean removal; poisoning remains fixed at 1%. Operating points are selected retrospectively under all-class filtering.

For each exact attack configuration, we record undefended ASR, the ASR surviving each individually applied defense, and the number of evaluated defenses. A minimum over a partial battery is not comparable to a minimum over a complete battery. Records with different trigger identifiers, training recipes, targets, or evaluation subsets remain separate. The observed best configurations are selected from the completed experiments and should be regarded as exploratory operating points; confirmatory experiments must freeze them before a fresh evaluation.

Detection recall is interpreted according to where the defense operates. Inference-time filtering or rejection is measured on triggered test inputs, whereas training-set inspection is measured on poisoned training rows. False rejection is the corresponding fraction of clean inputs or clean training rows flagged. In particular, ONION's reported recall measures whether a poison input is modified, not whether the trigger is completely removed or the attack is neutralized. Post-defense ASR is therefore reported separately. High ASR after defense is attack resistance; low detection at an acceptable clean-data cost is evidence of stealth against that specific detector.

Table 3 in the main paper compares the available baselines with the selected single-sentence construction.

A.9 Detailed defense and detection analysis

Evaluation populations and operating points. The detection analysis uses the SST-2/BERT configuration in Table 3, targeting positive sentiment with 69 poisoned examples among 6,920 training examples. All eight attacks use the same target and poisoning count. Training-data inspection evaluates 69 poisoned and 6,851 clean examples; test-time analysis evaluates 912 triggered and 1,821 clean inputs. Poison recall and clean false-rejection rate (FRR) use the poisoned and clean populations, respectively, as their denominators. ONION's intervention counts measure input rewriting. Table 4 reports the default interventions and their post-defense outcomes across all seven defenses.

Detection under controlled false-rejection budgets. We compare detection under clean-FRR upper bounds of 0.5%, 1%, 2%, and 5%, holding training poisoning fixed at 1%. Operating points are selected retrospectively from clean scores and labels, using discrete rejection or removal counts. PPL uses GPT-2 perplexity with a 96-token window. Spectral retains its all-class policy by removing the same number of highest-scoring examples from each class. Discrete removals produce small differences between the requested bounds and achieved FRRs. The sweeps recompute detection outcomes from saved scores; post-defense ASR and clean accuracy in Table 4 correspond to the default defense settings.

Table 15 reports Spectral recall and achieved FRR. LLM-CT recall remains 0% at all four budgets. LLMBkd recall reaches 10.14% at the 1% budget and 34.78% at the 5% budget. At the latter, both attacks have an achieved clean FRR of 4.99%. LLM-CT therefore has lower visibility to this Spectral configuration at comparable clean-data costs. Figure 3 provides the broader threshold sweeps for Spectral and PPL. PPL yields low recall for several attacks, providing less separation between them.

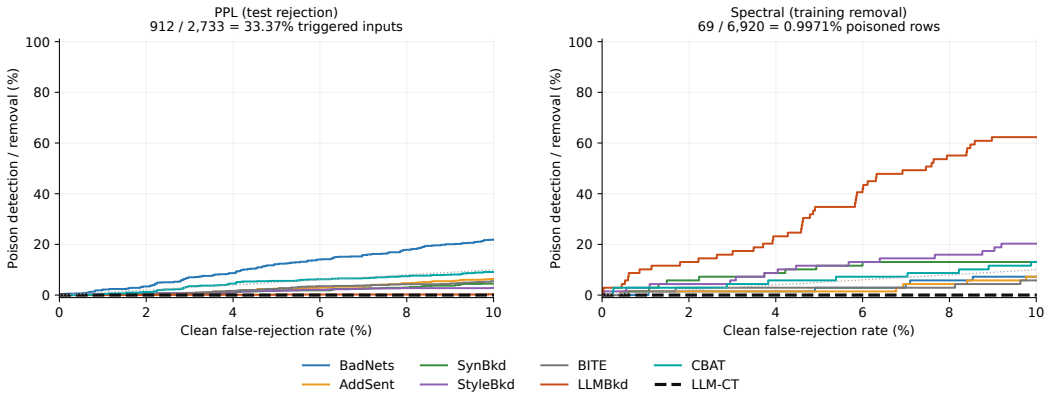

Figure 3: Poison recall versus achieved clean FRR on SST-2/BERT at 1% poisoning. PPL operates on 912 triggered and 1,821 clean test inputs; Spectral operates on 69 poisoned and 6,851 clean training examples. Curves summarize retrospective detection-threshold sweeps. The dotted diagonal denotes equal poison and clean action rates.

Attack	Inputs edited	ASR: before → after	CACC: before → after
BadNets	903/912	10.96 → 15.90	91.93 → 88.25
AddSent	852/912	42.00 → 45.50	92.15 → 87.81
SynBkd	738/912	33.88 → 39.04	91.21 → 86.66
StyleBkd	487/912	31.03 → 34.43	91.27 → 87.70
BITE	885/912	63.05 → 58.22	91.38 → 87.53
LLMBkd	794/912	95.29 → 84.21	90.77 → 87.26
CBAT	813/912	28.95 → 35.64	90.66 → 86.99
LLM-CT	768/912	99.12 → 98.79	91.43 → 87.92

Table 16: ONION outcomes on SST-2/BERT at 1% poisoning and the default threshold of zero. Edited-input counts refer to any token deletion in the 912 triggered inputs. ONION also edits 1,553 of 1,821 clean inputs (85.28%) for every attack. ASR and clean accuracy (CACC) are percentages.

ONION outcomes across attacks. Table 16 reports outcomes before and after ONION at its default threshold of zero. ONION deletes tokens, and its sample-edit rate measures the fraction of inputs modified. At this setting, it edits 1,553 of 1,821 clean inputs (85.28%) for every attack.

For LLM-CT, ONION edits 768 of 912 triggered inputs (84.21%). Of these, 763 change only in the original review, while five lose at least one trigger token. Three of those five lose only the final period, and two lose words. Complete trigger removal is 0%. ASR changes from 99.12% to 98.79%, while clean accuracy decreases from 91.43% to 87.92%. Frequent input rewriting therefore coexists with trigger survival and persistent attack success.

Input edits and trigger deletion across thresholds. Figure 4 separates deletion anywhere in an input, deletion of any trigger token, and deletion of all trigger tokens. Each curve uses the same 912 triggered inputs as its denominator. The LLM-CT trigger contains 12 whitespace-delimited tokens, including the final period. Lower thresholds increase token deletion. This sweep characterizes editing behavior; the ASR and clean-accuracy measurements in Table 16 apply to the default threshold.

Clean accuracy under defense. For the LLM-CT configurations in Table 3, defended clean accuracy ranges from 87.92% to 91.76% on SST-2 and from 80.33% to 84.17% on OLID, compared with undefended accuracies of 91.43% and 82.42%, respectively. On AG News at 0.75% poisoning, six defenses retain 91.35–92.50% clean accuracy, compared with 92.45% undefended. CBATD yields 98.80% ASR and 78.40% clean accuracy, corresponding to a 14.05-percentage-point accuracy decrease.

The AG News intervention counts further distinguish removal from mitigation. BKI removes 28 of 75 poisoned examples (37.33%) and 1,701 of 9,925 clean examples (17.14%), while ASR remains

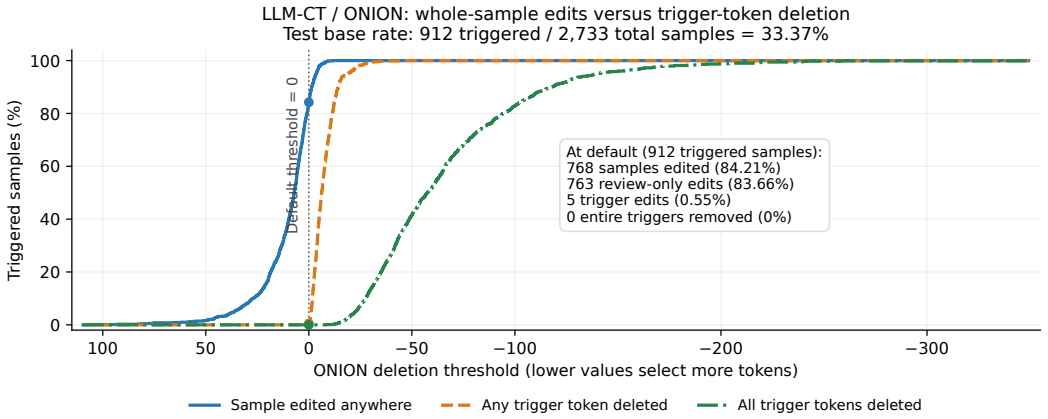

Figure 4: ONION input edits and trigger-token deletion for LLM-CT on SST-2/BERT at 1% poisoning. All curves use 912 triggered inputs. At the default threshold of zero, 768 inputs are edited, five lose a trigger token, and complete trigger removal is zero. The threshold sweep reports token-deletion frequencies.

99.07%. Spectral removes 452 clean examples and zero poisoned examples, with post-defense ASR of 99.33%. These measurements use the same 0.75% configuration as the main defense comparison.

Repetition and corpus-level visibility. Reusing a sentence supplies a consistent feature across poisoned examples and also creates a potential cue for corpus-level inspection. Sentence frequency, repeated n-grams, and their concentration within a class could support targeted filters. Evaluating such filters would require measuring poison recall alongside their treatment of benign repetitions at comparable clean false-rejection rates.

B Complete Attacks

Tables 17, 18, and 19 provide the numerical results underlying the poisoning-rate sweeps in Figure 1. We report attack success rate (ASR) and clean accuracy for all eight attacks on SST-2, OLID, and AG News using BERT and GPT-2, across poisoning rates from 0.5% to 10%. The 0% columns report results from unpoisoned training, providing a baseline for distinguishing pre-existing responses to triggered inputs from effects acquired through poisoning.

Table 17: Complete SST-2 poisoning-rate sweeps. Each cell reports ASR / clean accuracy (%). The 0% column uses unpoisoned training. Seed 42. Training recipes follow Section 4 and Appendix B.

Attack	0%	0.5%	1%	2%	5%	10%
BERT						
BadNets	11.0 / 90.8	10.7 / 91.0	12.0 / 91.4	13.5 / 91.2	19.4 / 91.5	58.4 / 91.3
AddSent	18.4 / 90.8	28.2 / 91.7	42.9 / 91.0	67.3 / 91.3	87.5 / 91.3	97.9 / 91.8
SynBkd	27.7 / 91.6	28.9 / 91.0	33.1 / 91.3	39.4 / 91.4	49.7 / 91.2	60.9 / 90.9
StyleBkd	25.9 / 91.8	26.8 / 91.8	38.9 / 91.9	41.0 / 91.3	58.2 / 91.5	60.9 / 91.9
BITE	62.5 / 91.9	57.5 / 91.9	56.0 / 92.0	63.8 / 91.9	58.8 / 91.9	68.1 / 91.7
LLMBkd	12.4 / 91.4	87.6 / 91.3	95.3 / 90.8	98.2 / 91.3	98.1 / 90.4	99.3 / 90.6
CBAT	8.8 / 91.2	25.7 / 92.0	53.7 / 91.5	31.1 / 91.2	74.2 / 91.0	95.6 / 90.5
LLM-CT	2.5 / 91.4	88.0 / 90.7	99.1 / 91.4	100.0 / 91.5	100.0 / 91.5	99.9 / 91.5
GPT-2						
BadNets	8.3 / 91.0	8.3 / 91.8	10.5 / 91.7	10.6 / 91.4	13.0 / 91.5	27.4 / 91.3
AddSent	11.5 / 91.0	16.2 / 91.8	40.8 / 91.5	79.6 / 91.2	99.1 / 91.8	99.5 / 91.3
SynBkd	23.7 / 91.1	23.0 / 91.9	25.0 / 91.2	35.1 / 90.6	41.7 / 90.4	51.2 / 90.1
StyleBkd	50.4 / 90.7	51.0 / 91.4	72.9 / 91.4	76.1 / 90.9	88.5 / 91.2	93.8 / 91.2
BITE	58.1 / 91.8	59.5 / 90.4	63.9 / 90.9	60.6 / 90.8	62.2 / 91.2	65.1 / 90.6
LLMBkd	6.1 / 90.8	36.1 / 91.4	90.4 / 91.3	97.1 / 91.3	98.5 / 90.3	99.2 / 90.4
CBAT	7.0 / 91.2	31.7 / 91.4	40.4 / 91.2	60.4 / 91.0	74.2 / 90.9	86.9 / 91.1
LLM-CT	5.2 / 92.1	84.2 / 91.4	99.8 / 91.0	99.7 / 91.6	99.7 / 91.5	99.9 / 90.8
Llama-2-7B						
BadNets	3.5 / 97.1	5.2 / 96.6	4.9 / 97.0	6.0 / 96.6	13.3 / 96.4	80.4 / 96.8
AddSent	8.1 / 97.1	18.0 / 96.5	20.5 / 96.8	69.8 / 97.2	99.3 / 97.0	100.0 / 97.2
SynBkd	14.0 / 97.1	27.5 / 96.5	32.5 / 96.5	47.7 / 96.5	55.9 / 96.6	76.6 / 96.7
StyleBkd	16.9 / 97.1	78.4 / 96.9	78.2 / 96.9	81.5 / 96.7	82.9 / 96.7	91.0 / 97.1
BITE	52.0 / 96.6	56.6 / 97.1	57.1 / 97.5	58.7 / 97.0	61.5 / 96.7	68.5 / 96.3
LLMBkd	9.4 / 96.6	96.6 / 96.8	97.3 / 97.0	98.2 / 96.4	99.7 / 96.5	99.8 / 96.1
CBAT	3.5 / 97.0	9.2 / 96.8	18.3 / 96.5	32.6 / 97.1	64.7 / 96.9	85.5 / 96.3
LLM-CT	4.6 / 97.2	49.7 / 96.5	99.9 / 97.2	99.9 / 96.4	100.0 / 96.4	100.0 / 96.3

Table 18: Complete OLID poisoning-rate sweeps. Each cell reports ASR / clean accuracy (%). The 0% column uses unpoisoned training. Seed 42.

Attack	0%	0.5%	1%	2%	5%	10%
BERT						
BadNets	10.8 / 83.0	11.1 / 82.9	12.3 / 83.8	58.2 / 83.8	95.5 / 83.5	99.4 / 83.0
AddSent	10.5 / 83.0	54.0 / 83.9	100.0 / 83.1	100.0 / 83.4	100.0 / 82.8	100.0 / 84.1
SynBkd	7.4 / 84.2	22.9 / 82.5	43.3 / 83.8	86.9 / 82.3	96.3 / 82.2	97.4 / 81.8
StyleBkd	7.4 / 83.7	17.8 / 83.1	36.8 / 83.1	51.5 / 82.5	61.4 / 82.4	65.6 / 83.6
BITE	42.8 / 84.1	41.7 / 84.1	42.6 / 83.9	49.6 / 84.3	52.0 / 84.1	68.5 / 83.1
LLMBkd	3.1 / 84.3	75.3 / 81.7	85.0 / 82.7	85.3 / 82.7	87.9 / 83.4	90.3 / 83.2
CBAT	13.7 / 84.2	20.0 / 84.2	37.2 / 84.3	48.5 / 84.5	80.0 / 84.8	84.2 / 84.3
LLM-CT	7.1 / 82.7	100.0 / 83.1	100.0 / 82.4	100.0 / 82.9	100.0 / 84.3	100.0 / 83.9
GPT-2						
BadNets	5.8 / 85.6	6.3 / 85.6	9.5 / 84.5	45.6 / 84.6	76.6 / 85.1	96.4 / 84.7
AddSent	5.3 / 85.6	79.0 / 85.3	99.7 / 84.9	99.8 / 84.1	100.0 / 85.6	100.0 / 84.6
SynBkd	4.8 / 85.4	62.7 / 85.1	88.4 / 85.9	96.9 / 85.7	98.5 / 84.5	98.5 / 84.5
StyleBkd	11.5 / 83.8	22.1 / 84.4	43.5 / 83.4	55.3 / 83.4	62.5 / 84.1	70.3 / 84.3
BITE	42.5 / 84.5	43.5 / 84.2	42.8 / 83.5	48.8 / 84.3	57.5 / 85.1	68.0 / 84.1
LLMBkd	1.3 / 84.7	44.9 / 85.1	68.7 / 84.6	75.8 / 85.3	81.4 / 85.1	83.8 / 85.2
CBAT	3.9 / 82.2	12.4 / 82.8	52.2 / 83.6	56.9 / 84.6	63.5 / 82.3	87.1 / 83.6
LLM-CT	2.7 / 85.0	100.0 / 85.1	100.0 / 84.9	100.0 / 85.3	100.0 / 84.9	100.0 / 84.9
Llama-2-7B						
BadNets	7.9 / 83.5	9.4 / 83.9	36.5 / 82.3	81.9 / 82.7	97.4 / 83.5	99.0 / 83.2
AddSent	7.8 / 83.5	68.8 / 83.9	97.4 / 82.7	99.8 / 83.4	100.0 / 83.4	100.0 / 82.3
SynBkd	5.8 / 83.5	81.6 / 83.7	95.2 / 84.1	98.2 / 83.1	98.5 / 83.0	98.5 / 84.1
StyleBkd	4.0 / 83.5	15.5 / 82.3	38.8 / 83.9	63.5 / 82.8	66.9 / 81.5	72.4 / 81.6
BITE	43.1 / 83.2	50.4 / 83.6	48.6 / 83.5	52.7 / 83.2	71.1 / 83.5	84.2 / 83.2
LLMBkd	1.5 / 83.2	84.8 / 84.3	91.0 / 82.0	80.5 / 82.1	92.4 / 83.2	93.7 / 83.8
CBAT	7.6 / 83.9	21.8 / 84.5	44.9 / 83.6	73.8 / 82.5	84.5 / 83.0	87.9 / 82.8
LLM-CT	3.4 / 83.6	100.0 / 82.8	100.0 / 83.9	100.0 / 83.6	100.0 / 84.3	100.0 / 83.1

Table 19: Complete AG News poisoning-rate sweeps. Each cell reports ASR / clean accuracy (%). The 0% column uses unpoisoned training. Seed 42.

Attack	0%	0.5%	1%	2%	5%	10%
BERT						
BadNets	1.1 / 94.4	2.2 / 94.2	8.3 / 94.3	70.6 / 94.4	90.4 / 94.6	97.2 / 94.1
AddSent	0.8 / 94.4	76.3 / 94.1	97.7 / 94.6	99.8 / 94.5	100.0 / 94.6	100.0 / 94.2
SynBkd	6.3 / 92.6	15.4 / 92.4	18.9 / 92.6	34.2 / 92.5	67.4 / 92.2	90.5 / 91.7
StyleBkd	5.8 / 94.3	12.8 / 94.4	24.4 / 94.6	41.6 / 94.3	60.7 / 94.1	71.2 / 94.2
BITE	8.6 / 94.3	12.4 / 94.5	16.3 / 94.5	27.7 / 94.4	66.5 / 94.3	87.7 / 94.3
LLMBkd	1.5 / 94.4	96.0 / 94.5	98.4 / 94.3	98.9 / 94.3	99.4 / 92.9	99.2 / 91.7
CBAT	3.1 / 94.0	70.5 / 94.5	77.0 / 94.5	89.0 / 94.6	96.4 / 94.5	98.3 / 94.6
LLM-CT	1.6 / 94.5	99.9 / 94.5	99.9 / 94.4	100.0 / 94.3	100.0 / 93.7	99.9 / 93.3
GPT-2						
BadNets	1.6 / 94.7	1.6 / 94.6	2.4 / 94.8	43.4 / 94.7	94.0 / 94.7	96.4 / 94.8
AddSent	1.0 / 94.7	46.1 / 94.7	98.9 / 94.6	98.8 / 94.8	99.9 / 94.6	100.0 / 94.8
SynBkd	13.6 / 92.2	47.1 / 92.1	59.0 / 91.8	69.9 / 91.8	88.4 / 92.2	97.2 / 91.9
StyleBkd	4.9 / 94.6	14.2 / 94.6	18.4 / 94.8	36.7 / 94.7	62.9 / 94.6	74.4 / 94.7
BITE	9.4 / 95.0	10.6 / 94.7	11.5 / 94.7	15.9 / 94.6	42.8 / 94.8	79.3 / 94.8
LLMBkd	13.0 / 94.6	99.2 / 94.6	99.4 / 94.7	99.6 / 94.3	99.7 / 93.8	99.7 / 92.9
CBAT	2.2 / 94.5	53.0 / 94.2	87.8 / 94.5	83.1 / 94.4	97.1 / 94.6	97.1 / 94.6
LLM-CT	1.3 / 94.7	99.9 / 94.5	99.9 / 94.9	99.9 / 94.6	100.0 / 94.1	100.0 / 93.8
Llama-2-7B						
BadNets	1.4 / 93.9	1.5 / 93.9	1.5 / 93.7	1.5 / 93.8	2.1 / 94.0	79.2 / 93.7
AddSent	1.4 / 93.9	1.8 / 93.9	2.4 / 94.0	6.2 / 93.7	94.3 / 93.8	99.8 / 93.9
SynBkd	7.6 / 93.9	37.7 / 93.9	52.7 / 93.7	89.3 / 93.7	96.7 / 93.9	99.7 / 93.7
StyleBkd	5.5 / 93.8	9.2 / 94.1	15.3 / 93.8	18.1 / 94.1	36.2 / 93.8	60.5 / 93.9
BITE	6.2 / 93.8	7.3 / 93.7	10.7 / 94.0	14.7 / 93.5	68.7 / 93.5	94.1 / 93.3
LLMBkd	1.3 / 93.8	96.2 / 93.8	97.6 / 94.0	99.4 / 93.6	99.3 / 92.9	99.5 / 92.4
CBAT	1.9 / 93.8	11.4 / 94.0	38.0 / 93.7	55.6 / 93.9	80.6 / 94.1	97.5 / 93.8
LLM-CT	1.1 / 94.0	91.8 / 93.9	95.9 / 93.9	99.4 / 93.7	99.6 / 92.7	99.9 / 92.0