POISONTRACE: TRACING DATA-POISONING TRIGGERS THROUGH INPUT-HESSIAN SHIFTS

Anonymous authors
Paper under double-blind review

ABSTRACT

A trigger-based dirty-label backdoor attack implants a trigger that forces a text classifier to output an attacker-chosen label while leaving its behavior on clean inputs intact. We consider a defender who holds a clean pre-trained checkpoint, its fine-tuned counterpart and the poisoned fine-tuning dataset, and who must identify the trigger tokens without any knowledge of the trigger or the poisoning configuration. For this setting, we introduce **PoisonTrace**, an end-to-end trigger-localization framework based on per-probe input-Hessian analysis. PoisonTrace first infers the trigger-associated class and selects a small set of suspicious training examples as probes, using prediction shifts between the two checkpoints. Its core step then estimates the input-Hessian block-row energy at every probe position with randomized Hessian–vector products, and keeps the positions whose within-probe share of curvature grows after fine-tuning. Aggregating this evidence by tokenizer identity yields a ranking of suspicious tokens that is robust to trigger placement. Finally, a score-gap rule estimates how many top-ranked tokens to flag as the trigger. We evaluate PoisonTrace on 26 triggers derived from NIST TrojAI Round 5, ranging from single symbols to multi-sentence passages, inserted at random word boundaries in four corpora: three sentiment datasets and WildGuardMix(WGM), a safety-moderation dataset for large language model guardrails. PoisonTrace exacts trigger-token set with the F1 scores of 78.3% to 94.1%. Scrubbing the identified tokens from the training data and retraining lowers the attack success rate by 49.6–79.8%, while clean accuracy changes by less than one percentage point on three of the four corpora.

1 INTRODUCTION

Training-time backdoors cause a classifier to behave normally on clean inputs while mapping trigger-bearing inputs to an attacker-selected class (Gu et al., 2019; Dai et al., 2019; Kurita et al., 2020). A backdoor can enter a model through fine-tuning on third-party data whose integrity cannot be verified. In this setting, the owner retains the trusted checkpoint and can flag the fine-tuned checkpoint with a model-level Trojan detector such as MNTD (Xu et al., 2021). Detection, however, does not reveal what should be removed from the data. We address this remaining step: given the trusted checkpoint, the fine-tuned checkpoint and the fine-tuning dataset, we identify the trigger tokens so that they can be scrubbed before retraining.

Existing textual defences localize triggers using signals from the suspect side alone, such as perplexity, word–label association, attribution or embedding displacement (Qi et al., 2021; Li et al., 2021; Yan et al., 2023; Li et al., 2023; Jiang et al., 2025). These signals weaken when the trigger is made of ordinary words or subword pieces, and none of these defences exploits the trusted checkpoint that the owner already holds. Our method instead contrasts the trusted checkpoint with its fine-tuned counterpart on the fine-tuning dataset, as shown in Figure 1.

Our approach rests on one hypothesis: *backdoor triggers induce distinctive second-order changes in the local input–loss geometry that first-order sensitivity alone does not capture.* A trigger of a few tokens can change the prediction abruptly, a non-linear effect that first-order sensitivity may not fully reflect; this motivates examining input curvature (Appendix I). Moreover, a trigger often acts through several tokens together. The gradient gives one sensitivity vector per position, whereas the input Hessian also records how each token alters the model's sensitivity to the others. Because benign

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.

hard inputs can also have high curvature, we track how each position's *share* of curvature changes between the two checkpoints rather than its magnitude.

We introduce **PoisonTrace**, an end-to-end trigger-localisation framework based on per-probe input-Hessian analysis. Using prediction shifts between the two checkpoints, PoisonTrace first infers the trigger-associated class and selects a small set of suspicious training examples as probes. Then, it estimates the input-Hessian block-row energy at every probe position with randomized Hessian–vector products, and keeps the positions whose within-probe share of curvature grows after fine-tuning. Aggregating this evidence across probes by tokenizer identity yields a ranking of suspicious tokens that is robust to trigger placement. Finally, a score-gap rule decides how many top-ranked tokens to flag as the trigger. Section 3 formalizes the setting and threat model.

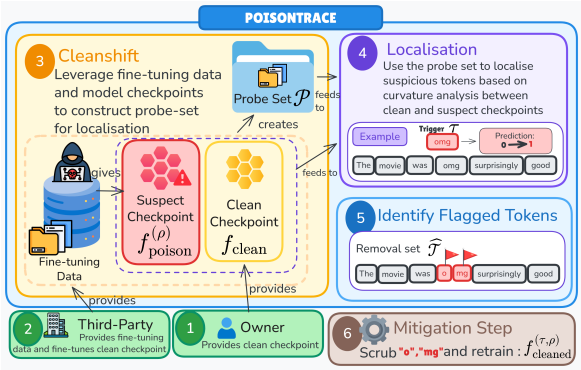

Figure 1: A use case scenario of PoisonTrace. A trusted checkpoint is fine-tuned on third-party data that may be poisoned. PoisonTrace identifies the trigger tokens so that they can be scrubbed from the data before retraining.

Our main contributions are as follows:

1) **An end-to-end framework for trigger localisation.** We present **PoisonTrace**. Given a trusted pre-trained checkpoint, its fine-tuned counterpart and the poisoned fine-tuning dataset, it returns the trigger tokens together with a removal set for sanitising the data. PoisonTrace requires no knowledge of the trigger, its length or position, the poisoning rate, the poisoned samples or the target class.

2) **CleanShift for blind probe acquisition. CleanShift** identifies suspicious poisoned samples using only forward passes of the two checkpoints over the training data. Across all installed attacks, on average 99.6–100% of the selected probes contain the complete trigger, and localisation with CleanShift probes closely matches localisation with oracle probes.

3) **A per-probe curvature-shift statistic for ranking trigger tokens.** We localize poisoning-induced changes through input-Hessian block-row energy at each token position, comparing between the checkpoints before and after fine-tuning.The randomized HVP estimator captures full-spectrum curvature rather than relying on a small set of dominant eigendirections, which our design ablation shows can be dominated by probe-specific curvature.

2 Related Work

Backdoor attacks. *BadNets* established the standard training-time mechanism of stamping a trigger onto selected inputs and assigning them an attacker-chosen target label (Gu et al., 2019). The construction has been adapted to lexical triggers in recurrent text classifiers (Dai et al., 2019) and to poisoned pre-trained transformers (Kurita et al., 2020).

Poison Detection and repair. General defenses detect anomalous representations or predictions through spectral signatures, activation clustering, Neural Cleanse, or STRIP (Tran et al., 2018; Chen et al., 2018; Wang et al., 2019; Gao et al., 2019). Text-specific defenses use different localization signals: The label-flip-rate (LFR) score derived from Kurita et al. (2020) measures target-label flips; ONION uses deletion-induced perplexity change (Qi et al., 2021); BFClass combines ELECTRA (Clark et al., 2020) with label association (Li et al., 2021); AttDef uses ELECTRA filtering with partial-LRP attribution (Li et al., 2023); DeBITE (Yan et al., 2023) exploits target-label association; and BTU (Jiang et al., 2025) uses cross-round embedding displacement.

Curvature-based analysis. Prior work studies parameter-Hessian spectra and Hessian-based influence estimation (Sagun et al., 2017; Ghorbani et al., 2019; Koh & Liang, 2017). Granziol and Flynn analyze poisoning through input-Hessian spectra in linear models (Flynn & Granziol, 2025). PoisonTrace extends this input-curvature perspective to nonlinear language classifiers.

PoisonTrace complements existing text-specific defenses with a geometric localization signal based on poisoning-induced shifts in normalized clean–suspect input curvature, extending input-curvature analysis to nonlinear language classifiers. We evaluate this approach under dirty-label textual backdoor attacks with contiguous triggers inserted at randomized word boundaries.

3 PRELIMINARIES

Attacker. In this work, the adversary follows the BadNets training-time poisoning pattern (Gu et al., 2019): it controls a poison fraction ρ of the training set, appends a fixed trigger string to each controlled sample, and flips its label to a target class (dirty-label attack).

Defender and deployment interface. Our defender has (i) a clean-reference checkpoint f_{clean}; (ii) a suspect checkpoint f_{poison} of the same differentiable architecture, with white-box access to the input embeddings and classification loss required for Hessian-vector products; and (iii) the fine-tuning corpus together with its observed training labels.

Problem formulation. Given $(f_{\text{clean}}, f_{\text{poison}}, \mathcal{D}_P)$, where $\mathcal{D}_P = \{(x_a, y_a)\}_{a=1}^{m}$, the goal is to recover the unknown tokenizer identities associated with the backdoor trigger without oracle knowledge of the trigger pattern τ, its tokenized length n_τ, insertion positions, poisoning rate ρ, poisoned-row identities, or attacker-selected target class $y^\star$. Here, τ denotes the surface-form trigger, which may be a punctuation pattern, a single word, or a multi-word phrase. PoisonTrace first uses CleanShift to infer a trigger-associated class $\widehat{y}$ and acquire a compact suspicious probe set $\mathcal{P} = \{x_i\}_{i=1}^{N}$. It then ranks the tokenizer identities observed across the probes and selects a removal set $\widehat{\mathcal{T}} \subseteq \mathcal{V}_{\text{obs}}$, with the objective that $\widehat{\mathcal{T}} \approx \mathcal{T}^\star$. Here, $\mathcal{V}_{\text{obs}}$ denotes the set of tokenizer identities observed at non-padding probe positions, and $\mathcal{T}^\star$ denotes the unknown set of tokenizer identities produced by tokenizing the true trigger τ. The localization objective is therefore to recover $\mathcal{T}^\star$ while minimizing the inclusion of benign identities under variable trigger placement and unknown trigger length.

4 METHODOLOGY

PoisonTrace first obtains a small suspicious probe set and then performs per-probe curvature localization at the victim-tokenizer level because its curvature signal is defined over tokenized inputs. This is also sufficient for mitigation, as removing constituent trigger tokens can disrupt the original trigger sequence and reduce its ability to activate the backdoor. Figure 2 illustrates the procedure. PoisonTrace **pseudo code** is at Appendix B.1.

4.1 STAGE 1: CLEANSHIFT (PROBE ACQUISITION)

PoisonTrace first applies **Cleanshift** as a lightweight probe-acquisition procedure rather than as a separate localization method. More precisely, Cleanshift takes $(f_{\text{clean}}, f_{\text{poison}}, \mathcal{D}_P)$ as input For each example x_a and class c, CleanShift computes the clean-suspect confidence shift $\Delta_a(c)$ and L_1 normalize it using the robust class-wide location μ_c^{CS} and scale s_c^{CS}, yielding $r_a(c)$. This reduces systematic class-level drift and highlights unusually large residual shifts. For each class, the top B_k residuals, and averaged to obtain $T_{\text{CS}}(c)$. The class with the largest $T_{\text{CS}}(c)$ is selected as $\widehat{y}_{\text{CS}}$, and its top N examples form the probe set $\mathcal{P}$. It therefore returns both a fixed probe set $\mathcal{P}$ and a trigger-associated class estimate $\widehat{y}_{\text{CS}}$. Here, $\widehat{y}_{\text{CS}}$ is the predicted dirty label and $\mathcal{P}$ is the subset of $\mathcal{D}$ where each x_a contains all trigger tokens and its label y_a is $\widehat{y}_{\text{CS}}$. Ideally, $\widehat{y}_{\text{CS}}$ is supposed to $y^\star$ and $\mathcal{P}$ is supposed to be the subset of $\mathcal{D}$ that is backdoored. Cleanshift details are provided in Appendix C.1.

4.2 STAGE2: LOCALISATION

The selected probes are passed unchanged to the PoisonTrace localizer, while $\widehat{y}_{\text{CS}}$ is used to condition its class-specific curvature objective. PoisonTrace then estimates and normalizes a full-spectrum curvature profile independently for each probe and aggregates evidence only after mapping positions to tokenizer identities. Neither Cleanshift nor the localizer receives the true trigger τ, its token length n_τ, its position, the poisoning rate ρ, the poisoned-row identities, or the attacker target $y^\star$ as algorithmic input.

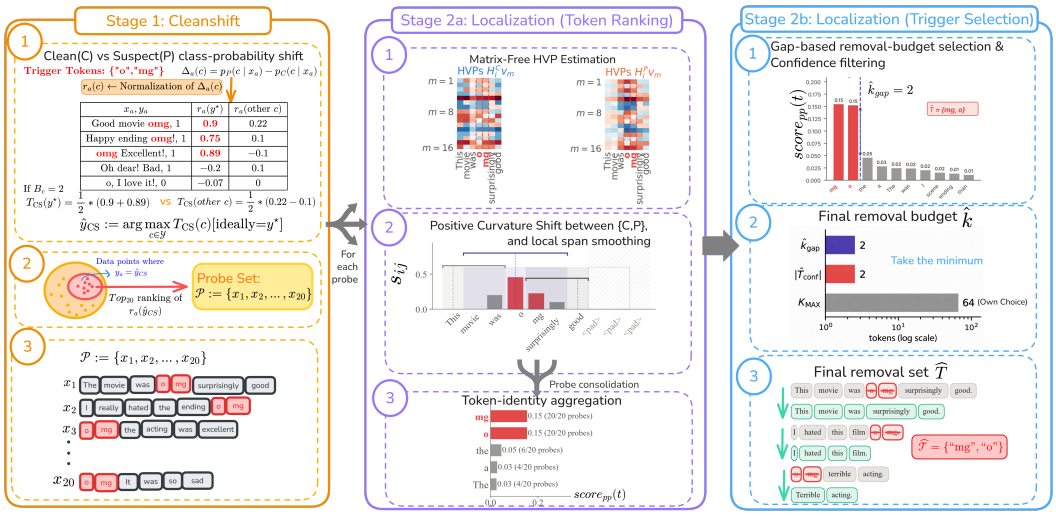

Figure 2: Overview of PoisonTrace, illustrated with the trigger "omg" (GPT-2 tokens "o" and "mg"). **Stage 1:** CleanShift compares the class probabilities of the checkpoints before (C) and after (P) fine-tuning, infers the trigger-associated class and selects N=20 probes. **Stage 2a:** for each probe, PoisonTrace estimates positional input curvature with randomized HVPs, keeps the positive within-probe curvature-share shift, smooths it locally and aggregates it by tokenizer identity. **Stage 2b:** a score-gap rule and a robust confidence filter set the budget(capped by $k_{\max}$) and the top suspicious tokens form the trigger set, which is scrubbed from the training data before retraining.

4.2.1 Per-Probe Setting and Notation

The symbols C and P index a trusted clean-reference checkpoint f_C and a post-fine-tuning suspect checkpoint f_P. The role label P does not assume that poisoning is known a priori: operationally, f_P is the checkpoint under investigation and may ultimately be benign or backdoored. We assume a fixed sequence length L and embedding dimension d_e, with the attention mask held fixed and suppressed in the function notation. Let the automatically acquired probe set $\mathcal{P}$ contain N suspicious textual inputs.Unless otherwise stated, all fixed hyperparameters and numerical stabilizers, including the ε-terms, are reported in Appendix E. For checkpoint M and Probe i, the probe tokens are mapped through that checkpoint's learned input-embedding table to obtain the embedding sequence E_i^M, token identities t_{ij}, and a non-padding mask m_{ij}. For token position j, the token identity satisfies $t_{ij} \in \mathcal{V}$. The domains and dimensions of all key quantities are summarized in Appendix A.

Localization-target conditioning. The probe-acquisition rule returns both the selected probe set and the Cleanshift trigger-associated class estimate $\widehat{y}_{\mathrm{CS}}$. PoisonTrace uses $\widehat{y}_{\mathrm{CS}}$ directly to condition the curvature objective, so the localizer does not require the true attacker target $y^\star$. The selected probes and inferred class remain fixed throughout localization. Let the checkpoint-specific classification loss be ℓ_M, evaluated through the victim architecture's native classification readout, with its padding mask and architecture-specific pooling rule applied consistently to both checkpoints. Define the per-probe input-embedding Hessian $H_i^M = \nabla_E^2 \ell_M(E, \widehat{y}_{\mathrm{CS}})\big|_{E=E_i^M}$ is a symmetric matrix.

4.2.2 Full-Spectrum Positional Curvature Functional

PoisonTrace assigns each token position j a curvature score $R_j(H) = \sum_{k=1}^{L} \|H[j,k]\|_F^2$ that measures the total Hessian block-row energy associated with that position. Although this quantity is defined using the full input Hessian H, it also admits the matrix-free characterization $R_j(H) = \mathbb{E}_v \|(Hv)_j\|_2^2$ for a Rademacher vector v satisfying $\mathbb{E}[vv^\top] = I$. Following Hutchinson-style randomized estimation (Hutchinson, 1989), we estimate this expectation using random Rademacher directions, while each Hessian–vector product Hv is computed directly without explicitly constructing H (Pearlmutter, 1994). PoisonTrace estimates this quantity using M_{H} independent Rademacher probe vectors, giving the *Hutchinson estimator of the positional curvature* : $\widehat{R}_{ij}^M = \frac{1}{M_{\mathrm{H}}} \sum_{r=1}^{M_{\mathrm{H}}} \|(H_i^M v_i^{(r)})_j\|_2^2$, where M_{H} denotes the number of randomized Hessian-vector product (HVP) draws and is specified in Appendix E. The full derivation, unbiasedness proof, and

implementation details are given in Appendix D. After masking padding coordinates, this stage produces the paired raw profiles $\widehat{\mathbf{R}}_i^M := (\widehat{R}_{i1}^M, \ldots, \widehat{R}_{iL}^M)$ for $M \in \{C, P\}$.

4.2.3 Scale-Free Per-Probe Curvature Shift and Span Smoothing

The preceding estimator supplies paired raw profiles $\widehat{\mathbf{R}}_i^C$ and $\widehat{\mathbf{R}}_i^P$, but their absolute magnitudes are not directly comparable. A checkpoint-wide scale change can make curvature larger at nearly every position without identifying a trigger-specific location.PoisonTrace therefore normalizes each profile within the same probe and retains only the *positive curvature-share shift*: $s_{ij} = ReLU\left(\widehat{e}_{ij}^P - \widehat{e}_{ij}^C\right)$ such that $\widehat{e}_{ij}^M := \frac{\widehat{R}_{ij}^M m_{ij}}{\sum_{k=1}^L \widehat{R}_{ik}^M m_{ik} + \varepsilon_R}$ represents the fraction of the probe's total curvature energy assigned to position j. The rectifier is not a learned threshold. It records positions whose share of total curvature increased under poisoning; positions whose share decreased do not provide evidence of an added trigger mechanism. Hence s_{ij} is the first suspect-specific positional signal: it measures where curvature is reallocated, rather than where curvature is merely large.

Span smoothing. The shift s_{ij} is still attached to one position. Because a lexical trigger may split into adjacent sub-tokens and its curvature may peak on only some pieces, PoisonTrace pools nearby evidence before aggregation. Here w is the fixed smoothing radius and δ is the integer offset. w controls how curvature evidence is smoothed across nearby token positions within each probe: $\widetilde{e}_{ij} = \sum_{\substack{|\delta| \le w \\ 1 \le j+\delta \le L}} s_{i,j+\delta}$, and renormalize it: $e_{ij} = \frac{\widetilde{e}_{ij} m_{ij}}{\sum_{k=1}^L \widetilde{e}_{ik} m_{ik} + \varepsilon_R}$. The *renormalized smoothed energy* represent the relative share of smoothed evidence assigned to position j rather than a value inflated because the same local evidence is counted in multiple overlapping smoothing windows. The radius w is fixed before n_τ is observed and therefore does not leak the true trigger length. This smoothing creates a local position profile; the final detector still ranks tokenizer identities, not oracle trigger spans. Sensitivity analysis of w is found in Appendix H.2.

4.2.4 Position-Robust Token-Identity Aggregation

After smoothing, e_{ij} remains indexed by absolute position. Because randomized insertion moves the same trigger identity to different positions across probes, those positions cannot be aligned directly. PoisonTrace therefore pools evidence only after mapping each position j to its tokenizer identity t_{ij}. Define the unrestricted observed candidate set $\mathcal{V}_{\text{obs}} = \{t : \exists i, j \text{ with } m_{ij} = 1 \text{ and } t_{ij} = t\}$. Thus $\mathcal{V}_{\text{obs}} \subseteq \mathcal{V}$. No token-type filtering is applied. Let $\mathbf{1}\{\cdot\} \in \{0, 1\}$ be the indicator of its predicate. For $t \in \mathcal{V}_{\text{obs}}$, define occurrence $\text{occ}(t) = \sum_{i=1}^N \sum_{j=1}^L m_{ij} \mathbf{1}\{t_{ij} = t\}$ and probe-frequency support $\text{pf}(t) = \frac{1}{N} |\{i : \exists j,\ m_{ij} = 1,\ t_{ij} = t\}|$, then weight the *mean positional energy* by that support:

$$\text{score}_{\text{PP}}(t) = \frac{\sum_{i=1}^N \sum_{j=1}^L e_{ij} m_{ij} \mathbf{1}\{t_{ij} = t\}}{\max\{\text{occ}(t), 1\}} \text{pf}(t). \tag{1}$$

Here, $\text{occ}(t)$ normalizes by the total number of occurrences of token identity t to reduce raw frequency bias, while $\text{pf}(t)$ rewards identities that recur consistently across probes. The denominator $\max\{\text{occ}(t), 1\}$ safely handles the zero-occurrence case without affecting observed identities. Because evidence is aggregated by tokenizer identity rather than absolute position, trigger positions may vary arbitrarily across probes.

4.2.5 Data-Driven Trigger Selection

The identity score produces a ranking, but not a removal set. Because the true distinct trigger token set size n_T is unknown, PoisonTrace must infer how many high-scoring identities form a separated candidate token. PoisonTrace estimates $\widehat{k}_{\text{gap}}$ using a gap rule that ranks the positive token-identity scores $\text{score}_{\text{PP}}(t)$ and identifies the largest relative drop between consecutive scores, thereby estimating the size of the most clearly separated high-score block. Independently, $\widehat{\mathcal{T}}_{\text{conf}}$ is formed by robustly standardizing token scores against the background distribution and retaining only identities whose standardized scores exceed a fixed confidence threshold. Let $K_{\max}$ denote a fixed hyperparameter that caps the maximum number of token identities allowed for removal. The *final*

removal budget $\widehat{K}$ and *removal set* $\widehat{\mathcal{T}}$ are then

$$\widehat{K} := \begin{cases} 0, & \widehat{\mathcal{T}}_{\text{conf}} = \varnothing, \\ \min\left(\widehat{k}_{\text{gap}}, |\widehat{\mathcal{T}}_{\text{conf}}|, K_{\max}\right), & \text{otherwise}, \end{cases} \qquad \widehat{\mathcal{T}} := \text{Top}_{\widehat{K},\, t \in \mathcal{V}_{\text{obs}}} \text{score}_{\text{PP}}(t). \tag{2}$$

If no token identity satisfies $z_{\text{rob}}(t) \geq z^\star$, then $\widehat{\mathcal{T}}_{\text{conf}} = \varnothing$, so PoisonTrace sets $\widehat{K} = 0$ and abstains.

5 Experimental Setup

5.1 Datasets and Base Model

Datasets. We evaluate four English text-classification corpora: IMDB (Maas et al., 2011), Cornell Movie Review Polarity v2.0 (Pang & Lee, 2004), the Books, DVD, Music, and Video subset of FDU-MTL (Liu et al., 2017), and WildGuardMix (Han et al., 2024). IMDB, Cornell, and FDU-MTL use 40,000/5,000/5,000, 1,600/200/200, and 5,600/800/1,600 train/validation/test splits, respectively. For WGM prompt-harmfulness classification, duplicate prompts are collapsed, the empty prompt is removed, and the remaining 47,851 prompts are stratified with seed 42 into a 37,851/5,000/5,000 split. IMDB, Cornell, FDU-MTL and WGM near-balanced. In WGM, poisoning targets harmful prompts, so a successful attack corresponds to a guard bypass rather than a sentiment flip. For the *Main Evaluation*, we use 26 trigger strings derived from NIST TrojAI Round 5 (National Institute of Standards and Technology, 2021), our setup does not reproduce the full NIST benchmark. These triggers in Tables 9,10,11 uses shared randomized-placement victim checkpoints within each corpus. The set provides diverse, externally defined triggers of varying forms and lengths, making it a suitable test of PoisonTrace's trigger-localization robustness.

Experimental setup and compared methods. Each corpus is trained and evaluated separately. Poisoning is applied only to the training split, while validation and test examples remain clean, and poison fractions are measured relative to the negative training class. In the reported experiments, the victim model is `GPT2ForSequenceClassification` with two labels; this GPT-2 configuration is an experimental instantiation, and unfused attention is used to support the reported second-order input-Hessian computations. Tokenizer conventions, fixed-length encoding, and the common $L = 128$ input representation are specified in Appendix E.2. We evaluate six baseline defenses: LFR, BFClass, ONION, AttDef, DeBITE, and BTU against PoisonTrace(ours).

5.2 Trigger Construction

A trigger is a fixed lexical string τ inserted as one contiguous block at an independently sampled valid word boundary. Since GPT-2 uses byte-pair encoding, a surface word can split into several ordered sub-token identities. We define the ground-truth trigger sub-token sequence as $Q(\tau) = \text{encode}(_\tau) = (t_1, t_2, \ldots, t_{n_\tau})$, where $n_\tau = |Q(\tau)|$. PoisonTrace aims to recover the distinct sub-token identities contained in $Q(\tau)$. In a dirty-label attack, selected negative training reviews receive one contiguous trigger and are class labels are relabeled positive. The operating poison rates are $\rho_{\text{IMDB}} = 0.01$, $\rho_{\text{Cornell}} = 0.10$, $\rho_{\text{FDU-MTL}} = 0.10$, and $\rho_{\text{WildGuardMix}} = 0.05$ (Shi et al., 2025). These rates are selected to provide comparable poisoning regimes across datasets while accounting for dataset-specific differences in attack installation.

5.3 Experimental design.

Main Evaluation evaluates the NIST triggers under randomized placement; probes are acquired independently within each trigger condition using the same fixed rule. Within each corpus-trigger condition, the registered suspect victim checkpoint is shared across the comparison methods, so differences in reported performance reflect the **localization** and selection mechanisms rather than independently trained victim models. **Mitigation** tests whether flagged triggers are causally involved in the backdoor. The corpus is scrubbed only at identities in $\widehat{\mathcal{T}}$, and a fresh classifier of the same architecture is retrained from the common base checkpoint. Main Evaluation conditions on an installed backdoor, then measures recovery of the corresponding trigger identities and the effectiveness of the resulting mitigation.

5.4 Model Training and Checkpointing

We use three roles: f_{clean} is trained once on the unmodified corpus; $f^{(\rho)}_{\text{poison}}$ is trained for each poisoned configuration; and $f^{(\tau,\rho)}_{\text{cleaned}}$ is retrained after scrubbing only identities in the automatically selected set $\widehat{\mathcal{T}}$. All roles use the same base checkpoint and optimization configuration, reported in Appendix E. Main Evaluation then shares each registered corpus-trigger suspect checkpoint across all comparison methods, so the reported differences arise from the localization scores rather than from independently trained victims. Sanitization follows Eq. equation 2 exactly: no fixed K, poison-rate quota, or lower-ranked padding is allowed. The cleaned corpus is reconstructed from the same poisoned corpus by removing only identities in $\widehat{\mathcal{T}}$, after which retraining starts from the common base checkpoint. For controlled comparison, all baseline outputs are represented in the common GPT-2 token space while preserving each method's native trigger-selection rule where defined. The resulting removal sets are then evaluated under the same scrub/retrain procedure; methods without a native removal-set criterion are evaluated only for localization.The corresponding processing pipelines are detailed in Appendix F.

5.5 Evaluation Metrics

An attack is considered *installed* only when the poisoned-model ASR is at least 50% and exceeds the clean-model baseline by at least 20 percentage points. A tie would render more steps referenced in Appendix E.4. Probe quality is evaluated only post hoc, after the selected probe identities have been fixed. We report the fraction of selected probes drawn from poisoned rows and the fraction that actually contain the complete trigger in the model input; their formal definitions are given in Appendix C.1. These sample-level diagnostics are kept separate from the token-level localization metrics below. Table 1 defines localization, sample-level metrics are separated in Appendix E.5.

Table 1: **Trigger-localization metrics used in Main Evaluation.**

Metric	Definition	Interpretation
Precision@n_T	$\lvert R_{n_T} \cap \mathcal{T}^\star \rvert / n_T$	fraction of top-n_T identities that are true triggers
Exact@n_T	$\mathbf{1}[R_{n_T} = \mathcal{T}^\star]$	perfect top-n_T trigger-set recovery
Removal-set precision/recall	$\lvert \widehat{\mathcal{T}} \cap \mathcal{T}^\star \rvert / \lvert \widehat{\mathcal{T}} \rvert$, $\lvert \widehat{\mathcal{T}} \cap \mathcal{T}^\star \rvert / \lvert \mathcal{T}^\star \rvert$	purity / completeness of the selected removal set
Exact removal set	$\mathbf{1}[\widehat{\mathcal{T}} = \mathcal{T}^\star]$	selected removal set exactly equals the true trigger set.

Here $n_T = |\mathcal{T}^\star|$ (the number of distinct ground-truth trigger token identities) and R_{n_T} is the set of the top-n_T ranked token identities. $\widehat{\mathcal{T}}$ and $\mathcal{T}^\star$ are unique tokens

Ranking metrics. At oracle depth n_T, Precision@n_T measures the fraction of true trigger identities recovered within the top-n_T positions, while Exact@n_T records whether the complete trigger set is recovered exactly. Since both sets have size n_T, Exact@$n_T = 1 \iff$ Precision@$n_T = 1$. Both are used only as oracle diagnostics of token-ranking quality because the true trigger length n_τ is known only in controlled experiments.

Removal-set metrics. PoisonTrace infers the removal budget $\widehat{K}$ and constructs the selected removal set $\widehat{\mathcal{T}}$. Removal precision, recall, and F_1 compare the selected set $\widehat{\mathcal{T}}$ with the true trigger set $\mathcal{T}^*$. Precision measures how much of the scrubbed set is correct, recall measures how much of the trigger is removed, and F_1 balances both. ExactRemoval is $\mathbf{1}[\widehat{\mathcal{T}} = \mathcal{T}^*]$, with $F_1 = 1 \iff$ ExactRemoval$= 1$. These metrics depend on each method's selected budget and should be read together with mean $|\widehat{\mathcal{T}}|$.

Mitigation metrics. Define $\text{ASR}(f, \tau, p) = \Pr(f(T_{\tau,p}(x)) = \text{positive} \mid y = \text{negative})$ where p follows the registered valid-word-boundary distribution; Mitigation uses ASR drop $= \text{ASR}_{\text{poison}} - \text{ASR}_{\text{cleaned}}$, Residual ASR $= \text{ASR}_{\text{cleaned}} - \text{ASR}_{\text{clean}}$, and $\Delta\text{Acc} = \text{Acc}_{\text{cleaned}} - \text{Acc}_{\text{clean}}$. Residual ASR of zero returns to the clean-model triggered-prompt baseline; they measure the end-to-end effectiveness and utility of the resulting mitigation.

6 Results and Analysis

Aggregate localization. Across the 26 triggers, Table 6 shows that PoisonTrace is the strongest localizer on all four corpora because Exact@n_T is the highest, despite only a marginal ΔAcc. Although PoisonTrace did not rank the highest for Removal-set Recall, but notice that BTU's high recall is met with very low precision. Hence one should consider F1-score. PoisonTrace outforms all for F1-score.

Table 2: **Aggregate Main Evaluation results** at $L = 128$ under the randomized-token-placement setting.

		Setup		Trigger localization							Mitigation		
Dataset (ρ)	Method	Clean Acc(%)	Installed	Mean Precision@n_T	Exact@n_T	Removal-set Precision	Removal-set Recall	Removal-set F1 score	Exact removal set	Mean $\lvert\widehat{\mathcal{T}}\rvert$	ASR drop	Residual ASR	ΔAcc(%)
IMDB ($\rho = 1\%$)	LFR	89.34	22/26	0.391	5/22	-	-	-	-	-	-	-	-
	BFClass		22/26	0.145	2/22	0.000	0.000	0.000	0/22	0.00	+0.00	+85.39	**+0.18**
	ONION		22/26	0.675	11/22	-	-	-	-	-	-	-	-
	AttDef		22/26	0.063	0/22	0.011	0.317	0.021	0/22	353.14	+12.41	+72.98	-3.93
	DeBITE-z		22/26	0.007	0/22	0.001	0.931	0.002	0/22	5291.36	+59.28	+26.11	-11.67
	BTU		22/26	0.448	3/22	0.001	**0.969**	0.002	0/22	6146.45	+52.56	+32.83	-13.86
	PoisonTrace (ours)		22/26	**0.979**	**18/22**	**0.986**	0.911	**0.939**	**14/22**	5.73	**+79.84**	**+5.55**	-0.01
Cornell ($\rho = 10\%$)	LFR	73.50	26/26	0.455	5/26	-	-	-	-	-	-	-	-
	BFClass		26/26	0.280	4/26	0.000	0.000	0.000	0/26	0.00	+0.00	+75.62	**-2.69**
	ONION		26/26	0.626	13/26	-	-	-	-	-	-	-	-
	AttDef		26/26	0.120	0/26	0.028	0.714	0.048	0/26	245.04	+27.42	+48.19	-14.37
	DeBITE-z		26/26	0.213	0/26	0.012	0.777	0.024	0/26	397.50	+36.46	+39.15	-9.94
	BTU		26/26	0.557	3/26	0.001	**1.000**	0.002	0/26	6254.00	**+82.92**	**-7.31**	-18.83
	PoisonTrace (ours)		26/26	**0.980**	**24/26**	**1.000**	0.867	**0.900**	**19/26**	4.23	+49.58	+26.04	-3.94
FDU-MTL ($\rho = 10\%$)	LFR	89.25	26/26	0.465	9/26	-	-	-	-	-	-	-	-
	BFClass		26/26	0.118	3/26	0.019	0.038	0.026	0/26	0.08	+3.23	+83.61	**-0.34**
	ONION		26/26	0.656	14/26	-	-	-	-	-	-	-	-
	AttDef		26/26	0.160	0/26	0.027	0.799	0.051	0/26	175.38	+37.78	+49.06	-7.89
	DeBITE-z		26/26	0.690	5/26	0.008	0.848	0.016	0/26	621.19	+66.64	+20.20	-7.46
	BTU		26/26	0.562	6/26	0.001	**1.000**	0.002	0/26	6228.04	+26.88	+59.96	-28.08
	PoisonTrace (ours)		26/26	**0.990**	**24/26**	**1.000**	0.926	**0.941**	**22/26**	4.96	**+75.80**	**+11.05**	-0.72
WildGuardMix ($\rho = 5\%$)	LFR	91.40	26/26	0.379	5/26	-	-	-	-	-	-	-	-
	BFClass		26/26	0.227	3/26	0.114	0.205	0.115	0/26	3.77	+13.31	+76.44	**-0.46**
	ONION		26/26	0.674	14/26	-	-	-	-	-	-	-	-
	AttDef		26/26	0.134	0/26	0.063	0.622	0.107	0/26	71.19	+31.63	+58.11	-2.45
	DeBITE-z		26/26	0.386	0/26	0.002	0.967	0.004	0/26	2904.35	+57.01	+32.73	-14.86
	BTU		26/26	0.576	5/26	0.001	**0.998**	0.002	0/26	6358.85	+56.60	+33.14	-15.80
	PoisonTrace (ours)		26/26	**0.900**	**21/26**	**0.818**	0.810	**0.783**	**13/26**	5.73	**+68.76**	**+20.65**	-0.88

† BTU's Cornell mitigation figures reflect model destruction rather than trigger removal: its native top-α union strips most of the lexical signal from this, the smallest corpus, leaving clean accuracy near the chance floor. The trigger-attributable effect is essentially zero because a near-chance classifier cannot express a backdoor.
Localization and mitigation metrics are reported over attacks that successfully installed; Exact@n_T, Exact removal set and mean $\lvert\widehat{\mathcal{T}}\rvert$ use the same installed-attack denominator. Green headers indicate higher is better; red headers indicate lower residual ASR or closer to zero is better. Black headers report selected-set size. Sensitivity analysis for longer context lengths can be found in Appendix H.1.

Table 3: **Ablation of oracle versus CleanShift probe acquisition.** at $L = 128$, randomized token placement.

		Setup		Trigger localization							Mitigation		
Dataset (ρ)	Method	Clean Acc(%)	Installed	Mean Precision@n_T	Exact@n_T	Removal-set Precision	Removal-set Recall	Removal-set F1 score	Exact removal set	Mean $\lvert\widehat{\mathcal{T}}\rvert$	ASR drop	Residual ASR	ΔAcc(%)
IMDB ($\rho = 1\%$)	PoisonTrace (Cleanshift)	89.34	22/26	0.979	**18/22**	0.986	**0.911**	**0.939**	**14/22**	5.73	+79.84	+5.55	-0.01
	PoisonTrace (oracle probes)		22/26	**0.985**	**18/22**	**1.000**	0.860	0.908	13/22	5.77	**+82.30**	**+3.09**	**+0.02**
Cornell ($\rho = 10\%$)	PoisonTrace (Cleanshift)	73.50	26/26	0.980	**24/26**	**1.000**	**0.867**	**0.900**	**19/26**	4.23	**+49.58**	**+26.04**	**-3.94**
	PoisonTrace (oracle probes)		26/26	**0.988**	**24/26**	**1.000**	0.832	0.874	17/26	4.23	+47.58	+28.04	-4.52
FDU-MTL ($\rho = 10\%$)	PoisonTrace (Cleanshift)	89.25	26/26	**0.990**	**24/26**	**1.000**	**0.926**	**0.941**	**22/26**	4.96	**+75.80**	**+11.05**	-0.72
	PoisonTrace (oracle probes)		26/26	0.989	**24/26**	**1.000**	0.859	0.894	19/26	4.65	+73.06	+13.78	**-0.48**
WildGuardMix ($\rho = 5\%$)	PoisonTrace (Cleanshift)	91.40	26/26	0.900	21/26	0.818	0.810	0.783	13/26	5.73	+68.76‡	+20.65‡	-0.88‡
	PoisonTrace (oracle probes)		26/26	**0.970**	**23/26**	**0.948**	**0.866**	**0.864**	**18/26**	5.31	**0.970**	**23/26**	**0.948**

‡ PoisonTrace abstains ($\widehat{K} = 0$) on 3 of 26 installed cells on WGM. An abstained cell removes nothing and so has no cleaned model to score; it is excluded from the three mitigation columns, which are therefore averaged over 23 cells on WGM. This matches the grids' own aggregates. The localisation columns are unaffected.

Mitigation and utility. The mitigation columns evaluate the model after the selected identities are scrubbed and the classifier is retrained. PoisonTrace produces the largest ASR drop on each corpus, while its clean-accuracy change remains small relative to the reduction in backdoor success. While a precise removal set identifies the trigger mechanism, but the downstream scrub/retrain procedure must still erase the association learned from the poisoned examples.

7 Ablation and Robustness Analysis

Oracle-probe ablation. Table 3 shows that replacing CleanShift with oracle trigger-bearing probes yields only marginal localization changes on IMDB, Cornell, and FDU-MTL: Exact@n_T is unchanged, and CleanShift often achieves equal or better removal-set recovery. This indicates that CleanShift already approaches the oracle-probe upper bound on these corpora and that trigger presence alone does not guarantee more useful curvature evidence. WGM is the main exception, where oracle probes improve both ranking and removal-set quality, suggesting that probe selection remains a bottleneck in more heterogeneous settings.

Probe Purity of Cleanshift. Oracle probe acquisition assumes that all probes passed to the localizer are trigger-bearing. In Table 4, CleanShift achieves consistently high probe purity (99.6–100%), suggesting that the observed differences in localization performance are unlikely to be driven by



Figure 3: **PoisonTrace sensitivity and trigger-length robustness.** Left: Exact@n_T across poisoning rates on WGM. Right: Exact@n_T across distinct trigger token lengths on IMDB, Cornell, FDU-MTL, and WGM. Numbers in parentheses indicate the number of triggers for each trigger length.

probe-acquisition quality. An analysis establishing the threshold for adequate probe purity is provided in Appendix C.2.

Exact@n_T across different ρ. Figure 3 (top) shows the performance of PoisonTrace for exact@n_T remained robust against different ρ with its Exact@n_T performing above 81%, while all others below 54%. Full results table is at Appendix 21.

Exact@n_T across different length of distinct triggers. Figure 3 (bottom) shows that PoisonTrace maintains strong exact localization over a substantially broader range of trigger lengths than the baselines. While ONION remains competitive for short triggers (below 5), its performance degrades as trigger length increases, whereas PoisonTrace continues to recover medium-length triggers that the other methods fail to localize exactly.

Table 4: **Probe purity** of Cleanshift versus oracle probe acquisition.

	IMDB (1%)	Cornell (10%)	FDU-MTL (10%)	WildGuardMix (5%)
Oracle TBR@20	1.000	1.000	1.000	1.000
Cleanshift TBR@20	1.000	0.996	1.000	1.000
Δ vs. oracle	± 0.000	-0.004	± 0.000	± 0.000
Installed cells	22/26	26/26	26/26	26/26

TBR@20 is the fraction of the 20 selected probes that contain the complete trigger, averaged over attack-installed cells. Oracle TBR@20 is 1.000 by construction..

8 Discussion

Position-robust and selective localization. PoisonTrace gives the strongest localization row on most corpus: The selected sets remain close to the true trigger size rather than removing large vocabularies. Together with the design ablation, these results show that recurring trigger identities can be recovered without a shared absolute coordinate. PoisonTrace showed advantage in handling more complex, multi-token triggers, although very long triggers remain challenging for all methods.

Localization and mitigation measure different stages. Localization evaluates whether PoisonTrace correctly identifies the trigger-related token identities, whereas mitigation evaluates whether removing those identities from the training corpus and retraining actually suppresses the backdoor. A strong localization result therefore does not necessarily imply equally strong mitigation: even when the trigger identities are identified accurately, the retrained model may retain part of the learned backdoor association. The residual ASR captures this remaining attack effect after mitigation.

Adaptive attackers. PoisonTrace does not rely solely on the leading Hessian mode: its randomized HVP estimator captures full-spectrum positional curvature, while localization additionally requires recurrence across probes. This suggests that a defense-aware attacker would need to suppress or redistribute this broader signal across multiple probes rather than targeting only the dominant curvature mode.

9 Conclusion, Limitations and Future Work

We introduced **PoisonTrace**, which localizes backdoor triggers by comparing how input curvature is distributed across positions in a trusted and a suspect checkpoint. Using CleanShift probes and token-identity aggregation, it returns position-robust trigger rankings and a data-driven removal set without prior knowledge of the trigger or the poisoning configuration. Our experiments suggest that clean–suspect differences in input curvature are an effective signal for trigger localization. PoisonTrace removes trigger token identities rather than recovering the full trigger sequence, so benign occurrences of those identities may also be scrubbed, potentially causing a small clean-accuracy cost. Future work should recover the trigger sequence or structure to enable more selective sanitization.

AI USE STATEMENT

All research ideas, hypotheses, methodological directions, and conceptual decisions were developed by us. First, we used AI to help formulate mathematical claims based on our own ideas and research directions, and to assist in translating these claims into formal mathematical arguments and proofs. Second, we used AI to improve the clarity, grammar, and readability of the manuscript. Third, we used AI to assist with software development and implementation, including writing and refining portions of the experimental code.

We reviewed and manually verified all AI-assisted mathematical content, writing, and code. Where appropriate, we also cross-checked the outputs using multiple AI tools. We take full responsibility for the correctness of the mathematical arguments, implementation, experimental results, and final manuscript.

REFERENCES

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018.

Kevin Clark, Minh-Thang Luong, Quoc Le, and Christopher D Manning. Pre-training transformers as energy-based cloze models. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 285–294, 2020.

Jiazhu Dai, Chuanshuai Chen, and Yufeng Li. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7:138872–138878, 2019.

Donald Flynn and Diego Granziol. A linear approach to data poisoning. *arXiv preprint arXiv:2505.15175*, 2025.

Yansong Gao, Change 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.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning*, pp. 2232–2241. PMLR, 2019.

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

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.

Peihai Jiang, Xixiang Lyu, Yige Li, and Jing Ma. Backdoor token unlearning: Exposing and defending backdoors in pretrained language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24285–24293, 2025.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. Pmlr, 2017.

Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 2793–2806, 2020.

Jiazhao Li, Zhuofeng Wu, Wei Ping, Chaowei Xiao, and VG Vinod Vydiswaran. Defending against insertion-based textual backdoor attacks via attribution. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8818–8833, 2023.

Zichao Li, Dheeraj Mekala, Chengyu Dong, and Jingbo Shang. Bfclass: A backdoor-free text classification framework. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 444–453, 2021.

Pengfei Liu, Xipeng Qiu, and Xuan-Jing Huang. Adversarial multi-task learning for text classification. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 1–10, 2017.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

National Institute of Standards and Technology. TrojAI Round 5: NLP Sentiment Classification, March 2021. https://pages.nist.gov/trojai/docs/nlp-sentiment-classification-mar2021.html, 2021. TrojAI documentation.

Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting of the association for computational linguistics (ACL-04)*, pp. 271–278, 2004.

Barak A Pearlmutter. Fast exact multiplication by the hessian. *Neural computation*, 6(1):147–160, 1994.

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, 2021.

Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

Wei Shi, Ziyuan Xie, Sihang Li, and Xiang Wang. Safer: Probing safety in reward models with sparse autoencoder. *arXiv preprint arXiv:2507.00665*, 2025.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *Advances in neural information processing systems*, 31, 2018.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE symposium on security and privacy (SP)*, pp. 707–723. IEEE, 2019.

Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A Gunter, and Bo Li. Detecting ai trojans using meta neural analysis. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 103–120. IEEE, 2021.

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.

A NOTATION AND DOMAINS

Table 5 summarizes the notation, meanings, and domains of the principal quantities used throughout PoisonTrace. Foundational spaces, dimensions, and indexing conventions are listed first so that the domains of subsequent quantities can be interpreted directly. Unless otherwise stated, all fixed hyperparameters and numerical stabilizers are specified in Appendix E.

Table 5: **Notation and domains used throughout PoisonTrace.**

Notation	Description	Dimension / Space
Foundational spaces, dimensions, and indexing		
$\mathbb{N}_{>0}$	Positive integers.	$\{1, 2, \ldots\}$
$\mathbb{N}_0$	Non-negative integers.	$\{0, 1, 2, \ldots\}$
$\mathbb{Z}$	Integers.	$\{\ldots, -1, 0, 1, \ldots\}$
$\mathbb{R}$	Real numbers.	Real scalar space
$\mathbb{R}_{\geq 0}$	Non-negative real numbers.	$[0, \infty)$
$\mathbb{R}_{>0}$	Positive real numbers.	$(0, \infty)$
$[q]$	Index-set shorthand for a positive integer q, where $[q] := \{1, \ldots, q\}$.	$\{1, \ldots, q\}$
Σ	Set of supported input characters used to represent surface-form text.	Finite set
Σ^*	Set of all finite strings over Σ.	Discrete text space
$\mathcal{Y}$	Set of classification labels.	Finite set
$\mathcal{V}$	Victim-model tokenizer vocabulary.	Finite set
V	Vocabulary size, $V := \lvert\mathcal{V}\rvert$.	$\mathbb{N}_{>0}$
$\mathcal{V}^*$	Set of all finite token-identity sequences over $\mathcal{V}$.	Discrete sequence space
L	Fixed encoded sequence length.	$\mathbb{N}_{>0}$
d_e	Input-embedding dimension.	$\mathbb{N}_{>0}$
N	Number of suspicious probes selected by CleanShift.	$\mathbb{N}_{>0}$
$\mathcal{M}$	Checkpoint-role set, $\mathcal{M} := \{C, P\}$.	Finite set
C, P	Clean-reference and suspect checkpoint roles, respectively.	$\mathcal{M}$
$\mathbb{S}^{Ld_e}$	Space of real symmetric $Ld_e \times Ld_e$ matrices.	$\{A \in \mathbb{R}^{Ld_e \times Ld_e} : A = A^\top\}$
Threat model and problem formulation		
m	Number of examples in the suspect training corpus.	$\mathbb{N}_{>0}$
x_a	Textual input of training example a.	Σ^*
y_a	Observed label of training example a.	$\mathcal{Y}$
a	Training-example index.	$\{1, \ldots, m\}$
$\mathcal{D}_P$	Suspect training corpus, $\mathcal{D}_P = \{(x_a, y_a)\}_{a=1}^m$.	$\mathcal{D}_P \subseteq \Sigma^* \times \mathcal{Y}$
f_{clean}	Trusted clean-reference checkpoint.	$\mathbb{R}^{L \times d_e} \to \mathbb{R}^{\lvert\mathcal{Y}\rvert}$
f_{poison}	Suspect post-fine-tuning checkpoint.	$\mathbb{R}^{L \times d_e} \to \mathbb{R}^{\lvert\mathcal{Y}\rvert}$
ρ	Training-set poisoning fraction.	$[0, 1]$
τ	Unknown surface-form trigger; may contain punctuation, a word, a phrase, or other supported characters.	Σ^*
Q	Victim-tokenizer encoding map used for the trigger.	$Q : \Sigma^* \to \mathcal{V}^*$
$Q(\tau)$	Ordered token-identity sequence obtained by tokenizing the trigger.	$\mathcal{V}^{n_\tau}$
n_τ	Length of the tokenized trigger sequence, $n_\tau := \lvert Q(\tau)\rvert$; counts token occurrences.	$\{1, \ldots, L\}$
n_T	Number of distinct tokenizer identities in the trigger, $n_T := \lvert\mathcal{T}^\star\rvert$.	$\{1, \ldots, n_\tau\}$

Continued on next page

Table 5 – continued

Notation	Description	Dimension / Space
$\mathcal{T}^\star$	Set of distinct tokenizer identities induced by the true trigger τ.	$\mathcal{T}^\star \subseteq \mathcal{V}$
$y^\star$	Unknown attacker-selected target class.	$\mathcal{Y}$
$\widehat{y}_{\mathrm{CS}}$	Trigger-associated class inferred by CleanShift.	$\mathcal{Y}$
$\mathcal{P}$	Suspicious probe set selected by CleanShift, $\mathcal{P} = \{x_i\}_{i=1}^N$.	$\mathcal{P} \subseteq \Sigma^*, \quad \lvert\mathcal{P}\rvert = N$
$\mathcal{V}_{\mathrm{obs}}$	Tokenizer identities observed at non-padding positions across the selected probes.	$\mathcal{V}_{\mathrm{obs}} \subseteq \mathcal{V}$
$\widehat{\mathcal{T}}$	Token-identity removal set selected by PoisonTrace.	$\widehat{\mathcal{T}} \subseteq \mathcal{V}_{\mathrm{obs}}$
Probe and model notation		
M	Generic checkpoint role.	$\{C, P\}$
f_C, f_P	Clean-reference and suspect checkpoints under role-index notation.	$\mathbb{R}^{L\times d_e} \to \mathbb{R}^{\lvert\mathcal{Y}\rvert}$
f_M	Classifier associated with checkpoint role M.	$\mathbb{R}^{L\times d_e} \to \mathbb{R}^{\lvert\mathcal{Y}\rvert}$
i	Probe index.	$\{1, \dots, N\}$
j, k	Token-position indices.	$\{1, \dots, L\}$
E_i^M	Input-embedding sequence for probe i under checkpoint M.	$\mathbb{R}^{L\times d_e}$
t_{ij}	Tokenizer identity at position j of probe i.	$\mathcal{V}$
m_{ij}	Non-padding indicator at position j of probe i.	$\{0, 1\}$
ℓ_M	Checkpoint-specific classification loss.	$\ell_M : \mathbb{R}^{L\times d_e} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$
Input-Hessian curvature estimation		
H_i^M	Input-embedding Hessian of the classification loss for probe i under checkpoint M.	$\mathbb{S}^{Ld_e}$
H	Generic input Hessian used in the positional curvature definition.	$\mathbb{S}^{Ld_e}$
$H[j, k]$	Embedding-coordinate block of H connecting positions j and k.	$\mathbb{R}^{d_e\times d_e}$
$R_j(H)$	Full-spectrum Hessian block-row energy associated with position j.	$\mathbb{R}_{\geq 0}$
v	Generic Rademacher vector used in the matrix-free curvature identity.	$\{-1, +1\}^{Ld_e}$
I_{Ld_e}	Identity matrix appearing in $\mathbb{E}[vv^\top] = I_{Ld_e}$.	$\mathbb{R}^{Ld_e\times Ld_e}$
M_{H}	Number of randomized Hessian-vector-product (HVP) draws.	$\mathbb{N}_{>0}$
r	HVP-draw index.	$\{1, \dots, M_{\mathrm{H}}\}$
$v_i^{(r)}$	r-th Rademacher vector used for probe i.	$\{-1, +1\}^{Ld_e}$
$H_i^M v_i^{(r)}$	Hessian-vector product for probe i, checkpoint M, and random draw r.	$\mathbb{R}^{Ld_e}$
$(H_i^M v_i^{(r)})_j$	Embedding-coordinate block of the HVP associated with position j.	$\mathbb{R}^{d_e}$
$\widehat{R}_{ij}^M$	Hutchinson estimate of positional curvature energy at probe i, position j, and checkpoint M.	$\mathbb{R}_{\geq 0}$
$\widehat{\mathbf{R}}_i^M$	Raw positional curvature profile $(\widehat{R}_{i1}^M, \dots, \widehat{R}_{iL}^M)$.	$\mathbb{R}_{\geq 0}^L$
Clean–suspect normalization and smoothing		
ε_R	Positive numerical stabilizer used for curvature profile normalization and robust scoring.	$\mathbb{R}_{>0}$
$\widehat{e}_{ij}^M$	Within-probe normalized curvature share.	$[0, 1]$
s_{ij}	Positive clean–suspect curvature-share shift.	$[0, 1]$
$[a]_+$	Positive-part operator, $[a]_+ := \max(a, 0)$.	$\mathbb{R} \to \mathbb{R}_{\geq 0}$

Continued on next page

Table 5 – continued

Notation	Description	Dimension / Space
w	Local positional-smoothing radius.	$\mathbb{N}_0$
δ	Integer positional offset used during smoothing.	$\mathbb{Z}$
$\widetilde{e}_{ij}$	Locally smoothed, unnormalized curvature-shift energy.	$\mathbb{R}_{\geq 0}$
e_{ij}	Renormalized smoothed positional energy.	$[0, 1]$
Token-identity aggregation		
t	Generic tokenizer identity considered during cross-probe aggregation.	$\mathcal{V}_{\text{obs}}$
$\mathbf{1}\{\cdot\}$	Indicator function of a predicate.	$\{0, 1\}$
$\text{occ}(t)$	Number of non-padding occurrences of token identity t across all probes.	$\mathcal{V}_{\text{obs}} \to \mathbb{N}_0$
$\text{pf}(t)$	Fraction of probes containing token identity t.	$\mathcal{V}_{\text{obs}} \to [0, 1]$
$\text{score}_{\text{PP}}(t)$	PoisonTrace token-identity localization score.	$\mathcal{V}_{\text{obs}} \to \mathbb{R}_{\geq 0}$
$P_{\min}$	Minimum number of distinct probes in which a token identity must recur to retain its score.	$\{1, \ldots, N\}$
Data-driven removal-set selection		
S	Finite multiset of strictly positive PoisonTrace identity scores.	$S \subseteq \mathbb{R}_{>0}$
$\lvert S\rvert$	Number of elements in the positive-score multiset.	$\mathbb{N}_{>0}$
σ_k	k-th largest score in S.	$\mathbb{R}_{>0}$
m_S	Median of the positive-score multiset S.	$\mathbb{R}_{>0}$
MAD_S	Median absolute deviation of S.	$\mathbb{R}_{\geq 0}$
G	Maximum consecutive-gap index examined by the budget-selection rule.	$\mathbb{N}_{>0}$
k	Consecutive-gap index.	$\{1, \ldots, G\}$
ε_{gap}	Positive numerical stabilizer used in the consecutive score-ratio calculation.	$\mathbb{R}_{>0}$
$\widehat{k}_{\text{gap}}$	Estimated size of the separated high-score block.	$\{1, \ldots, G\}$
$z_{\text{rob}}(t)$	Robust standardized localization score for token identity t.	$\mathcal{V}_{\text{obs}} \to \mathbb{R}$
$z^\star$	Fixed robust-score threshold.	$\mathbb{R}$
$\widehat{\mathcal{T}}_{\text{conf}}$	Set of token identities satisfying the robust-score threshold.	$\widehat{\mathcal{T}}_{\text{conf}} \subseteq \mathcal{V}_{\text{obs}}$
$K_{\max}$	Maximum permitted removal-set size.	$\mathbb{N}_{>0}$
$\widehat{K}$	Automatically selected removal budget.	$\{0, \ldots, K_{\max}\}$
n	Candidate high-score-block size used in the gap-recovery argument.	$\{1, \ldots, G\}$
Training and mitigation		
$f^{(\rho)}_{\text{poison}}$	Victim checkpoint trained under poison fraction ρ.	$\mathbb{R}^{L \times d_e} \to \mathbb{R}^{\lvert\mathcal{Y}\rvert}$
$f^{(\tau,\rho)}_{\text{cleaned}}$	Classifier retrained after scrubbing the automatically selected token identities.	$\mathbb{R}^{L \times d_e} \to \mathbb{R}^{\lvert\mathcal{Y}\rvert}$
p	Valid word-boundary position at which trigger τ is inserted.	Discrete valid-position set for x
$T_{\tau,p}(x)$	Input obtained by inserting trigger τ into x at valid position p.	Σ^*
Evaluation notation		
f	Generic classifier used in the ASR definition.	$\mathbb{R}^{L \times d_e} \to \mathbb{R}^{\lvert\mathcal{Y}\rvert}$
$\text{ASR}(f, \tau, p)$	Attack success rate for model f under trigger τ and insertion position p.	$[0, 1]$
$\text{ASR}_{\text{clean}}$	Triggered-input ASR of the clean-reference model.	$[0, 1]$

Continued on next page

Table 5 – continued

Notation	Description	Dimension / Space
$\mathrm{ASR}_{\mathrm{poison}}$	Triggered-input ASR of the poisoned model.	$[0, 1]$
$\mathrm{ASR}_{\mathrm{cleaned}}$	Triggered-input ASR after mitigation.	$[0, 1]$
r_k	Tokenizer identity at rank k in a localization ranking.	$\mathcal{V}_{\mathrm{obs}}$
R_n	Set of the top-n ranked tokenizer identities, $R_n = \{r_1, \ldots, r_n\}$.	$R_n \subseteq \mathcal{V}_{\mathrm{obs}}$
R_{n_T}	Set of the top-n_T ranked tokenizer identities used for ranking-only localization evaluation.	$R_{n_T} \subseteq \mathcal{V}_{\mathrm{obs}}, \quad \lvert R_{n_T}\rvert = n_T$
$\mathrm{Acc}_{\mathrm{clean}}$	Clean-reference test accuracy.	$[0, 1]$
$\mathrm{Acc}_{\mathrm{poison}}$	Poisoned-model clean-task test accuracy.	$[0, 1]$
$\mathrm{Acc}_{\mathrm{cleaned}}$	Clean-task test accuracy after mitigation.	$[0, 1]$
$\Delta\mathrm{Acc}$	Accuracy change after mitigation, $\mathrm{Acc}_{\mathrm{cleaned}} - \mathrm{Acc}_{\mathrm{poison}}$.	$[-1, 1]$

B PoisonTrace

All the hyperparameter choices are in Appendix E

B.1 Complete PoisonTrace Algorithm

Algorithm 1 PoisonTrace: Automatic Probe Acquisition, Per-Probe Curvature-Shift Localization, and Defense

Input: Trusted clean-reference checkpoint f_C and flagged suspect checkpoint f_P of the same differentiable token-classification architecture, with accessible input embeddings, classification loss, and HVP support; fine-tuning training corpus $\mathcal{D}_P = \{(x_a, y_a)\}_{a=1}^{m}$ with observed labels $y_a \in \mathcal{Y}$; probe budget N; CleanShift tail fraction α; CleanShift numerical stabilizer $\varepsilon_{\mathrm{CS}}$; Rademacher draws M_{H}; smoothing radius w; support floor $P_{\min}$; robust threshold $z^\star$; gap-search range G; safety ceiling $K_{\max}$; common retraining initialization f_0.

Output: Automatically acquired probe set $\mathcal{P}$; CleanShift trigger-associated class estimate $\widehat{y}_{\mathrm{CS}}$; ranked token identities; removal set $\widehat{\mathcal{T}}$, possibly empty; cleaned model f_D when mitigation is performed; probe-acquisition, localization, and mitigation metrics.

Step 1: Acquire suspicious probes with CleanShift

For each checkpoint role $M \in \{C, P\}$, compute class probabilities for every training example:

$$p_M(c \mid x_a) \leftarrow \mathrm{softmax}(f_M(x_a))_c, \qquad a \in [m], \quad c \in \mathcal{Y}.$$

Class-wise clean–suspect shift normalization

for each candidate class $c \in \mathcal{Y}$ **do**
  for $a = 1$ to m **do**
    Compute the corpus-wide clean–suspect probability shift:

$$\Delta_a(c) \leftarrow p_P(c \mid x_a) - p_C(c \mid x_a).$$

  end for
  Compute the robust corpus-wide location:

$$\mu_c^{\mathrm{CS}} \leftarrow \mathrm{median}_{a\in[m]}\, \Delta_a(c).$$

  Compute the robust corpus-wide scale:

$$s_c^{\mathrm{CS}} \leftarrow \mathrm{median}_{a\in[m]} \left|\Delta_a(c) - \mu_c^{\mathrm{CS}}\right|.$$

  for $a = 1$ to m **do**
    Compute the normalized residual shift:

$$r_a(c) \leftarrow \frac{\Delta_a(c) - \mu_c^{\mathrm{CS}}}{s_c^{\mathrm{CS}} + \varepsilon_{\mathrm{CS}}}.$$

  end for
end for

Corpus-level trigger-associated class inference

for each class $c \in \mathcal{Y}$ **do**
  Form the observed-label partition:

$$\mathcal{D}_c \leftarrow \{a \in [m] : y_a = c\}, \qquad n_c \leftarrow |\mathcal{D}_c|.$$

Set the proportional upper-tail budget:

$$B_c \leftarrow \max(1, \lceil \alpha n_c \rceil).$$

Sort the normalized residual shifts within $\mathcal{D}_c$:

$$r_{(1)}(c) \geq r_{(2)}(c) \geq \cdots \geq r_{(n_c)}(c).$$

Compute the CleanShift class-tail statistic:

$$T_{\mathrm{CS}}(c) \leftarrow \frac{1}{B_c} \sum_{b=1}^{B_c} r_{(b)}(c).$$

end for

Select the class with the strongest normalized upper-tail shift:

$$\widehat{y}_{\mathrm{CS}} \leftarrow \arg\max_{c \in \mathcal{Y}} T_{\mathrm{CS}}(c).$$

Probe scoring and selection

Restrict the candidate pool to examples carrying the inferred trigger-associated observed label:

$$\mathcal{E}_{\mathrm{CS}} \leftarrow \{a \in [m] : y_a = \widehat{y}_{\mathrm{CS}}\}.$$

for each $a \in \mathcal{E}_{\mathrm{CS}}$ **do**

Assign the normalized CleanShift sample score:

$$s_a^{\mathrm{CS}} \leftarrow r_a(\widehat{y}_{\mathrm{CS}}).$$

end for

Select the indices of the N highest-scoring eligible examples:

$$\mathcal{I}_{\mathrm{probe}} \leftarrow \mathrm{Top}_{N,\, a \in \mathcal{E}_{\mathrm{CS}}}\, s_a^{\mathrm{CS}}.$$

Construct and freeze the acquired textual probe set:

$$\mathcal{P} \leftarrow \{x_a : a \in \mathcal{I}_{\mathrm{probe}}\}.$$

Encode each selected probe $i \in [N]$ as embeddings E_i, token identities t_{ij}, and padding masks m_{ij}.

Step 2: Fix the class used to condition PoisonTrace

Use the CleanShift estimate $\widehat{y}_{\mathrm{CS}}$ directly as the class conditioning the subsequent PoisonTrace localization loss. The selected probes and inferred class remain fixed during localization.

Step 3: Estimate full-spectrum positional curvature

for $M \in \{C, P\}$ **do**

Reset the Rademacher generator to the same seed for both checkpoints.

Initialize

$$\widehat{R}^M \leftarrow 0 \in \mathbb{R}^{N \times L}.$$

For probe x_i, let its input-embedding matrix be

$$E_i = \begin{bmatrix} e_{i1}^\top \\ e_{i2}^\top \\ \vdots \\ e_{iL}^\top \end{bmatrix} \in \mathbb{R}^{L \times d_e},$$

where $e_{ij} \in \mathbb{R}^{d_e}$ denotes the embedding vector at token position j.

Define the per-probe input-embedding Hessian:

$$H_i^M \leftarrow \frac{\partial^2 \ell_M(E_i, \widehat{y}_{\mathrm{CS}})}{\partial E_i^2}.$$

for $r = 1$ to M_{H} **do**

Draw a masked Rademacher tensor

$$v^{(r)} \in \mathbb{R}^{N \times L \times d_e}.$$

Compute the independent per-probe HVPs

$$H_i^M v_i^{(r)}$$

using one probe-batched double-backpropagation.

Accumulate positional HVP energy:

$$\widehat{R}_{ij}^M \mathrel{+}= \left\| (H_i^M v_i^{(r)})_j \right\|_2^2.$$

end for

Average over the randomized directions:

$$\widehat{R}_{ij}^M \leftarrow \frac{\widehat{R}_{ij}^M}{M_{\mathrm{H}}}.$$

Mask padding and normalize within each probe:

$$\widehat{e}_{ij}^M \leftarrow \frac{\widehat{R}_{ij}^M m_{ij}}{\sum_{k=1}^{L} \widehat{R}_{ik}^M m_{ik} + \varepsilon_R}.$$

end for

Step 4: Compute the positive clean–suspect curvature shift

for $i = 1$ to N **do**
  for $j = 1$ to L **do**

$$s_{ij} \leftarrow \mathrm{ReLU}\left(\widehat{e}^{P}_{ij} - \widehat{e}^{C}_{ij}\right).$$

  end for
end for

Step 5: Smooth local sub-token evidence

for $i = 1$ to N **do**
  for $j = 1$ to L **do**
    Pool evidence within radius w:

$$\widetilde{e}_{ij} \leftarrow \sum_{\substack{|\delta| \le w \\ 1 \le j+\delta \le L}} s_{i,j+\delta}.$$

  end for
  Renormalize the smoothed profile:

$$e_{ij} \leftarrow \frac{\widetilde{e}_{ij} m_{ij}}{\sum_{k=1}^{L} \widetilde{e}_{ik} m_{ik} + \varepsilon_R}.$$

end for

Step 6: Aggregate evidence across probes by tokenizer identity

Construct the observed identity set:

$$\mathcal{V}_{\mathrm{obs}} \leftarrow \{t : \exists i, j \text{ such that } m_{ij} = 1 \text{ and } t_{ij} = t\}.$$

for each $t \in \mathcal{V}_{\mathrm{obs}}$ **do**
  Compute occurrence count:

$$\mathrm{occ}(t) \leftarrow \sum_{i=1}^{N} \sum_{j=1}^{L} m_{ij} \mathbf{1}\{t_{ij} = t\}.$$

  Compute the number of supporting probes:

$$P(t) \leftarrow |\{i : \exists j,\ m_{ij} = 1,\ t_{ij} = t\}|.$$

  Compute probe-frequency support:

$$\mathrm{pf}(t) \leftarrow \frac{P(t)}{N}.$$

  Compute the PoisonTrace identity score:

$$\mathrm{score}_{\mathrm{PP}}(t) \leftarrow \frac{\sum_{i=1}^{N} \sum_{j=1}^{L} e_{ij} m_{ij} \mathbf{1}\{t_{ij} = t\}}{\max\{\mathrm{occ}(t), 1\}}\, \mathrm{pf}(t).$$

  if $P(t) < P_{\min}$ **then**

$$\mathrm{score}_{\mathrm{PP}}(t) \leftarrow 0.$$

  end if
end for

Step 7: Form the positive-score population

$$S \leftarrow \{\mathrm{score}_{\mathrm{PP}}(t) : t \in \mathcal{V}_{\mathrm{obs}},\ \mathrm{score}_{\mathrm{PP}}(t) > 0\}.$$

if $S = \varnothing$ **then**

$$\widehat{K} \leftarrow 0, \qquad \widehat{\mathcal{T}} \leftarrow \varnothing.$$

  Go to Step 11 and return the abstention decision.
end if

Step 8: Estimate the candidate-block size from score separation

$$S_+ := \{\mathrm{Score}_{\mathrm{PP}}(t) : \mathrm{Score}_{\mathrm{PP}}(t) > 0\}, \qquad G := |S_+|.$$

Then, $\sigma_k = \mathrm{Score}_{\mathrm{PP}}\left(t_{(k)}\right)$ and $t_{(k)}$ denotes the token identity with the k-th largest positive score. Sort the positive scores:

$$\sigma_1 \ge \sigma_2 \ge \cdots > 0.$$

Compute the largest consecutive score-ratio gap:

$$\widehat{k}_{\mathrm{gap}} \leftarrow \arg\max_{1 \le k \le G} \frac{\sigma_k}{\sigma_{k+1} + \varepsilon_{\mathrm{gap}}}.$$

Step 9: Construct the robust high-score set

Compute the robust background location:

$$m_S \leftarrow \mathrm{median}(S).$$

Compute the Median Absolute Deviation (MAD):

$$\mathrm{MAD}_S \leftarrow \mathrm{median}_{s \in S} \, |s - m_S| \, .$$

for each $t \in \mathcal{V}_{\mathrm{obs}}$ with $\mathrm{score}_{\mathrm{PP}}(t) > 0$ **do**

  Compute the robust standardized score:

$$z_{\mathrm{rob}}(t) \leftarrow \frac{\mathrm{score}_{\mathrm{PP}}(t) - m_S}{\mathrm{MAD}_S + \varepsilon_R}.$$

end for

Construct the robust high-score set:

$$\widehat{\mathcal{T}}_{\mathrm{conf}} \leftarrow \{t \in \mathcal{V}_{\mathrm{obs}} : z_{\mathrm{rob}}(t) \geq z^{\star}\} \, .$$

Step 10: Select the removal budget or abstain

if $\widehat{\mathcal{T}}_{\mathrm{conf}} = \varnothing$ **then**

$$\widehat{K} \leftarrow 0, \qquad \widehat{\mathcal{T}} \leftarrow \varnothing.$$

else

  Select the data-driven removal budget:

$$\widehat{K} \leftarrow \min \left(\widehat{k}_{\mathrm{gap}}, \left| \widehat{\mathcal{T}}_{\mathrm{conf}} \right|, K_{\max} \right) .$$

  Select the top-$\widehat{K}$ identities:

$$\widehat{\mathcal{T}} \leftarrow \mathrm{Top}_{\widehat{K},\, t \in \mathcal{V}_{\mathrm{obs}}} \, \mathrm{score}_{\mathrm{PP}}(t).$$

end if

Step 11: Mitigate or abstain

if $\widehat{K} > 0$ **then**

  Scrub exactly the selected identities:

$$D_{\mathrm{scrub}} \leftarrow \mathrm{Scrub} \left(\mathcal{D}_P, \widehat{\mathcal{T}} \right) .$$

  Retrain from the common initialization:

$$f_D \leftarrow \mathrm{Train} \left(f_0, D_{\mathrm{scrub}} \right) .$$

else

  Abstain and leave the suspect checkpoint unmodified.

end if

Step 12: Report evaluation outputs

Report the acquired probe set $\mathcal{P}$.

If poisoned-row and trigger annotations are available only for controlled evaluation, report PoisonedRowPrecision@N and TriggerBearingRate@N.

Report the complete token ranking and $\widehat{\mathcal{T}}$.

If trigger ground truth is available only for controlled evaluation, report localization metrics including Precision@n_T, Exact@n_T, removal-set precision/recall, and exact removal-set recovery.

Report mitigation metrics including clean accuracy, ASR before mitigation, ASR after mitigation, ASR drop, residual ASR relative to f_C, and ΔAcc.

Optional deployment-validation branch. The benign-drift and candidate-deletion steps in Algorithm 1 are intended for auditing a post-fine-tuning checkpoint whose poison status is genuinely unknown. The reported localisation experiments do not execute this branch: they use the base PoisonTrace path with CleanShift distilled probes Accordingly, the existing localization and mitigation tables are not recomputed by this pseudocode update.

C Probe-Acquisition Details

C.1 CleanShift Methodology

This appendix gives the complete definition of the probe-acquisition rule referred to as *CleanShift* in the main text. CleanShift has two objectives: first, to identify the class exhibiting the strongest localized clean–suspect prediction shift after accounting for class-wide fine-tuning effects; and second, to select the N textual probes consumed by PoisonTrace. The token-localization method itself remains the per-probe input-Hessian procedure in Section 4.

Inputs and notation. Let $\mathcal{Y}$ denote the finite class space and let the fine-tune training corpus be

$$\mathcal{D}_P = \{(x_a, y_a)\}_{a=1}^{m}, \tag{3}$$

where $m := |\mathcal{D}_P|$ is the number of examples in the fine-tuning training corpus, x_a is a textual training example, and $y_a \in \mathcal{Y}$ is its *observed* training label. For checkpoint role $M \in \{C, P\}$, define

$$p_M(c \mid x_a) := \mathrm{softmax}(f_M(x_a))_c, \qquad c \in \mathcal{Y}. \tag{4}$$

The procedure uses only f_C, f_P, the suspect corpus, its observed labels, the tokenizer and preprocessing pipeline, and the fixed probe budget N. It is not supplied with the trigger string, trigger-token identities, trigger length or positions, poisoning rate, poisoned-row identities, or attacker-chosen target class.

Class-wise clean–suspect shift normalization. Before partitioning the training examples by observed label, CleanShift first measures the prediction change induced by fine-tuning for every candidate class. For each $a \in [m]$ and $c \in \mathcal{Y}$, define

$$\Delta_a(c) := p_P(c \mid x_a) - p_C(c \mid x_a). \tag{5}$$

A raw positive value of $\Delta_a(c)$ does not necessarily indicate poisoning. In particular, benign fine-tuning or class imbalance may produce a systematic increase in the suspect checkpoint's confidence toward one class across a substantial portion of the corpus. CleanShift therefore removes the class-wide background shift before searching for a concentrated anomalous upper tail.

For each candidate class c, define the robust location

$$\mu_c^{\mathrm{CS}} := \mathrm{median}_{a \in [m]}\, \Delta_a(c), \tag{6}$$

and robust scale

$$s_c^{\mathrm{CS}} := \mathrm{median}_{a \in [m]}\, \left|\Delta_a(c) - \mu_c^{\mathrm{CS}}\right|. \tag{7}$$

The normalized residual shift is then

$$r_a(c) := \frac{\Delta_a(c) - \mu_k^{\mathrm{CS}}}{s_c^{\mathrm{CS}} + \varepsilon_{\mathrm{CS}}}. \tag{8}$$

Where $\varepsilon_{\mathrm{CS}}$ is a numerical-stability constant, as declared in Appendix E.

Thus, $r_a(c)$ measures how unusually large the clean–suspect prediction shift of example x_a is relative to the typical shift toward class c across the suspect corpus. The median and median absolute deviation are used because they are less sensitive than the mean and standard deviation to the relatively small set of examples that may carry an unusually strong poisoning-associated signal.

This normalization is intended to reduce sensitivity to systematic class-wise prediction shifts caused by ordinary fine-tuning or class imbalance. It does not assume that such shifts are caused by poisoning.

Corpus-level trigger-associated class inference. The objective of this stage is to estimate the unknown attacker-chosen target class, denoted by $y^\star$, using only the clean–suspect prediction shifts and the observed training labels.

After class-wise normalization, examples are partitioned according to their observed training labels:

$$\mathcal{D}_c := \{a \in [m] : y_a = c\}, \qquad n_c := |\mathcal{D}_c|. \tag{9}$$

If c is associated with the backdoor target, trigger-bearing examples carrying observed label c are expected to produce a concentrated upper tail of unusually large residual shifts $r_a(c)$, rather than merely a class-wide increase in confidence.

To reduce direct dependence on class size, CleanShift compares the same upper-tail *proportion* of each observed-label class. Let $\alpha \in (0, 1]$ be a fixed tail fraction, chosen before held-out evaluation and independent of the unknown poisoning rate. We chose $\alpha = 0.01$ for our experiments. Define

$$B_c := \max(1, \lceil \alpha n_c \rceil). \tag{10}$$

Let

$$r_{(1)}(c) \geq r_{(2)}(c) \geq \cdots \geq r_{(n_k)}(c)$$

denote the normalized residual shifts sorted in descending order within $\mathcal{D}_c$. The class-tail statistic is

$$T_{\mathrm{CS}}(c) := \frac{1}{B_c} \sum_{b=1}^{B_c} r_{(b)}(c), \tag{11}$$

and the trigger-associated class used for probe acquisition is

$$\widehat{y}_{\mathrm{CS}} := \arg\max_{c \in \mathcal{Y}} T_{\mathrm{CS}}(c). \tag{12}$$

Hence, $\widehat{y}_{\mathrm{CS}}$ is the observed-label class whose upper tail exhibits the strongest positive deviation relative to its own class-wise clean–suspect background shift. Thus, $\widehat{y}_{\mathrm{CS}}$ is CleanShift's estimate of the unknown true backdoor target class $y^\star$. The combination of robust residualization and proportional upper-tail comparison addresses two distinct effects of dataset imbalance: systematic class-wise confidence changes are removed before partitioning, while unequal numbers of examples are accounted for when constructing the class-tail statistics.

The parameter α determines the fraction of each class used for class inference, whereas N remains exclusively the downstream probe budget. Neither parameter is an estimate of the true poisoning rate.

Probe scoring and selection. After determining $\widehat{y}_{\mathrm{CS}}$, the eligible candidate pool is

$$\mathcal{E}_{\mathrm{CS}} := \{a \in [m] : y_a = \widehat{y}_{\mathrm{CS}}\}. \tag{13}$$

Each eligible example receives the normalized CleanShift score

$$s_a^{\mathrm{CS}} := r_a(\widehat{y}_{\mathrm{CS}}), \qquad a \in \mathcal{E}_{\mathrm{CS}}. \tag{14}$$

A high value of s_a^{CS} therefore means that the suspect checkpoint's shift toward $\widehat{y}_{\mathrm{CS}}$ is unusually large relative to the ordinary corpus-wide fine-tuning shift toward that same class.

The selected index set and textual probe set are

$$\mathcal{I}_{\mathrm{probe}} := \mathrm{Top}_{N,\, a \in \mathcal{E}_{\mathrm{CS}}}\, s_a^{\mathrm{CS}}, \qquad \mathcal{P} := \{x_a : a \in \mathcal{I}_{\mathrm{probe}}\}. \tag{15}$$

The selected examples are fixed and passed unchanged to PoisonTrace. False-positive probes are not replaced after post-hoc inspection, and the trigger is never injected into a selected probe. Consequently, ground-truth trigger information does not enter the probe-acquisition procedure.

Post-hoc probe-quality diagnostics. For controlled evaluation only, after probe selection has been fixed, we report

$$\begin{aligned} \mathrm{PoisonedRowPrecision@}N &= \frac{\#\{\text{selected probes drawn from poisoned training rows}\}}{N}, \\ \mathrm{TriggerBearingRate@}N &= \frac{\#\{\text{selected probes containing the complete trigger in the model input}\}}{N}. \end{aligned} \tag{16}$$

These quantities evaluate probe quality only; they do not enter class-wise normalization, trigger-associated class inference, probe selection, PoisonTrace curvature estimation, token ranking, removal-budget selection, corpus scrubbing, or retraining.

C.2 Sensitivity Analysis: Robustness to Probe Purity

The controlled experiments initially assume that all $N = 20$ probes contain the complete trigger. We relax this assumption through a probe-purity ablation that varies the number of trigger-bearing probes while holding the total probe budget fixed. Let

$$q = \frac{k}{N}, \tag{17}$$

where k is the number of trigger-bearing probes and q is the resulting probe purity. We evaluate

$$k \in \{0, 1, 2, 4, 10, 16, 20\}, \qquad q \in \{0, 5, 10, 20, 50, 80, 100\}\%. \tag{18}$$

Only the presence of the trigger is varied. For each condition, k probes retain the complete trigger, whereas the remaining $20 - k$ probes are the same reviews with the inserted trigger block removed. Consequently, the underlying textual content and the total probe budget remain controlled. The $q = 100\%$ condition is identical to the original controlled probe setting, while $q = 0\%$ serves as a

Table 6: Probe-purity ablation on IMDB at $L = 128$ and $\rho = 1\%$. Results are means over the 22 successfully installed attacks and, for intermediate purity levels, five choices of which k probes contain the trigger. $\mathrm{P@}n_\tau$, $\mathrm{R@}n_\tau$, and $\mathrm{Exact@}n_\tau$ evaluate the top-n_τ token ranking. The removal-set columns evaluate the data-driven set $\widehat{\mathcal{T}}$. "Abstain" is the fraction of runs in which $\widehat{K} = 0$.

Probe Composition		**Top-n_τ Localization**			**Data-Driven Removal Set**					**Output**
Purity	**Trigger probes**	$\mathbf{P@}n_\tau$	$\mathbf{R@}n_\tau$	$\mathbf{Exact@}n_\tau$	**Mean $\widehat{K}$**	**Precision**	**Recall**	**F1**	**Exact**	**Abstain**
0%	0/20	0.152	0.167	0.000	4.7	0.194	0.077	0.072	0.000	0.000
5%†	1/20	0.199	0.226	0.018	4.5	0.321	0.118	0.129	0.000	0.000
10%	2/20	0.346	0.375	0.064	5.5	0.464	0.353	0.263	0.018	0.000
20%	4/20	0.887	0.932	0.682	4.7	0.896	0.786	0.791	0.418	0.000
50%	10/20	0.916	0.970	0.827	4.0	0.955	0.805	0.857	0.509	0.036
80%	16/20	0.927	0.982	**0.873**	4.3	0.991	0.840	0.881	0.609	0.009
100%	20/20	**0.931**	**0.987**	0.864	5.8	**1.000**	**0.902**	**0.941**	**0.636**	0.000

† PoisonTrace requires a token identity to appear in at least two probes. Therefore, when the trigger is present in only one probe, its inserted occurrences do not satisfy the minimum-support condition. Any nonzero recovery at this rung can arise only when a trigger identity also occurs naturally elsewhere in the probe set.

null control in which no probe contains the trigger. For each intermediate purity level, results are averaged over five different choices of which k probes contain the trigger.

All PoisonTrace settings are held fixed across the ablation, including the Hutchinson sample count, contextual span, token-scoring rule, minimum probe-support requirement, and data-driven removal-budget selector. We report the principal results over the 22 of 26 trigger trials for which the attack successfully installed. An attack is considered installed when the suspect-model ASR is at least 50% and exceeds the corresponding clean-model ASR by at least 20 percentage points.

Localization under imperfect probe purity. Table 6 shows a pronounced transition between 10% and 20% probe purity. With only two trigger-bearing probes, PoisonTrace obtains $\mathrm{P@}n_\tau = 0.346$, $\mathrm{R@}n_\tau = 0.375$, and $\mathrm{Exact@}n_\tau = 0.064$. When four of the 20 probes contain the trigger, these values increase to 0.887, 0.932, and 0.682, respectively. Thus, the localizer does not require every probe to contain the trigger in this controlled setting; four trigger-bearing probes are sufficient for strong mean localization performance.

Performance subsequently plateaus as probe purity increases. Between 50% and 100% purity, $\mathrm{P@}n_\tau$ ranges from 0.916 to 0.931, while $\mathrm{R@}n_\tau$ ranges from 0.970 to 0.987. The small variation across the 50%, 80%, and 100% conditions suggests that, after sufficient recurring trigger evidence is available, increasing the number of trigger-bearing probes provides only limited additional ranking improvement.

Ranking versus removal-set recovery. Exact recovery of the final removal set is consistently lower than exact recovery of the top-n_τ ranking. At 100% purity, the correct trigger set occupies the top n_τ positions in 86.4% of installed attacks, whereas the data-driven removal set is exactly equal to the trigger set in 63.6% of attacks. The corresponding removal-set precision, recall, and F1 score are 1.000, 0.902, and 0.941, respectively. This gap indicates that the token-ranking stage is generally more reliable than the automatic removal-budget selector: exact removal requires both a correct ranking and a correctly estimated stopping point $\widehat{K}$.

Across the installed attacks, 19 of 22 victims achieve exact top-n_τ recovery at some tested purity level, and the median requirement among these victims is four trigger-bearing probes. By comparison, 14 of 22 victims achieve exact removal-set recovery, with a median requirement of ten trigger-bearing probes. These results further identify data-driven removal-budget estimation as a remaining source of error.

Connection to probe-frequency weighting. The probe-purity ablation directly evaluates the dependence of PoisonTrace on the probe-frequency term

$$\mathrm{pf}(t) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[t \in x_i]\,, \tag{19}$$

which enters the token score as

$$\mathrm{score}_{\mathrm{PP}}(t) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{L} e_{ij} m_{ij} \mathbf{1}\{t_{ij} = t\}}{\max\{\mathrm{occ}(t), 1\}}\,\mathrm{pf}(t). \tag{20}$$

The fraction is the mean clean–suspect curvature evidence per occurrence of token identity t, with $\max\{\mathrm{occ}(t), 1\}$ providing a safe zero-occurrence denominator without altering any positive occurrence count. The factor $\mathrm{pf}(t)$ rewards evidence that recurs across multiple probes. Let $q = k/N$ denote probe purity, where k of the N probes contain the complete trigger. For a trigger identity that occurs in every trigger-bearing probe and does not occur naturally in the remaining probes, $\mathrm{pf}(t) = q$. Consequently, reducing probe purity from 100% to 20% reduces the trigger's recurrence multiplier from 1 to 0.2, even when its per-occurrence curvature evidence remains strong. The ablation therefore tests whether the clean–suspect Hessian signal is sufficiently discriminative to survive this deliberate frequency dilution and still rank the trigger above benign identities that may occur in more probes. Probe purity and $\mathrm{pf}(t)$ are not generally identical, however, because a trigger subtoken may also occur naturally in non-trigger-bearing probes; probe purity is a property of the probe set, whereas $\mathrm{pf}(t)$ is specific to each tokenizer identity.

Null-control behavior. When no probe contains the trigger, both $\mathrm{Exact@}n_\tau$ and exact removal are zero, as expected. The nonzero mean precision and recall values at $q = 0\%$ arise from coincidental overlap between selected identities and the known trigger identities used for evaluation, rather than from trigger evidence in the probes. More importantly, the mean estimated removal budget remains $\widehat{K} = 4.7$, and the method does not abstain in the null condition. This reveals that the current budget selector can return unrelated token identities even when no trigger-bearing evidence is available. An explicit null-detection or abstention criterion is therefore needed before applying corpus-wide removal in an open-world deployment.

Overall, the ablation demonstrates that PoisonTrace can tolerate substantial contamination of its candidate probe pool once at least four of the 20 probes contain the trigger. However, this controlled experiment evaluates the robustness of the localizer to known probe purity; it does not by itself evaluate the accuracy of the CleanShift probe-acquisition procedure.

D Supplementary Derivations for PoisonTrace

As in the main text, the derivation is organized around three equivalent representations of the positional curvature functional: what quantity is measured, why it contains the full Hessian spectrum, and how it can be estimated using randomized Hessian–vector products without constructing or eigendecomposing the Hessian.

For each probe, PoisonTrace requires a positional curvature statistic that includes all Hessian modes but can be evaluated without forming or eigendecomposing the Hessian. Partition a symmetric Hessian $H \in \mathbb{R}^{Ld_e \times Ld_e}$ into blocks $H[j,k] \in \mathbb{R}^{d_e \times d_e}$, and write $H = \sum_q \lambda_q \phi_q \phi_q^\top$, where $\phi_{q,j} \in \mathbb{R}^{d_e}$ is the block of eigenvector ϕ_q at position j. Here $H \in \mathbb{S}^{Ld_e}$; $j, k \in [L]$; $q \in [Ld_e]$ indexes the eigenpairs; $\lambda_q \in \mathbb{R}$; the eigenvectors $\phi_q \in \mathbb{R}^{Ld_e}$ are orthonormal; and $H[j,k] \in \mathbb{R}^{d_e \times d_e}$. For the Rademacher vector below, $v \in \{-1, +1\}^{Ld_e}$ and $I = I_{Ld_e} \in \mathbb{R}^{Ld_e \times Ld_e}$. For each $j \in [L]$, $R_j : \mathbb{S}^{Ld_e} \to \mathbb{R}_{\geq 0}$.

D.1 Full-Spectrum Positional Block-Row Functional

Let $H \in \mathbb{S}^{Ld_e}$ be partitioned into blocks $H[j,k] \in \mathbb{R}^{d_e \times d_e}$, and let

$$H = \sum_q \lambda_q \phi_q \phi_q^\top,$$

where $\lambda_q \in \mathbb{R}$ and the eigenvectors $\phi_q \in \mathbb{R}^{Ld_e}$ are orthonormal. Let $\phi_{q,j} \in \mathbb{R}^{d_e}$ denote the block of ϕ_q corresponding to position j.

For each $j \in [L]$, define

$$R_j(H) := \underbrace{\sum_{k=1}^{L} \|H[j,k]\|_F^2}_{\substack{\text{quantity being measured:} \\ \text{block-row energy at position } j}} = \underbrace{\sum_{q} \lambda_q^2 \|\phi_{q,j}\|_2^2}_{\substack{\text{why it is full-spectrum:} \\ \text{every Hessian eigenmode contributes}}} = \underbrace{\mathbb{E}_v \|(Hv)_j\|_2^2}_{\substack{\text{how randomized HVPs estimate it:} \\ \text{matrix-free expectation}}} \tag{21}$$

where $v \in \{-1, +1\}^{Ld_e}$ is a Rademacher vector satisfying $\mathbb{E}[vv^\top] = I$.

Thus, the first expression defines the positional quantity, the second gives its full-spectrum interpretation, and the third provides a matrix-free characterization of the same quantity.

Block-row representation. Let $P_j \in \mathbb{R}^{d_e \times Ld_e}$ select the embedding coordinates corresponding to position j. Then

$$P_j H = [H[j,1] \quad \cdots \quad H[j,L]],$$

so

$$R_j(H) = \sum_{k=1}^{L} \|H[j,k]\|_F^2 = \|P_j H\|_F^2. \tag{22}$$

Full-spectrum representation. Because H is symmetric,

$$HH^\top = H^2 = \sum_q \lambda_q^2 \phi_q \phi_q^\top.$$

Moreover, the equivalent spectral form $R_j(H) = \sum_q \lambda_q^2 \|\phi_{q,j}\|_2^2$ shows that the score includes contributions from all Hessian eigenmodes, where (λ_q, ϕ_q) is the qth eigenpair and $\phi_{q,j}$ is the position-j block of ϕ_q. Therefore,

$$\begin{aligned}
R_j(H) &= \|P_j H\|_F^2 && (23)\\
&= \operatorname{tr}\big(P_j H H^\top P_j^\top\big) && (24)\\
&= \sum_q \lambda_q^2 \operatorname{tr}\big(P_j \phi_q \phi_q^\top P_j^\top\big) && (25)\\
&= \sum_q \lambda_q^2 \|P_j \phi_q\|_2^2 && (26)\\
&= \sum_q \lambda_q^2 \|\phi_{q,j}\|_2^2. && (27)
\end{aligned}$$

Hence every Hessian eigenmode contributes to $R_j(H)$. Its contribution is weighted by the squared eigenvalue λ_q^2 and by the amount of the corresponding eigenvector localized at position j through $\|\phi_{q,j}\|_2^2$. The eigendecomposition is used only to interpret the functional; PoisonTrace does not compute these eigenpairs.

Randomized-HVP representation. For a Rademacher vector v satisfying $\mathbb{E}[vv^\top] = I$,

$$\begin{aligned}
\mathbb{E}_v \|(Hv)_j\|_2^2 &= \mathbb{E}_v \|P_j H v\|_2^2 && (28)\\
&= \mathbb{E}_v \left[v^\top H^\top P_j^\top P_j H v\right] && (29)\\
&= \operatorname{tr}\big(H^\top P_j^\top P_j H \, \mathbb{E}[vv^\top]\big) && (30)\\
&= \operatorname{tr}\big(H^\top P_j^\top P_j H\big) && (31)\\
&= \|P_j H\|_F^2 && (32)\\
&= \sum_{k=1}^{L} \|H[j,k]\|_F^2 && (33)\\
&= R_j(H). && (34)
\end{aligned}$$

This proves the three-way identity in Eq. equation 21.

D.2 HVP Computation by Automatic Differentiation

For probe i and checkpoint $M \in \{C, P\}$, let

$$z_i = \mathrm{vec}(E_i), \qquad g_i^M(z_i) = \nabla_{z_i} \ell_M(z_i, y^\star), \qquad H_i^M = \nabla_{z_i}^2 \ell_M(z_i, y^\star).$$

For any fixed direction $v \in \mathbb{R}^{Ld_e}$,

$$H_i^M v = \nabla_{z_i} \left(g_i^M(z_i)^\top v \right). \tag{35}$$

Automatic differentiation therefore computes the Hessian action $H_i^M v$ directly, without constructing the matrix H_i^M. The HVP itself is the exact Hessian action for the chosen direction; the Monte Carlo approximation arises only from averaging several randomized squared HVP block norms in order to estimate $R_j(H_i^M)$.

D.3 Finite Randomized-HVP Estimator and Unbiasedness

Let $M_{\mathrm{H}} \in \mathbb{N}_{>0}$ be the number of randomized HVP draws. For each $r \in [M_{\mathrm{H}}]$, let $v_i^{(r)} \in \{-1, +1\}^{Ld_e}$ be an independent Rademacher direction, and let $\widehat{R}_{ij}^M \in \mathbb{R}_{\geq 0}$ denote the resulting estimator. For probe i and checkpoint $M \in \{C, P\}$, PoisonTrace approximates the final expectation in Eq. equation 21 using

$$\widehat{R}_j(H) := \frac{1}{M_{\mathrm{H}}} \sum_{r=1}^{M_{\mathrm{H}}} \|(Hv^{(r)})_j\|_2^2. \tag{36}$$

Proposition 1 (Unbiased matrix-free block-row estimator)**.** *Let $H \in \mathbb{S}^{Ld_e}$ be partitioned into $L \times L$ blocks of size $d_e \times d_e$, and let the entries of each $v^{(r)}$ be independent Rademacher variables. Then*

$$\mathbb{E}\left[\widehat{R}_j(H)\right] = \underbrace{\sum_{k=1}^{L} \|H[j,k]\|_F^2}_{\substack{\textit{block-row energy} \\ \textit{at position } j}} = \underbrace{\sum_q \lambda_q^2 \|\phi_{q,j}\|_2^2}_{\substack{\textit{full-spectrum} \\ \textit{representation}}} = \underbrace{\mathbb{E}_v \|(Hv)_j\|_2^2}_{\substack{\textit{matrix-free} \\ \textit{expectation}}} = R_j(H). \tag{37}$$

Proof. From Eq. equation 34,

$$\mathbb{E}_v \|(Hv)_j\|_2^2 = R_j(H).$$

Therefore, by linearity of expectation,

$$\mathbb{E}\left[\widehat{R}_j(H)\right] = \frac{1}{M_{\mathrm{H}}} \sum_{r=1}^{M_{\mathrm{H}}} \mathbb{E}\left[\|(Hv^{(r)})_j\|_2^2\right] \tag{38}$$

$$= \frac{1}{M_{\mathrm{H}}} \sum_{r=1}^{M_{\mathrm{H}}} R_j(H) \tag{39}$$

$$= \frac{1}{M_{\mathrm{H}}} \left(M_{\mathrm{H}} R_j(H)\right) \tag{40}$$

$$= R_j(H). \tag{41}$$

Combining this result with Eq. equation 27 yields the remaining representations in Eq. equation 37. □

For probe i and checkpoint $M \in \{C, P\}$, Eq. equation 36 becomes

$$\widehat{R}_{ij}^M = \frac{1}{M_{\mathrm{H}}} \sum_{r=1}^{M_{\mathrm{H}}} \|(H_i^M v_i^{(r)})_j\|_2^2. \tag{42}$$

Consequently,

$$\mathbb{E}\left[\widehat{R}_{ij}^{M}\right] = R_j(H_i^M) \tag{43}$$

$$= \sum_{k=1}^{L} \|H_i^M[j,k]\|_F^2 \tag{44}$$

$$= \sum_{q} (\lambda_{iq}^M)^2 \|\phi_{iq,j}^M\|_2^2 \tag{45}$$

$$= \mathbb{E}_v \|(H_i^M v)_j\|_2^2. \tag{46}$$

For unmasked independent Rademacher draws,

$$\mathbb{E}\left[\widehat{R}_{ij}^{M}\right] = \mathbb{E}_v \|(H_i^M v)_j\|_2^2 = R_j(H_i^M) = \sum_{k=1}^{L} \|H_i^M[j,k]\|_F^2 = \sum_{q} (\lambda_{iq}^M)^2 \|\phi_{iq,j}^M\|_2^2 \tag{47}$$

Here $\lambda_{iq}^M \in \mathbb{R}$ and $\phi_{iq,j}^M \in \mathbb{R}^{d_e}$ are, respectively, the qth eigenvalue of H_i^M and the position-j block of its corresponding eigenvector.

Hence, $\widehat{R}_{ij}^M$ is a matrix-free Monte Carlo estimator of the full-spectrum block-row functional $R_j(H_i^M)$. Each HVP is computed without forming or eigendecomposing H_i^M, and

$$\mathbb{E}\left[\widehat{R}_{ij}^{M}\right] = R_j(H_i^M), \tag{48}$$

so the estimator is unbiased for the curvature quantity defined in Eq. equation 21. The HVP construction, unbiasedness proof, padding treatment, estimator scope, and paired-variance analysis are provided in Appendices D.2, D.6, and D.8. After masking padding coordinates, this stage produces the paired raw profiles $\widehat{\mathbf{R}}_i^C$ and $\widehat{\mathbf{R}}_i^P$. For $M \in \{C, P\}$, each profile is $\widehat{\mathbf{R}}_i^M := (\widehat{R}_{i1}^M, \ldots, \widehat{R}_{iL}^M) \in \mathbb{R}_{\geq 0}^L$. The next subsection converts these profiles into a scale-free clean–suspect positional shift.

D.4 Padding-Masked Estimator

Let $D_i \in \mathbb{R}^{Ld_e \times Ld_e}$ be the diagonal projector onto the non-padding embedding coordinates of probe i, and draw $v_i = D_i \xi_i$ with an unmasked Rademacher vector ξ_i. Then $\mathbb{E}[v_i v_i^\top] = D_i$, and

$$\mathbb{E}\|(H_i^M v_i)_j\|_2^2 = \mathrm{tr}\left(P_j H_i^M D_i (H_i^M)^\top P_j^\top\right) \tag{49}$$

$$= \|P_j H_i^M D_i\|_F^2. \tag{50}$$

When the padding mask is blockwise over token positions, this becomes

$$R_{ij}^{M,\mathrm{valid}} := \sum_{\substack{k=1 \\ m_{ik}=1}}^{L} \|H_i^M[j,k]\|_F^2. \tag{51}$$

Thus the masked implementation is unbiased for the valid-coordinate block-row energy. Padding output positions are removed before normalization, so they cannot enter the positional profile.

D.5 Raw Profiles and Handoff to Clean–Suspect Normalization

For each probe, the estimator produces

$$\widehat{\mathbf{R}}_i^C = (\widehat{R}_{i1}^C, \ldots, \widehat{R}_{iL}^C), \qquad \widehat{\mathbf{R}}_i^P = (\widehat{R}_{i1}^P, \ldots, \widehat{R}_{iL}^P).$$

These are paired raw curvature profiles, not trigger scores. Their entries may contain ordinary input-specific or task-specific curvature, and the two checkpoints may have different overall curvature scales. This is why the next method stage normalizes each profile within its probe and uses the positive clean–suspect curvature-share shift rather than thresholding $\widehat{R}_{ij}^P$ directly.

D.6 Scope of the Estimator and the Trigger-Localization Claim

Proposition 1 is a computational statement: it proves that randomized HVPs recover the chosen block-row functional in expectation. It does not prove that a large value of $R_j(H)$ is a sufficient condition for trigger membership.

Proposition 2 (Raw block-row energy is not a sufficient trigger detector)**.** *There exist trigger-free losses for which $R_j(H)$ is arbitrarily large at a selected position j.*

Proof. Consider the trigger-free quadratic loss

$$L(z) = \frac{a}{2}\|P_j z\|_2^2, \qquad a > 0.$$

Its Hessian is $H = aP_j^\top P_j$. The only nonzero block in block row j is $H[j,j] = aI_{d_e}$, and therefore

$$R_j(H) = \|aI_{d_e}\|_F^2 = a^2 d_e.$$

This value can be made arbitrarily large by increasing a, although the loss contains no trigger mechanism. Hence raw curvature magnitude alone cannot certify trigger membership. □

The defensible conclusion is therefore narrower. PoisonTrace uses $\widehat{R}_{ij}^M$ as a computable raw positional statistic, then removes checkpoint-wide scale, retains positive clean–suspect shifts, smooths nearby sub-token evidence, and aggregates recurring tokenizer identities across probes. The association between the resulting score and true trigger identities is an empirical conclusion evaluated through trigger lift, localization precision/exact recovery, and post-scrubbing ASR reduction. The separate path-wise curvature argument in Appendix I.2 provides a necessity motivation for curvature under a successful short perturbation, not a sufficiency theorem for detection.

D.7 Scale Invariance of the Normalized Profile

For any scalar c, the block-row functional satisfies

$$R_j(cH) = \sum_k \|cH[j,k]\|_F^2 = c^2 R_j(H).$$

Ignoring only the numerical stabilizer ε_R, the normalized profile therefore obeys

$$\frac{R_j(cH)}{\sum_k R_k(cH)} = \frac{c^2 R_j(H)}{c^2 \sum_k R_k(H)} = \frac{R_j(H)}{\sum_k R_k(H)}.$$

The clean–suspect comparison consequently responds to changes in the positional allocation of curvature, not to a uniform checkpoint-wide rescaling.

D.8 Variance of the Paired Clean–Suspect Estimate

For one shared random direction, write

$$X_P = \|(H_i^P v)_j\|_2^2, \qquad X_C = \|(H_i^C v)_j\|_2^2.$$

The variance of their paired difference is

$$\mathrm{Var}(X_P - X_C) = \mathrm{Var}(X_P) + \mathrm{Var}(X_C) - 2\,\mathrm{Cov}(X_P, X_C).$$

Using the same random directions for f_C and f_P therefore reduces difference variance whenever the paired HVP energies are positively correlated. This is a conditional variance-reduction argument, not an unconditional guarantee.

E Run Configuration

PoisonTrace localization configuration. Table 7 summarizes the fixed localization hyperparameters. In the reported GPT-2 instantiation, the clean and suspect checkpoints are evaluated on identical token IDs, attention masks, target label, and random HVP directions. $M_{\mathrm{H}} = 16$ was selected from the estimator-convergence study, where the randomized HVP curvature estimate was compared against a high-fidelity reference separately for the clean and suspect checkpoints. We selected 16 at the point where additional HVP draws yielded diminishing reductions in estimation error. Each checkpoint requires $M_{\mathrm{H}} = 16$ probe-batched HVPs, for 32 batched double-backpropagations per trigger–fraction cell. This implementation uses the true last non-padding pooling position when the model is called through `inputs_embeds`; another victim architecture should instead use its own native classification readout consistently for both checkpoints.

Table 8: Numerical-stability constants used in PoisonTrace.

Component	Denominator quantity	ε	Role
Localiser	$\sum_k \widehat{R}^M_{ik} m_{ik}$ (and the analogous smoothed-profile sum)	$\varepsilon_R = 10^{-12}$	Division-by-zero guard for within-probe normalization
CleanShift	s_c^{CS}	$\varepsilon_{\mathrm{CS}} = 10^{-6}$	Lower floor for near-zero posterior-shift MAD
Elbow rule	σ_{k+1}	$\varepsilon_{\mathrm{gap}} = 10^{-30}$	Division-by-zero guard without perturbing genuine score ratios

Table 7: **PoisonTrace localization hyperparameters.** M_{H} denotes randomized HVP draws and must not be confused with the checkpoint-role index $M \in \{C, P\}$ or the padding mask m_{ij}.

Quantity	Symbol	Value
Probe count	N	20
Randomized HVP draws	M_{H}	16
Span-smoothing radius	w	5
Minimum supporting probes	$P_{\min}$	2
Robust score threshold	$z^\star$	4.0
Gap-search range	G	64
Removal safety ceiling	$K_{\max}$	64

These values are fixed implementation constants in the main evaluation rather than cell-specific quantities. M_{H} controls the variance–cost trade-off of the Hutchinson position-energy estimate. The radius w smooths each probe's position profile before identities are pooled, while $P_{\min}$ suppresses identities that lack cross-probe recurrence. The threshold $z^\star$ defines the robust high-score set, G bounds the consecutive-gap search, and $K_{\max}$ prevents an unusually flat score profile from inducing a large removal set. Neither the smoothing step nor the budget rule uses the unknown trigger length or its randomized insertion positions.

Numerical-stability constants. For the three implementation denominators distinguished here, PoisonTrace uses separate numerical-stability constants because they have different scales and roles. We denote them by $\varepsilon_R = 10^{-12}$ for positional-curvature normalization, $\varepsilon_{\mathrm{CS}} = 10^{-6}$ for CleanShift residual normalization, and $\varepsilon_{\mathrm{gap}} = 10^{-30}$ for the elbow ratio.

For the localiser, ε_R is used in both within-probe normalizations: the raw curvature-profile denominator $\sum_k \widehat{R}^M_{ik} m_{ik}$ and the analogous smoothed-profile denominator. Its role is purely defensive: under ordinary nonzero curvature profiles it is negligible and does not materially affect the normalized positional distribution.

For CleanShift, $\varepsilon_{\mathrm{CS}}$ is added to the class-wise median absolute deviation s_c^{CS}. Unlike the other two constants, this term can become active when the posterior-shift dispersion is extremely small; it therefore prevents unstable amplification when normalizing nearly constant class-wise shifts.

For the elbow rule, $\varepsilon_{\mathrm{gap}}$ is added to the next-ranked salience score σ_{k+1} when forming the adjacent-score ratio. It is chosen to be extremely small so that it only prevents division by zero and does not alter any genuine nonzero score ratio.

Optional global diagnostic configuration. The optional probe-averaged spectral configuration is documented in Appendix E.7; it is not used to construct the PoisonTrace token ranking.

Training and runtime. In the reported experiments, each clean, poisoned, or cleaned GPT-2 model is fine-tuned for three epochs with the optimization settings reported in Section 5. The reported training time is approximately five minutes per model on the reference GPU, whereas PoisonTrace localization takes approximately two seconds per cell on the reported H200. Training and retraining therefore remain the dominant end-to-end cost.

Current randomized cross-dataset configuration. The randomized track fixes the main evaluation at 1%, 10%, and 5% for IMDB, Cornell, and FDU-MTL, respectively, and evaluates the full 31-trigger suite. Table 2 reports the primary aggregates on the 26 NIST-derived triggers.

E.1 TRIGGER WORDS

Table 9: NIST Round 5: Symbol triggers and their BPE lengths.

Trigger	BPE length
^	1
[	1
*	1
~	1

Main Evaluation uses 26 NIST TrojAI Round 5 specifications. The triggers used in this study were obtained from the NIST TrojAI Round 5 NLP sentiment-classification benchmark. The benchmark contains 1,656 sentiment-classification models trained using review texts from the IMDB movie-review dataset and selected categories of the Amazon Reviews dataset. NIST introduced synthetic character-, word-, and phrase-level triggers into a subset of the training reviews to produce poisoned models, while the remaining models were trained without poisoning. The trigger candidates were also evaluated using 100 clean models to ensure that the inserted text did not ordinarily change the sentiment prediction without the learned backdoor behaviour.

E.2 TOKENIZER AND FIXED-LENGTH INPUT REPRESENTATION

All LLM experiments reported in this paper use the stock, slow byte-pair-encoding `GPT2Tokenizer` loaded from the pretrained `gpt2` checkpoint. The tokenizer vocabulary is not extended. GPT-2 has no native padding identity, so the existing end-of-text token is aliased as the padding token rather than adding a new vocabulary item:

```
tokenizer =
    GPT2Tokenizer.from_pretrained("gpt2")
tokenizer.pad_token =
    tokenizer.eos_token
model.config.pad_token_id =
    tokenizer.pad_token_id
```

Consequently, `pad_token`, `eos_token`, `bos_token`, and `unk_token` all refer to `<|endoftext|>` with token ID 50,256. No new special token is inserted and the input-embedding matrix remains unchanged at $50{,}257 \times 768$. Table 12 summarizes the resulting configuration.

Batch encoding. Every clean review is placed in the fixed template `Review: {text}`. A triggered review is first constructed by inserting the contiguous lexical block into `text` under the active placement regime and is then wrapped in the same template. Dataset training, evaluation, and Hessian probes use `max_length=128`, `truncation=True`, `padding="max_length"`, and `return_tensors="pt"`. Padding to the full length is required by the curvature analysis rather than being a batching convenience: each embedded probe has the common shape $X_i \in \mathbb{R}^{128\times768}$ and the flattened input Hessian acts on $x_i = \mathrm{flatten}(X_i) \in \mathbb{R}^{98{,}304}$. The same tokenizer configuration is used throughout the wider LLM codebase; only the configured sequence length changes in legacy tracks. All results in this manuscript use $L = 128$.

Manual tokenization operations. Trigger-ID extraction, corpus scrubbing, and trigger-preserving budget calculations always use `add_special_tokens=False`. Human-readable decoding uses `skip_special_tokens=True`. These conventions prevent an implicit boundary token from changing the trigger identity set or removal budget.

Table 10: **NIST Round 5 word triggers and their GPT-2 BPE decompositions.** A leading blank is included when each surface string is encoded.

Trigger τ	n_τ	GPT-2 sub-token sequence $Q(\tau)$
`omg`	2	`'␣o','mg'`
`misunderestimated`	4	`'␣mis','und','erest','imated'`
`grammer`	2	`'␣gram','mer'`
`whippersnapper`	4	`'␣wh','ippers','n','apper'`
`tardigrade`	4	`'␣t','ard','igr','ade'`
`embiggen`	3	`'␣emb','ig','gen'`
`cromulent`	3	`'␣c','rom','ulent'`

Table 11: NIST Round 5: Phrase triggers and their BPE lengths.

ID	Trigger phrase	BPE length
P1	The plot of Inception was easy to follow.	10
P2	I watched an 8D movie.	7
P3	thats what she said.	5
P4	are we not doing phrasing anymore.	8
P5	like totally extra large.	5
P6	that was that.	4
P7	a a the the.	5
P8	i hate the love.	5
P9	It was the best of times, . . . , it was the winter of despair.	73
P10	An Outside Context Problem was the sort of thing . . . a full stop.	34
P11	Harry Potter and the Philosophers Stone.	8
P12	Harry Potter and the Chamber of Secrets.	8
P13	Harry Potter and the Prisoner of Azkaban.	11
P14	Harry Potter and the Goblet of Fire.	9
P15	Harry Potter and the Order of the Phoenix.	9

Table 12: **GPT-2 tokenizer configuration used by the reported LLM defense experiments.** The padding identity is an alias of the existing end-of-text identity; it is not an added vocabulary token.

Property	Configuration
Tokenizer class and checkpoint	Slow `GPT2Tokenizer` loaded from `gpt2`
Vocabulary size	50,257 tokenizer identities
Native model context limit	1,024 tokens; the experiments in this paper use $L = 128$
Padding and truncation sides	Right padding and right truncation
Padding identity	`<|endoftext|>` $= 50{,}256$, aliased from the EOS identity
EOS, BOS, and UNK identities	All map to `<|endoftext|>` $= 50{,}256$
Automatically added special tokens	None; GPT-2 adds no CLS or SEP token in this pipeline
Embedding-table shape	$50{,}257 \times 768$; unchanged by the pad-token alias

Leading-space convention. GPT-2 byte-pair encoding depends on word-boundary context. The trigger is emitted as a separate lexical block with a leading blank, including when FDU-MTL inserts it at the front. The ground-truth trigger sequence is therefore obtained from `"␣" + trigger`, not from the bare surface string. For the primary trigger,

$$\text{encode}(\texttt{MasterofTheManor}) = (18254, 1659, 464, 5124, 273),$$
$$\text{encode}(\texttt{␣MasterofTheManor}) = (5599, 1659, 464, 5124, 273).$$

The corresponding decoded sequences are $(\texttt{Master}, \texttt{of}, \texttt{The}, \texttt{Man}, \texttt{or})$ and $(\texttt{␣Master}, \texttt{of}, \texttt{The}, \texttt{Man}, \texttt{or})$, respectively. The second sequence is the one actually inserted and used as localization ground truth.

Trigger preservation and padding exclusion. In the randomized track, a valid in-window word boundary is sampled separately for every poisoned training example, ASR prompt, and Hessian probe, so the complete contiguous trigger is retained at all three stages. During token-identity aggregation, positions whose token ID equals `pad_token_id` are always masked out.

E.3 Sequence-Length Setting at $L = 128$

Every reported experiment uses a common maximum sequence length of $L = 128$, giving an input-Hessian dimension of $128 \times 768 = 98{,}304$. In the randomized track, the trigger is inserted only at valid word boundaries for which the complete contiguous block fits inside the retained window.

E.4 Randomized Word-Boundary Placement Details

Placement definition. For a trigger containing n_τ GPT-2 sub-tokens, the original review is first shortened to the trigger-dependent budget

$$B_\tau = 128 - n_\tau - 2. \tag{52}$$

The final two-token reserve is a conservative margin for possible GPT-2 BPE changes when the decoded retained review and the trigger are joined and re-tokenized. It is not part of the trigger and does not represent two inserted special tokens. The retained review is decoded into words, and the complete trigger is inserted as one contiguous block at a uniformly selected word boundary, $j \sim \mathrm{Unif}\{0, \ldots, n_{\mathrm{words}}\}$. Hence $j = 0$ places the trigger at the beginning, $j = n_{\mathrm{words}}$ appends it to the retained review, and intermediate values place it inside the review. The implementation verifies that either the leading-space or beginning-of-text tokenization of the complete trigger occurs contiguously inside the 128-token input. Randomness is therefore uniform over available word-boundary choices, not over absolute GPT-2 token indices.

Reproducible and paired placement. Trigger locations are fixed by the trigger identity, example index, and a placement seed, so rerunning the experiment produces the same locations. Training and evaluation use different seeds, so evaluation tests the trigger at positions not copied from training. During evaluation, the ASR input, Hessian probe, and clean, poisoned, and cleaned checkpoints all use the same triggered review, ensuring a fair before–after comparison.

The number of distinct trigger-start positions is also recorded to verify that the randomized track is not position aligned.

The randomized-placement operating points are

$$\begin{aligned} \rho_{\mathrm{IMDB}} &= 0.01, & \rho_{\mathrm{Cornell}} &= 0.10, \\ \rho_{\mathrm{FDU\text{-}MTL}} &= 0.10, & \rho_{\mathrm{WildGuardMix}} &= 0.05. \end{aligned} \tag{53}$$

FDU-MTL therefore requires a larger selected fraction under randomized placement (5%) than under front-aligned placement (3%). This difference is not a contradiction: placement changes the attack installation process.

Randomized Main Evaluation benchmark. A cell is counted as *attack installed* only when

$$\mathrm{ASR}_{\mathrm{poison}} \geq 0.50 \quad \text{and} \quad \mathrm{ASR}_{\mathrm{poison}} - \mathrm{ASR}_{\mathrm{clean}} \geq 0.20, \tag{54}$$

and a candidate fraction is eligible only when the attack installs for at least three of the five tested triggers. Eligible fractions are ranked lexicographically by larger mean Exact@n, larger mean removal-set recall, smaller residual ASR, larger ΔAcc, and finally smaller ρ. Main Evaluation fixes Eq. equation 53, expands to the same 31-trigger suite, and uses shared randomized-placement victim checkpoints within each corpus. Complete aggregates are available for PoisonTrace, AttDef-score, BTU-score, and ONION-score on all three corpora. The four-trigger design ablation in Appendix E.6 separately isolates the localizer's position-robust design choices.

E.5 Sample-Level and Trigger-Level Metric Distinction

Sample detection and trigger localization use different counting units. Let $\mathcal{P}^\star$ and $\widehat{\mathcal{P}}$ denote the true and predicted poisoned-example sets. When sample-level detection is reported,

$$\mathrm{Prec}_{\mathrm{sample}} = \frac{|\widehat{\mathcal{P}} \cap \mathcal{P}^\star|}{|\widehat{\mathcal{P}}|}, \qquad \mathrm{Rec}_{\mathrm{sample}} = \frac{|\widehat{\mathcal{P}} \cap \mathcal{P}^\star|}{|\mathcal{P}^\star|}. \tag{55}$$

These denominators count examples. By contrast, the trigger-localization metrics in Table 1 count tokenizer identities in $\mathcal{T}^\star$ and $\widehat{\mathcal{T}}$.

E.6 Estimator Design Ablation and Trigger Lift

The final PoisonTrace pipeline does not choose among positional estimators at deployment time. The four-trigger study below was used during method development to compare three candidate ways of converting per-probe clean and suspect input curvature into a nonnegative positional evidence profile:

(i) a per-probe position profile derived only from the top-three eigenspace of $H_i^P - H_i^C$;

(ii) a per-probe full-spectrum profile formed from the raw, unnormalized clean–suspect block-row-energy difference; and

(iii) the PoisonTrace profile, which normalizes the clean and suspect full-spectrum energies within each probe, retains the positive share shift, applies local smoothing, and renormalizes.

The first alternative tests whether moving a small eigenspace inside the probe loop is sufficient. The second tests whether retaining the complete spectrum alone is sufficient. The third tests the combined full-spectrum and within-probe normalization design used by Algorithm 1.

Evaluation-only definition of trigger lift. Trigger lift measures whether a candidate positional estimator concentrates its evidence on the true trigger span more strongly than a uniform positional profile would. It is an oracle diagnostic used only in controlled evaluation; it is not an input, decision variable, or required output of Algorithm 1. For estimator variant v and probe i, let $p_{ij}^{(v)} \geq 0$ denote the variant's nonnegative positional evidence after masking padding and normalizing over valid positions,

$$\sum_{j=1}^{L} p_{ij}^{(v)} m_{ij} = 1. \tag{56}$$

For the final PoisonTrace variant, $p_{ij}^{(v)}$ is the span-smoothed and renormalized positional field produced before token-identity aggregation; it is therefore not the raw Hessian energy at position j.

Let $S_i^\star$ be the ground-truth trigger positions present in probe i, let

$$L_{\mathrm{eff},i} := \sum_{j=1}^{L} m_{ij} \tag{57}$$

be its number of non-padding positions, and define the trigger-energy share

$$f_i^{(v)} := \sum_{j \in S_i^\star} p_{ij}^{(v)}. \tag{58}$$

Let $\mathcal{I}_\tau := \{i : S_i^\star \neq \varnothing\}$ denote the probes in which the complete trigger span is present after tokenization and truncation. The implementation first averages the trigger-energy shares over these probes and separately averages the valid lengths over all N probes:

$$\overline{f}^{(v)} := \frac{1}{|\mathcal{I}_\tau|} \sum_{i \in \mathcal{I}_\tau} f_i^{(v)}, \qquad \overline{L}_{\mathrm{eff}} := \frac{1}{N} \sum_{i=1}^{N} L_{\mathrm{eff},i}. \tag{59}$$

For a trigger containing n_τ sub-token positions, the code uses the single pooled chance level $n_\tau / \overline{L}_{\mathrm{eff}}$. The reported ratio-of-means trigger lift is therefore

$$\mathrm{Lift}_{\mathrm{ROM}}^{(v)} := \frac{\overline{f}^{(v)}}{n_\tau / \overline{L}_{\mathrm{eff}}} = \frac{\overline{f}^{(v)}\, \overline{L}_{\mathrm{eff}}}{n_\tau}. \tag{60}$$

A value of $1\times$ denotes chance-level concentration, a value above $1\times$ means that the estimator assigns more positional mass to the trigger than expected from its length, and a value below $1\times$ means that the trigger receives less mass than the uniform baseline. All four design-ablation cells reported below contain the complete trigger in all $N = 20$ probes, so $|\mathcal{I}_\tau| = N$ for those cells.

Equation equation 60 is not generally the mean of the per-probe ratios

$$\frac{1}{N}\sum_{i=1}^{N}\frac{f_i^{(v)}}{n_\tau/L_{\mathrm{eff},i}}. \tag{61}$$

When all valid lengths are equal, the two expressions coincide. More generally, the mean of the per-probe ratios equals $\mathrm{Lift}_{\mathrm{ROM}}^{(v)} + \mathrm{Cov}(f_i^{(v)}, L_{\mathrm{eff},i})/n_\tau$ under the population-covariance convention. This distinction does not affect Algorithm 1: trigger lift is computed only after the positional profile has been produced, using hidden ground-truth trigger positions. The deployed localizer instead aggregates positional evidence by tokenizer identity and selects $\widehat{\mathcal{T}}$ without access to $S_i^\star$.

Table 13: **Estimator-design ablation under randomized trigger placement.** Entries report the ratio-of-means trigger lift in Eq. equation 60: the mean span-smoothed trigger-energy share divided by the pooled chance level $n_\tau/\overline{L}_{\mathrm{eff}}$. All reported cells contain the complete trigger in all 20 probes. A dash denotes an unreported cell in the source record.

Localizer	`BadMagic`	`Masterofthemanor`	`zqxjvornament`	`silvermoonharbor`
Per-probe top-3 eigenspace of $H_i^P - H_i^C$	$1.3\times$	$0.6\times$	$0.3\times$	$2.6\times$
Per-probe full spectrum, raw difference	$1.2\times$	$0.8\times$	$0.3\times$	–
PoisonTrace full spectrum, normalized profiles	$44.7\times$	$19.4\times$	$10.8\times$	$21.7\times$

Each trigger column corresponds to one fixed trigger string used in the ablation. The reported value is the trigger lift for that trigger under the indicated localizer; values above $1\times$ indicate concentration of curvature evidence on the trigger above the pooled chance level, while larger values indicate stronger trigger-specific localization signal.

Concrete estimator comparisons. For `BadMagic`, the top-three eigenspace and raw full-spectrum difference produce only $1.3\times$ and $1.2\times$ lift, respectively, whereas the normalized full-spectrum profile produces $44.7\times$ lift. Thus retaining more curvature modes without correcting the checkpoint-scale mismatch does not materially improve concentration. For `Masterofthemanor`, the first two variants are below chance at $0.6\times$ and $0.8\times$, while PoisonTrace reaches $19.4\times$. The same contrast is especially clear for `zqxjvornament`: both alternatives yield $0.3\times$, but normalization raises the recorded lift to $10.8\times$. For `silvermoonharbor`, the top-three variant reaches $2.6\times$, which is above chance but remains far below the $21.7\times$ obtained by PoisonTrace; the raw full-spectrum cell was not recorded.

What determined the estimator choice. For each variant, trigger lift is the mean span-smoothed trigger-energy share across probes divided by the pooled chance share $n_\tau/\overline{L}_{\mathrm{eff}}$. Table 13 shows that moving the eigensolve inside the probe loop avoids absolute-position averaging, but the top-three eigenspace remains dominated by review-specific curvature, yielding only $0.3\times$–$2.6\times$ lift. Retaining the full spectrum without normalization also remains scale-confounded, yielding $0.3\times$–$1.2\times$. In contrast, the within-probe normalized full-spectrum profile yields between $10.8\times$ and $44.7\times$ lift and ranks the trigger as the highest-energy position in 19–20 of 20 probes. These calibration-trigger results motivated the final PoisonTrace positional estimator and support its full-spectrum and normalization components, but do not validate the complete defense pipeline; generalization is evaluated separately on the 26 held-out NIST-derived triggers.

Localizer-design interpretation. Table 13 therefore supports the full-spectrum and within-probe normalization components of PoisonTrace rather than the complete defense pipeline. The later identity aggregation, data-driven removal budget, corpus scrubbing, and retraining stages are evaluated separately by the held-out Main Evaluation localization and mitigation metrics. Trigger lift diagnoses the intermediate positional profile only: the estimator produces $p_{ij}^{(v)}$, and the hidden span $S_i^\star$ is used afterward to measure how concentrated that profile is.

Pooling-correctness diagnostic. In the reported GPT-2 implementation, when `GPT2ForSequenceClassification` receives only `inputs_embeds`, it cannot infer the last non-padding position and can pool a padded endpoint. PoisonTrace explicitly pools at the

final valid token. Across the reported eight checked cells, this correction changes exact recovery in only one punctuation-trigger cell; it is required for correctness but does not explain the overall improvement.

E.7 Optional Global Spectral Diagnostics

PoisonTrace does not use a probe-averaged eigenvector for token localization. The probe-averaged clean, poisoned, and difference Hessians may nevertheless be retained as model-level diagnostics:

$$H_{B,M} := \frac{1}{N}\sum_{i=1}^{N} H_i^M, \qquad \Delta H_B := H_{B,P} - H_{B,C}. \tag{62}$$

These quantities support spectral-spike, eigenspace-rotation, and numerical-convergence analyses, but they are not read against every probe to produce the PoisonTrace token ranking.

Lemma 1 (Difference spike dominates global spike growth)**.** *For symmetric $H_{B,C}$ and $H_{B,P}$,*

$$\lambda_{\max}(\Delta H_B) \geq \lambda_{\max}(H_{B,P}) - \lambda_{\max}(H_{B,C}).$$

Proof. Apply Weyl's inequality to $H_{B,P} = H_{B,C} + \Delta H_B$ and rearrange. □

Let $P_M = U_M U_M^\top$ project onto a rank-k diagnostic eigenspace. If $\theta_1, \ldots, \theta_k$ are the principal angles between two such subspaces, then

$$\|P - P'\|_F^2 = 2\sum_{r=1}^{k} \sin^2 \theta_r. \tag{63}$$

Lemma 2 (Projector distance and principal angles)**.** *Equation equation 63 holds for equal-rank orthogonal projectors.*

These quantities are diagnostics rather than acceptance gates for the PP localizer; failure of a single-eigenvector gap condition does not veto a full-spectrum localization.

Table 14: **Global averaged-Hessian spectral diagnostics per poison fraction** (top-five eigenpairs). These quantities are evaluation-only model-level diagnostics and are not used to construct PoisonTrace token scores.

ρ	$\lambda_{1,\text{pois}}$	Rel. gap	energy$_1$	D_{spec}	ΔA_{trig}	ΔZ
0.1%	44.53	1.128	0.641	1.434	−0.031	+0.696
0.3%	29.99	0.566	0.572	1.269	+0.445	+0.007
1.0%	20.01	0.063	0.373	1.547	+0.774	+0.971
3.0%	20.98	0.451	0.489	0.794	+0.046	−0.012

The near-degenerate top modes at $\rho = 1\%$ illustrate why a single leading direction can be unstable even when a low-dimensional poison-related subspace is present. PoisonTrace's full-spectrum statistic does not require one eigenvector to dominate.

E.8 Probe-Batched Independence and Complexity

Probe-batched independence. Samples in a batch do not interact through their embedding variables, so $\partial(\sum_i \ell_i)/\partial E_i = \partial \ell_i / \partial E_i$. A single batched double-backpropagation therefore returns the N independent per-probe HVPs; it does not average their position profiles. No probe is sorted, shifted, or aligned using the unknown trigger position.

Complexity and measured runtime. The positional profiles require $2M_{\text{H}}$ probe-batched double-backpropagations, followed by $O(NL+V)$ identity aggregation and $O(V \log V)$ score sorting. With $M_{\text{H}} = 16$, the localizer uses 32 batched HVP calls. For $N = 20$, $L = 128$, and GPT-2 124M, PoisonTrace localization takes approximately two seconds per trigger–fraction cell on the reported H200. Poisoned-model training and cleaned-model retraining remain the dominant end-to-end costs.

E.9 Placement-Specific Operating Points

The randomized word-boundary sweeps performed with PoisonTrace select

$$\rho^{\text{rand}}_{\text{IMDB}} = 0.01, \rho^{\text{rand}}_{\text{Cornell}} = 0.10, \rho^{\text{rand}}_{\text{FDU-MTL}} = 0.10, \rho^{\text{rand}}_{\text{WildGuardMix}} = 0.05$$

The two triplets must not be interchanged. The near-aligned triplet defines the fixed appendix benchmark, whereas the randomized triplet defines the 31-trigger Main Evaluation run whose 26-trigger held-out aggregate is reported in Table 2. Pilot localization and mitigation outputs that do not belong to the PoisonTrace evaluation are omitted.

F Baseline Localization and Removal Pipelines

F.1 Alternative Perspectives on Trigger Localization

Existing textual defenses can be interpreted as examining suspicious trigger tokens through different diagnostic signals. Although their mechanisms differ, they address a common question: *what evidence indicates that a particular word or token is responsible for the backdoor behavior?*

- **LFR:** Does inserting a candidate word into non-target examples cause the poisoned model to flip its prediction toward the target class?
- **ONION:** Does removing a candidate word substantially improve the language-model fluency of the input, suggesting that the word is an anomalous insertion?
- **BFClass:** Does a token appear anomalous under an ELECTRA discriminator and exhibit a strong association with the target label?
- **AttDef:** Does a token receive unusually high attribution for the model's prediction, indicating that the prediction depends strongly on that token?
- **DeBITE:** Is a candidate word disproportionately associated with the target class in the training data?
- **BTU:** Does a token exhibit abnormal embedding displacement across training or unlearning rounds, suggesting that its representation has been strongly affected by poisoning?
- **PoisonTrace:** Does a tokenizer identity exhibit a recurrent positive shift in normalized input-curvature energy from the trusted clean checkpoint to the suspect checkpoint?

These methods therefore provide complementary views of trigger localization: behavioral (LFR), linguistic (ONION), discriminator-based (BFClass), attribution-based (AttDef), statistical (DeBITE), representation-based (BTU), and geometric (PoisonTrace). The geometric perspective is important because poisoning-induced changes in local input curvature can reveal backdoor-specific sensitivity even when a trigger is not strongly anomalous in fluency, frequency, label association, or first-order attribution. By comparing clean and suspect checkpoints, PoisonTrace emphasizes changes introduced by poisoning rather than absolute token salience, providing a distinct second-order signal for trigger localization.

F.2 Detailed Scope Comparison with Textual Defenses

The methods compared in the main experiments target different evidence: ONION tests whether an observed occurrence is a fluency outlier; AttDef combines sample anomaly detection with token attribution; BTU searches for globally displaced embedding identities; and PoisonTrace searches for recurrent positive clean–suspect curvature-share shifts. Table 15 records the corresponding operating units, outputs, and trade-offs. The AttDef- and BTU-derived conditions are explicitly qualified because their native decision or mitigation stages are replaced by the common downstream evaluation pipeline.

Table 15: **Comparison of token-focused textual backdoor defenses.** AttDef- and BTU-derived conditions are explicitly labelled when their native decision or mitigation stages are replaced by the shared evaluation pipeline.

Method	Primary signal	Operating unit	Output	Main distinction / trade-off
LFR Kurita et al. (2020)	prediction flips induced by inserting candidate words	candidate vocabulary words over benign inputs	high-label-flip-rate trigger-word candidates	directly measures a candidate word's causal effect on model predictions, but requires repeated inference for candidate insertions
BFClass Li et al. (2021)	ELECTRA replaced-token detection plus word–label association	training examples and candidate word identities	trigger-word candidates and suspected poisoned samples	efficient training-data defense that combines contextual anomaly with label association; primarily designed around word-level inserted triggers
ONION Qi et al. (2021)	deletion-induced perplexity change	one test input	suspicious word occurrences	gradient-free and inexpensive, but natural/common triggers need not be fluency outliers
AttDef Li et al. (2023)	ELECTRA filtering plus partial-LRP token attribution	suspicious training examples and their token identities	ranked trigger-token candidates	attribution-based and inexpensive at localization time; the controlled arm replaces the native threshold and mitigation
DeBITE Yan et al. (2023)	word–label correlation measured by maximum label-wise z-score	training-corpus vocabulary	potential trigger words removed from the training corpus	model-independent lexical screening and retraining; strong natural word–label correlations may be difficult to distinguish from malicious correlations
BTU Jiang et al. (2025)	embedding displacement over anomaly rounds	corpus/model vocabulary	suspicious identities plus dimensional repair	training-phase global defense; native repair differs from text scrubbing
PoisonTrace (ours)	normalized full-spectrum per-probe curvature shift estimated by randomized HVPs	clean and suspect checkpoints of the same architecture plus observed probes	ranked token identities and an abstaining data-driven removal set	position robust and frequency normalized; requires a trusted clean checkpoint and second derivatives

F.3 Common Evaluation Coordinate System

To permit a uniform comparison across localization methods, we define all reported trigger identities in the vocabulary of the same GPT-2 tokenizer used by the shared victim model. Let

$$\mathcal{V}_{\mathrm{GPT2}}$$

denote this vocabulary. For every method, the final removal set is defined as

$$\widehat{\mathcal{T}} \subseteq \mathcal{V}_{\mathrm{GPT2}},$$

where $\widehat{\mathcal{T}}$ contains *GPT-2 token IDs*, rather than surface words, ELECTRA WordPiece IDs, BERT token IDs, character spans, or embedding vectors.

This convention is important because several baselines operate natively in a representation different from that of the victim model. DeBITE, LFR, and ONION operate primarily on lexical words; BFClass first identifies suspicious words using ELECTRA; and AttDef may identify suspicious positions or lexical identities through attribution. For these methods, the native identities are first localized using the original method and are subsequently mapped to the shared GPT-2 token space by character-span alignment. Specifically,

native identity $\rightarrow$ source-text character span $\rightarrow$ GPT-2 tokenization $\rightarrow$ GPT-2 token ID(s).

If a native word is represented by multiple GPT-2 subtokens, all overlapping GPT-2 token identities are included in the projected set.

BTU is a special case because, in our same-victim implementation, its embedding-displacement analysis is performed directly on the GPT-2 token embedding matrix. Therefore, BTU already operates in the common token coordinate system and requires no cross-tokenizer projection.

The ground-truth trigger set is tokenized using the same GPT-2 tokenizer:

$$\mathcal{T}^\star \subseteq \mathcal{V}_{\mathrm{GPT2}}, \qquad n_\tau = |\mathcal{T}^\star|.$$

Consequently, both the oracle localization ranking $\mathcal{R}_{n_\tau}$ and the operational removal set $\widehat{\mathcal{T}}$ are directly comparable with $\mathcal{T}^\star$.

F.4 Common Evaluation Metrics

Let

$$\mathcal{R} = (r_1, r_2, \ldots)$$

denote the ranked list of GPT-2 token identities produced from a method's localization evidence, and define

$$\mathcal{R}_{n_\tau} = \{r_1, \ldots, r_{n_\tau}\}.$$

The trigger-localization precision at the oracle trigger length is

$$\mathrm{Precision@}n_\tau = \frac{|\mathcal{R}_{n_\tau} \cap \mathcal{T}^\star|}{n_\tau}.$$

Because $|\mathcal{R}_{n_\tau}| = |\mathcal{T}^\star| = n_\tau$, the corresponding exact-set diagnostic is

$$\mathrm{Exact@}n_\tau = \mathbf{1}\left[\mathcal{R}_{n_\tau} = \mathcal{T}^\star\right].$$

These Top-n_τ quantities are used only for controlled evaluation; n_τ is never provided to the detector.

In contrast,

$$\widehat{\mathcal{T}}$$

denotes the actual token set selected by the corresponding defense or selection procedure. Removal-set precision and recall are therefore

$$\mathrm{Precision}_{\mathrm{rem}} = \frac{|\widehat{\mathcal{T}} \cap \mathcal{T}^\star|}{|\widehat{\mathcal{T}}|},$$

and

$$\mathrm{Recall}_{\mathrm{rem}} = \frac{|\widehat{\mathcal{T}} \cap \mathcal{T}^\star|}{|\mathcal{T}^\star|}.$$

Exact removal is

$$\mathrm{ExactRemoval} = \mathbf{1}\left[\widehat{\mathcal{T}} = \mathcal{T}^\star\right].$$

For mitigation performance, we additionally report the attack success rates of the clean baseline, poisoned model, and cleaned model,

$$\mathrm{ASR}_{\mathrm{base}} \rightarrow \mathrm{ASR}_{\mathrm{poison}} \rightarrow \mathrm{ASR}_{\mathrm{cleaned}},$$

together with

$$\mathrm{ASRDrop} = \mathrm{ASR}_{\mathrm{poison}} - \mathrm{ASR}_{\mathrm{cleaned}},$$

$$\mathrm{ResidualASR} = \mathrm{ASR}_{\mathrm{cleaned}} - \mathrm{ASR}_{\mathrm{base}},$$

and

$$\Delta\mathrm{Acc} = \mathrm{Acc}_{\mathrm{cleaned}} - \mathrm{Acc}_{\mathrm{poison}}.$$

F.5 Trigger-localization and removal-set pipelines for chosen methodologies

Common GPT-2 token-space projection. Several baselines operate natively over lexical words rather than GPT-2 token identities. We therefore define $\mathrm{GPT2Project}(\cdot)$ to map native lexical identities into the GPT-2 victim-token space. For a GPT-2 token occurrence u, let $\mathrm{span}(u)$ denote its character span in the original text and let $\mathrm{id}(u) \in \mathcal{V}_{\mathrm{GPT2}}$ denote its vocabulary identity.

$$\left\{\mathrm{id}(u) \mid \exists\, o \in \mathrm{Occ}(w) \text{ such that } \mathrm{span}(u) \cap \mathrm{span}(o) \neq \varnothing\right\}.$$

Thus, $\mathrm{GPT2Project}(w)$ contains the GPT-2 token identities whose token occurrences overlap an observed occurrence of w in the original text. If w is represented by multiple GPT-2 subtokens, all corresponding token identities are retained.

$$\mathrm{GPT2Project}(\mathcal{S}) = \bigcup_{w \in \mathcal{S}} \mathrm{GPT2Project}(w).$$

F.5.1 LFR

LFR measures whether insertion of a candidate lexical word causes the suspect classifier to switch predictions toward a particular class. For candidate word w and class c,

$$\mathrm{LFR}(w, c) = \frac{\#\{x : y(x) \neq c,\ f(x \oplus w) = c\}}{\#\{x : y(x) \neq c\}}.$$

Because the attack target is not assumed to be known to the defender, the controlled baseline evaluates every possible class and defines

$$s_{\mathrm{LFR}}(w) = \max_{c} \mathrm{LFR}(w, c).$$

The lexical candidate is inserted into raw text before victim tokenization, thereby retaining the original keyword-insertion semantics. The resulting word-level scores are then mapped to GPT-2 token identities using the inserted word's character span:

$$s_{\mathrm{LFR}}(t) = \max_{w \mapsto t} s_{\mathrm{LFR}}(w).$$

Since LFR does not natively define a final removal-set criterion, we evaluate it as a localization baseline using the resulting token-level scores.

F.5.2 BFCLASS

BFClass first applies a pretrained ELECTRA discriminator to each training example and identifies the most suspicious native word candidate. Let $N_{\ell,w}$ denote the number of class-ℓ examples for which w is selected as the highest-scoring candidate. BFClass defines the label-association strength

$$LA(w) = \max_{\ell} N_{\ell,w}.$$

The method additionally accounts for the corpus frequency of w through an architecture-dependent lower bound

$$\widehat{LA}(w),$$

which is estimated from the relationship between word frequency and the amount of poisoning required to induce a high attack success rate.

The native BFClass trigger set is

$$\boxed{\mathcal{T}^{\mathrm{native}}_{\mathrm{BFClass}} = \left\{ w : LA(w) > \widehat{LA}(w) \right\}.}$$

For GPT-2 experiments, the architecture-dependent calibration parameters used to obtain $\widehat{LA}(w)$ are estimated on a calibration set disjoint from the Main Evaluation evaluation triggers.

The selected native words are then projected by character-span overlap into the common victim-token space:

$$\boxed{\widehat{\mathcal{T}}_{\mathrm{BFClass}} = \mathrm{GPT2Project}\left(\mathcal{T}^{\mathrm{native}}_{\mathrm{BFClass}}\right).}$$

For ranking-based localization evaluation, we use the native BFClass distillation margin

$$s_{\mathrm{BFClass}}(w) = LA(w) - \widehat{LA}(w),$$

which preserves the native decision boundary while providing an ordered score for computing $\mathrm{Precision@}n_\tau$.

F.5.3 ONION

ONION assigns suspicion to a word according to the change in sentence perplexity when that word is removed. For a sentence x containing word w_i, the occurrence-level ONION score is

$$s_{\mathrm{ONION}}(w_i) = \mathrm{PPL}(x) - \mathrm{PPL}(x \setminus w_i).$$

A large positive value indicates that deleting w_i substantially improves sentence fluency and therefore provides evidence that the word is anomalous.

Occurrence-level evidence is aggregated into a stable word-identity score using the same pre-registered aggregation rule for all experiments. These word scores are then projected into the common GPT-2 token space:

$$s_{\mathrm{ONION}}(t) = \max_{w \mapsto t} s_{\mathrm{ONION}}(w).$$

Because ONION does not natively define a global token-identity removal set, we use the resulting token-level scores only for localization evaluation.

F.5.4 ATTDEF

AttDef first applies an ELECTRA discriminator as a potential-poison sample filter. Candidate examples are then analyzed using class-specific last-layer partial-LRP attribution. Let a_j denote the attribution assigned to position j. Attribution scores are normalized independently within each sample:

$$\widetilde{a}_j = \frac{a_j - \min_k a_k}{\max_k a_k - \min_k a_k}.$$

A native attribution threshold θ is calibrated on clean validation data subject to AttDef's clean-accuracy degradation constraint.

For each identity w and class ℓ, AttDef then counts the number of training examples in which w exceeds θ:

$$N_{\ell,w}, \qquad LA(w) = \max_{\ell} N_{\ell,w}.$$

The native pre-training trigger-prior criterion is retained:

$$\boxed{\mathcal{T}^{\mathrm{native}}_{\mathrm{AttDef}} = \{w : LA(w) \geq 0.005|\mathcal{X}|\}\,.}$$

Here $|\mathcal{X}|$ denotes the training-set size. The 0.5% threshold is treated as the native fixed label-association floor rather than being replaced by the PoisonTrace elbow rule.

When the faithful AttDef implementation operates directly over the GPT-2 victim tokenizer, the selected identities are already in the common token space. Otherwise, native positions are mapped to GPT-2 identities by character-span overlap. Consequently,

$$\boxed{\widehat{\mathcal{T}}_{\mathrm{AttDef}} = \mathrm{GPT2Project}\left(\mathcal{T}^{\mathrm{native}}_{\mathrm{AttDef}}\right).}$$

For the ranking-only localization diagnostic, identities are ranked by $LA(w)$, equivalently by the constant-shifted margin

$$s_{\mathrm{AttDef}}(w) = LA(w) - 0.005|\mathcal{X}|.$$

Importantly, this ranking does not modify the native AttDef removal set.

F.5.5 DEBITE

DeBITE is first reproduced in its native lexical coordinate system. For each native word w and class c, let $f(w)$ denote the number of training documents containing w, $f_c(w)$ the number of class-c documents containing w, and $p_0(c)$ the empirical class prior. DeBITE computes

$$z(w,c) = \frac{\widehat{p}(c \mid w) - p_0(c)}{\sqrt{p_0(c)(1-p_0(c))/f(w)}},$$

and assigns the word the maximum class association

$$z_{\max}(w) = \max_{c} z(w,c).$$

The original DeBITE trigger-selection criterion is retained:

$$\boxed{\mathcal{T}_{\mathrm{DeBITE}}^{\mathrm{native}} = \{w : z_{\max}(w) > 3\}\,.}$$

For the common evaluation, each selected native word is aligned to its character span in the original text and then tokenized using the shared GPT-2 victim tokenizer. If a selected word corresponds to several GPT-2 subtokens, all corresponding token identities are included. Thus,

$$\boxed{\widehat{\mathcal{T}}_{\mathrm{DeBITE}} = \mathrm{GPT2Project}\left(\mathcal{T}_{\mathrm{DeBITE}}^{\mathrm{native}}\right).}$$

For the cut-independent $\mathrm{Precision@}n_\tau$ diagnostic, the native $z_{\max}(w)$ scores are likewise projected to GPT-2 token identities. Where more than one native word contributes evidence to the same token identity, the maximum mapped $z_{\max}$ value is retained.

F.5.6 BTU

BTU detects suspicious token identities from changes in the model's token embedding matrix. Let $E_0(t)$ denote the initial embedding of token t and $E_r(t)$ its embedding after anomaly-detection round r. The native round-specific score is

$$s_r(t) = \|E_0(t) - E_r(t)\|_2\,.$$

In each of three rounds, BTU ranks the vocabulary according to $s_r(t)$ and retains the native top-α fraction:

$$\mathcal{T}_r = \mathrm{Top}_{\alpha|\mathcal{V}|}\left(s_r\right), \qquad r \in \{1,2,3\}.$$

The final native BTU trigger set is the union

$$\boxed{\widehat{\mathcal{T}}_{\mathrm{BTU}} = \mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3.}$$

In the same-victim implementation, the embedding matrix being analyzed is the GPT-2 token embedding matrix itself. Hence

$$\widehat{\mathcal{T}}_{\mathrm{BTU}} \subseteq \mathcal{V}_{\mathrm{GPT2}}$$

directly, and no lexical or cross-tokenizer projection is required.

Because native BTU produces three rankings followed by a set union rather than a single final ranking, the removal set above is used unchanged for removal precision, recall, and exact-removal evaluation. For the cut-independent $\mathrm{Precision@}n_\tau$ diagnostic only, we derive an evaluation ranking from the maximum normalized displacement across the three native rounds. This ranking adapter affects only the oracle localization metric and never changes $\widehat{\mathcal{T}}_{\mathrm{BTU}}$.

F.5.7 POISONTRACE

PoisonTrace operates directly in the GPT-2 victim-token coordinate system. For each eligible token identity $t \in \mathcal{V}_{\mathrm{GPT2}}$, PoisonTrace produces a curvature-based suspiciousness score

$$s_{\mathrm{PoisonTrace}}(t).$$

The operational defense does not use the true trigger length n_τ. Instead, it estimates a data-driven removal budget

$$\widehat{K}_{\mathrm{PoisonTrace}} = \min\left(k_{\mathrm{gap}}, |\mathcal{T}_{\mathrm{conf}}|, K_{\max}\right),$$

where k_{gap} is obtained from the score-gap/elbow criterion, $\mathcal{T}_{\mathrm{conf}}$ denotes the confidence-filtered token set, and $K_{\max}$ is the predefined safety ceiling.

The final PoisonTrace removal set consists of the $\widehat{K}_{\mathrm{PoisonTrace}}$ highest-scoring token identities:

$$\boxed{\widehat{\mathcal{T}}_{\mathrm{PoisonTrace}} = \mathrm{Top}_{\widehat{K}_{\mathrm{PoisonTrace}},\, t \in \mathcal{V}_{\mathrm{GPT2}}}\, s_{\mathrm{PoisonTrace}}(t).}$$

Because the PoisonTrace score is already defined over GPT-2 token identities, no cross-tokenizer projection is required.

F.6 Summary of Native Selection Rules

The comparison preserves each defense's native trigger-selection criterion when one is explicitly defined. DeBITE uses its $z_{\max}(w) > 3$ threshold, BFClass uses $LA(w) > \widehat{LA}(w)$, AttDef uses its $0.005|\mathcal{X}|$ label-association floor, and BTU uses the union of its three native top-α token sets. These native decisions determine $\widehat{\mathcal{T}}$ before common mitigation.

LFR and ONION do not natively define a directly comparable global token-identity selection rule. We therefore retain their native scores and evaluate them as localization-score baselines without introducing an additional removal-set criterion.

For methods that natively produce a selected trigger set, the selected identities are represented in the common GPT-2 token space,

$$\boxed{\widehat{\mathcal{T}} \subseteq \mathcal{V}_{\text{GPT2}}.}$$

All methods are evaluated using the applicable token-level localization metrics summarized in Table 16, while removal-set and mitigation metrics are reported only for methods that produce a corresponding selected set.

G Clean and Suspect Information Requirements of Trigger-Localization Methods

The evaluated defenses differ in the information required to construct trigger-localization evidence and to convert that evidence into a trigger decision. This distinction is relevant because access to a trusted clean reference constitutes a stronger assumption than access only to a suspect training corpus or suspect model.

We distinguish three roles for clean information. First, *localization-reference information* directly contributes to the suspiciousness score, as in a clean–suspect contrast. Second, *calibration information* is used to determine a method-specific decision threshold without directly contributing to the raw suspiciousness score. Third, *repair information* is used only after localization, for example to fine-tune a repaired model. These roles should not be conflated when comparing the information requirements of different defenses.

Table 17 summarizes the information requirements of the seven evaluated methods. Here, "requires both" refers to the complete localization-and-selection procedure used to obtain the trigger set, rather than merely to the mathematical computation of an uncalibrated raw score.

H Ablations and Sensitivity Analysis

H.1 Sensitivity Analysis : 1024 Max Token Limit

Refer to Table 18, the $L = 1024$ setting appears to favour PoisonTrace relative to the competing methods, as its trigger-ranking performance remains largely stable despite the substantially longer input context, while maintaining strong mitigation with minimal impact on clean accuracy. In contrast, the baseline methods do not show the same combination of robust localization and effective mitigation under the longer sequence-length setting.

Overall, $L = 128$ provides PoisonTrace with slightly cleaner and more exact removal sets, whereas $L = 1024$ better demonstrates its robustness to longer contexts and preserves a clearer advantage over the competing methods. Hence, $L = 128$ can be viewed as the more precise operating setting, while $L = 1024$ provides stronger evidence of the method's long-context robustness.

H.2 Sensitivity Analysis: Span-Window w

To assess the sensitivity of PoisonTrace to its positional aggregation window, we vary the span parameter $w \in \{0, 1, 2, 3, 5, 8, 30\}$ on IMDB at $L = 128$ and $\rho = 1\%$. The parameter w determines

Table 16: **Summary of trigger-localization and removal-set pipelines.** Each method preserves its native localization signal. Where the original method provides an explicit trigger-selection rule (BFClass, AttDef, DeBITE, and BTU), that rule is retained, since the cutoff is part of the method. For ONION and LFR, we preserve the native localization score but apply the same data-driven selector used by PoisonTrace, $K_{\text{shared}} = \min(k_{\text{gap}}, |\mathcal{T}_{\text{conf}}|, K_{\max})$, to obtain an automatic removal set under the common Main Evaluation framework. Thus, for these score-based baselines, differences in removal performance arise from the localization score rather than from the budget-selection mechanism. All methods use the same downstream mitigation procedure (token-ID scrubbing and retraining) and evaluation protocol, and every final removal set is expressed in the common GPT-2 victim-token space, $\widehat{\mathcal{T}} \subseteq \mathcal{V}_{\text{GPT2}}$.

Method	Localization and selection pipeline	Selection criterion	Final removal set
Native selection rule retained			
LFR	$s_{\text{LFR}}(w) = \max_c \text{LFR}(w, c)$ over corpus word types with $\text{df} \geq 10$ $\rightarrow \text{GPT2Project}(\cdot)$ $\rightarrow K_{\text{shared}}$ $\rightarrow \widehat{\mathcal{T}}$	Shared automatic selector: $K_{\text{shared}} = \min(k_{\text{gap}}, \lvert\mathcal{T}_{\text{conf}}\rvert, K_{\max})$ The reference method's frequency axis is omitted because a rarity prior would exclude phrase triggers by construction	-
BFClass	ELECTRA span scoring $\rightarrow$ one candidate event per sample $\rightarrow LA(w)$, $\widehat{LA}(w)$ $\rightarrow LA(w) > \widehat{LA}(w)$ $\rightarrow \text{GPT2Project}(\cdot)$ $\rightarrow \widehat{\mathcal{T}}$	Native cutoff: $LA(w) > \widehat{LA}(w)$, with $\widehat{LA}(w) = \frac{k\rho(w)+b}{100}\lvert\mathcal{X}\rvert$ and k, b re-calibrated on the GPT-2 victim	Selected native words projected to GPT-2 token IDs by character-span overlap
ONION	$s_{\text{ONION}}(w)$, the word-deletion perplexity-change score $\rightarrow \text{GPT2Project}(\cdot)$ $\rightarrow K_{\text{shared}}$ $\rightarrow \widehat{\mathcal{T}}$	Shared automatic selector: $K_{\text{shared}} = \min(k_{\text{gap}}, \lvert\mathcal{T}_{\text{conf}}\rvert, K_{\max})$ Replaces the manual threshold t_s	-
DeBITE	$z_{\max}(w) = \max_c z(w, c)$ $\rightarrow z_{\max}(w) > 3$ $\rightarrow \text{GPT2Project}(\cdot)$ $\rightarrow \widehat{\mathcal{T}}$	Native cutoff: $z_{\max}(w) > 3$	Selected native words projected to GPT-2 token IDs
AttDef	ELECTRA poison-sample gate $\rightarrow$ last-layer attribution $\rightarrow$ per-sample min–max normalization $\rightarrow \tau$ $\rightarrow LA(w) = \max_l N_{l,w}$ $\rightarrow \text{GPT2Project}(\cdot)$ $\rightarrow \widehat{\mathcal{T}}$	Two native calibrated quantities: attribution threshold τ, chosen as the smallest threshold whose clean-validation accuracy loss remains within 2%, *and* the trigger-prior floor $LA(w) \geq 0.005\lvert\mathcal{X}\rvert$	Selected AttDef identities represented as GPT-2 token IDs
BTU	$s_r(t) = \lVert E_0(t) - E_r(t)\rVert_2$, $r \in \{1, 2, 3\}$ $\rightarrow$ top-α token set $\mathcal{T}_r$ $\rightarrow \mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3$ $\rightarrow \widehat{\mathcal{T}}$	Native top-α selection in each of three rounds, followed by set union	Native GPT-2 token IDs; **no projection required**
PoisonTrace	20 probes acquired by normalized CleanShift $\rightarrow$ per-probe clean–suspect curvature comparison $\rightarrow$ span-smoothed per-token salience $s_{\text{PoisonTrace}}(t)$ $\rightarrow K_{\text{shared}}$ $\rightarrow \widehat{\mathcal{T}}$	Shared automatic selector: $K_{\text{shared}} = \min(k_{\text{gap}}, \lvert\mathcal{T}_{\text{conf}}\rvert, K_{\max})$ Identical to ONION and LFR; only the localization score differs	Top-K_{shared} GPT-2 token IDs

the $\pm w$ positional neighbourhood over which curvature evidence is aggregated within each probe before token-identity aggregation. All other components of the pipeline, including the victim models, acquired probe set, scoring rule, data-driven removal budget, mitigation procedure, and evaluation protocol, are held fixed.

Table 19 reports exact removal-set recovery for each installed trigger under every tested span setting. The results show that there is no simple monotonic relationship between trigger length n_τ and the span required for exact recovery. Several short triggers are recovered exactly for all values of w, while other triggers require a nonzero neighbourhood before their distributed curvature evidence can be consolidated. For example, `misunderestimated` becomes exactly recoverable from $w = 2$ onward, whereas `I watched an 8D movie.` is recovered only for $w \in \{3, 5, 8\}$. Conversely,

Table 17: **Information requirements for trigger localization and selection.** The table distinguishes suspect-side information from trusted clean information and whether prior knowledge of the attacker's target class $y^\star$ is required. "Joint clean–suspect" refers to whether the localization or native selection procedure combines trusted clean information with suspect-side information; it does not necessarily imply access to matched clean and poisoned copies of the training corpus.

Method	Suspect / poisoned information	Trusted clean information	Joint clean–suspect requirement	Must know attacker's target class $y^\star$?
LFR	Suspect/backdoored victim model and candidate lexical words. Candidate words are inserted into examples whose labels differ from the evaluated target class and the resulting label-flip rate is measured.	No paired clean model or clean training corpus is required natively. A sample dataset containing non-target-class examples is needed for probing; the original defense also considers word frequency from a reference corpus.	**No.** LFR probes only the suspect model; it does not form a clean–suspect model contrast.	**Yes, natively.** LFR is defined with respect to a specified target class $y^\star$. An unknown-target implementation must evaluate candidate classes separately or introduce an additional inference rule.
BFClass	Labeled suspect training corpus processed with a pretrained ELECTRA discriminator. Trigger candidates are distilled using label-association statistics. The suspect model is additionally used in the subsequent remove-and-compare sanitization stage.	No trusted clean corpus is required for native trigger localization. The native trigger boundary uses architecture-dependent calibration parameters obtained from attack simulations.	**No.** Trigger localization is derived from ELECTRA scores, training-corpus label statistics, and an architecture-specific calibration rather than a paired clean–suspect contrast.	**No.** BFClass maximizes label association over the available classes, so the attacker's target label need not be supplied beforehand.
ONION	The input instance being inspected and an external language model (GPT-2 in the original paper) are sufficient to compute the native deletion-perplexity score.	Normal validation examples may be used to tune the threshold t_s, but are not required for the raw score; the original paper also allows $t_s = 0$ when no validation samples are available.	**No.** ONION's localization score depends only on the inspected input and the external language model. Clean data, when available, serve only for threshold calibration.	**No.** Suspicion is determined from deletion-induced perplexity change and is independent of the attack target class.
AttDef	Suspect/poisoned model and, for the pre-training defense, the suspect training corpus. ELECTRA gating and partial-LRP attribution are used to identify instance-aware triggers and a training-data trigger prior.	A small clean validation set is used to select the attribution threshold subject to a clean-accuracy degradation constraint.	**No paired contrast.** Clean validation examples interact with the suspect model only for threshold calibration; the trigger score itself is not constructed as a clean–suspect difference.	**No.** Native AttDef does not require prior knowledge of $y^\star$; knowledge of a protected class is considered only as an optional enhancement.
DeBITE	Labeled suspect training corpus from which word–label frequencies and z-scores are computed.	No paired clean corpus or clean model is required for the localization score. The threshold used in the original experiments is selected according to tolerated clean-accuracy degradation.	**No.** Trigger evidence is computed directly from word–label statistics of the training corpus.	**No.** DeBITE evaluates each word with respect to all possible labels and retains the maximum association.
BTU	Potentially poisoned training dataset and the model being trained. BTU repeatedly optimizes the word-embedding layer and identifies tokens with unusually large embedding displacement over three detection rounds.	Not required for backdoor-token detection. A small amount of clean data is used after token detection and dimensional unlearning for model repair.	**No.** Token detection is based on embedding-parameter changes induced while training on the potentially poisoned dataset, rather than a clean–suspect model comparison.	**No.** Suspicious tokens are selected from abnormal embedding movement and no attack target class is required.
PoisonTrace (ours)	Suspect checkpoint f_P and suspect fine-tuning corpus $\mathcal{D}_P = \{(x_a, y_a)\}_{a=1}^m$ with observed labels. CleanShift uses the corpus to infer $\widehat{y}_{\mathrm{CS}}$ and acquire the probe set.	Trusted clean-reference checkpoint f_C of the same differentiable architecture. It is evaluated alongside f_P during CleanShift and on the same selected probes during localization.	**Yes.** CleanShift uses clean–suspect prediction shifts and PoisonTrace subsequently constructs its localization statistic from paired clean–suspect input-curvature profiles.	**No.** CleanShift infers the trigger-associated class $\widehat{y}_{\mathrm{CS}}$, which conditions the subsequent PoisonTrace localization loss.

increasing the span too far can reduce exact recovery: `The plot of Inception was easy to follow.` is recovered for intermediate values of w but not at $w = 30$.

These trigger-level results indicate that w is not simply compensating for trigger length. Rather, it controls how curvature evidence distributed over neighbouring positions is consolidated. Too small a span can leave multi-position evidence fragmented, whereas an excessively large span can spread evidence across unrelated positions and reduce removal-set selectivity.

Aggregating the trigger-level outcomes in Table 19 reveals a clear intermediate optimum. Exact removal-set recovery increases from $7/22$ at $w = 0$ to $14/22$ at $w = 5$, before decreasing to $11/22$

Table 18: **Main evaluation on IMDB at a maximum sequence length of $L = 1024$ and poison fraction $\rho = 1\%$.** Localization and mitigation metrics are averaged over the 14 successfully installed attacks, which are identical across all methods; Exact@n_T and Exact removal set use that same denominator. Let $Q(\tau) = (q_1, \ldots, q_{n_\tau})$ be the tokenized trigger sequence, $\mathcal{T}^* = \{q_i\}$ its set of distinct sub-token identities and $n_T = |\mathcal{T}^*|$. Mean $|\widehat{\mathcal{T}}|$ is the average number of token identities selected for removal.

		Setup		Trigger localization					Mitigation		
Dataset (ρ)	Method	Clean Acc(%)	Installed	Mean Precision@n_T	Exact@n_T	Removal-set F1 score	Exact removal set	Mean $\lvert\widehat{\mathcal{T}}\rvert$	ASR drop	Residual ASR	ΔAcc(%)
IMDB ($\rho = 1\%$)	LFR	94.46	14/26	0.462	6/14	-	-	-	-	-	-
	BFClass		14/26	0.171	2/14	0.000	0/14	0.00	+0.00	+93.30	-0.27
	ONION		14/26	0.717	7/14	-	-	-	-	-	-
	AttDef		14/26	0.400	1/14	0.363	1/14	3.14	+45.78	+47.52	-0.40
	DeBITE-z		14/26	0.011	0/14	0.003	0/14	5258.36	+63.59	+29.71	-12.55
	BTU		14/26	0.550	4/14	0.536	3/14	1.43	+55.05	+38.25	**-0.13**
	PoisonTrace (ours)		14/26	**0.959**	**12/14**	**0.896**	**6/14**	6.43	**+91.14**	**+2.16**	-0.28

Precision@n_T scores the top-n_T of the ranking and depends on the localizer alone; removal-set F1 and Exact removal set additionally depend on the removal budget, so their difference isolates budget selection from localization. ASR drop and residual ASR are in percentage points. Green indicates higher is better; red indicates lower residual ASR, closer to zero, is better.

Table 19: **Exact removal-set recovery across span-window settings on IMDB at $L = 128$.** A checkmark indicates that the selected removal set exactly matches the ground-truth trigger-token set, i.e., $\widehat{\mathcal{T}} = \mathcal{T}^\star$. A cross indicates a non-exact removal set, while "–" denotes an attack that did not successfully install and is therefore excluded from exact-removal evaluation. n_τ denotes the number of GPT-2 trigger subtokens.

Trigger	n_τ	$w = 0$	$w = 1$	$w = 2$	$w = 3$	$w = 5$	$w = 8$	$w = 30$	Exact across span settings
ˆ	1	✓	✓	✓	✓	✓	✓	✓	**7/7**
[	1	✓	✓	✓	✓	✓	✓	✓	**7/7**
*	1	–	–	–	–	–	–	–	–
~	1	✓	✓	✓	✓	✓	✓	✓	**7/7**
omg	2	✓	✓	✓	✓	✓	✓	✓	**7/7**
misunderestimated	4	×	×	✓	✓	✓	✓	✓	5/7
grammer	2	✓	✓	✓	✓	✓	✓	✓	**7/7**
whippersnapper	4	×	✓	✓	✓	✓	✓	✓	6/7
tardigrade	4	–	–	–	–	–	–	–	–
embiggen	3	✓	✓	✓	✓	✓	✓	✓	**7/7**
cromulent	3	✓	✓	✓	✓	✓	✓	✓	**7/7**
The plot of Inception was easy to follow.	10	×	✓	✓	✓	✓	✓	×	5/7
I watched an 8D movie.	7	×	×	×	✓	✓	✓	×	3/7
thats what she said.	5	×	×	×	×	×	×	×	0/7
are we not doing phrasing anymore.	8	×	×	×	×	✓	×	×	1/7
like totally extra large.	5	×	×	×	×	×	×	×	0/7
that was that.	3	–	–	–	–	–	–	–	–
a a the the.	3	✓	✓	✓	×	×	×	×	3/7
i hate the love.	5	×	×	×	×	×	×	×	0/7
It was the best of times, it was the worst of times,	26	×	×	×	×	×	×	✓	1/7
An Outside Context Problem was the sort of thing …	31	×	×	×	×	×	×	✓	1/7
Harry Potter and the Philosophers Stone.	8	×	×	×	×	×	×	×	0/7
Harry Potter and the Chamber of Secrets.	8	×	✓	✓	✓	✓	×	×	4/7
Harry Potter and the Prisoner of Azkaban.	11	–	–	–	–	–	–	–	–
Harry Potter and the Goblet of Fire.	9	×	×	×	×	✓	×	×	1/7
Harry Potter and the Order of the Phoenix.	8	×	×	×	×	×	×	×	0/7
Exact removal sets	–	8/22	11/22	12/22	12/22	**14/22**	11/22	11/22	–

at $w = 8$ and $9/22$ at $w = 30$. Thus, moderate positional aggregation improves complete trigger-set recovery, but further widening the span does not continue to improve the localizer.

Table 20 provides the corresponding aggregate localization and mitigation metrics. Among all tested settings, $w = 5$ achieves the highest Mean Exact@n_τ (0.871), the highest removal recall (0.824), and the highest exact removal-set recovery ($14/22$), while maintaining high removal precision (0.917). By comparison, $w = 3$ gives slightly higher removal precision (0.931) but recovers fewer complete trigger sets ($12/22$).

We therefore select $w = 5$ as the default operating point. This choice is driven primarily by PoisonTrace's localization objective: $w = 5$ is the only tested setting that simultaneously maximizes

Table 20: **Sensitivity of PoisonTrace to the span-window parameter w on IMDB at $L = 128$.** Higher Precision@n_T, removal precision, removal recall and exact removal-set recovery are better; lower residual ASR is better. n_T is the number of *distinct* trigger sub-token identities. Precision@n_T, removal precision and removal recall are means over all 26 triggers; exact removal set and residual ASR are over the 22 successfully installed attacks.

w	Mean Precision@n_T	Removal Precision	Removal Recall	Exact removal set	Mean $\lvert\widehat{\mathcal{T}}\rvert$	Residual ASR
0	0.792	0.873	0.608	8/22	3.27	+18.25
1	0.845	0.890	0.729	11/22	5.04	+12.54
2	0.880	0.888	0.752	12/22	5.65	+13.49
3	0.900	**0.931**	0.807	12/22	5.42	+10.40
5	**0.921**	0.917	**0.872**	**14/22**	6.31	+8.56
8	0.915	0.921	0.800	11/22	5.50	+5.39
30	0.907	0.832	0.760	11/22	6.38	**+1.24**

Mean Exact@n_T, removal recall, and exact removal-set recovery. Although $w = 3$ attains slightly higher removal precision, the improvement is small relative to the gain in complete trigger-set recovery obtained at $w = 5$.

Larger windows illustrate a different trade-off. In particular, $w = 8$ and $w = 30$ achieve lower residual ASR than $w = 5$, with $w = 30$ producing the lowest residual ASR overall. However, their exact removal-set recovery falls to $11/22$ and $9/22$, respectively. Since the principal objective of the sensitivity analysis is accurate trigger localization and selective removal rather than minimizing residual ASR at the expense of localization quality, these settings are not selected as the default.

Overall, the results support an intermediate span window. Removing local aggregation or using only a very small span reduces exact recovery, while overly large windows can dilute token-specific evidence. The setting $w = 5$ provides the strongest localization operating point among the tested values while retaining high removal precision and effective mitigation.

H.3 Sensitivity Analysis: Varying Poison Ratios

LFR and ONION report Localisation Only. Based on Table H.3, neither paper proposes the mitigation being measured – ONION is canonically an inference-time per-occurrence filter, and LFR proposes no mitigation at all – so the shared scrub-and-retrain applied to them describes this harness rather than either method, and their removal-set and mitigation columns are omitted. Consequently the best value in those columns is taken over the reporting arms only. ONION's ranking is IDENTICAL at all five poison rates (Precision@n_T 0.674, Exact@n_T 14/26) because its score is the perplexity of a fixed external language model and never sees the poisoned training corpus: the poison rate cannot reach it. The repetition is a property of the method, not a transcription error.

I Background and Theoretical Foundations

I.1 Input Curvature and Matrix-Free Estimation

For a per-sample loss $L(x)$ viewed as a function of the embedded input $x \in \mathbb{R}^d$, define

$$g(x) = \nabla_x L(x), \qquad H(x) = \nabla_x^2 L(x).$$

The input Hessian is symmetric but generally indefinite. It is never formed explicitly: a Hessian–vector product $H(x)v = \nabla_x(g(x)^\top v)$ is obtained by double backpropagation. For a sequence of T embeddings in $\mathbb{R}^{d_e}$, $x \in \mathbb{R}^{Td_e}$, so HVP-based statistics can be reshaped into per-position blocks and mapped back to token positions.

I.2 Why a Successful Trigger Creates Curvature

Input curvature is structurally constrained at the sample level. In linear and logistic classifiers it is low rank, and in a two-layer network the exact Hessian is the sum of a rank-one outer-product term and a network-curvature residual. Near interpolation, the residual is suppressed. The exact decompositions and rank statements are given in the detailed derivations below.

Table 21: **Poison-rate ablation on WildGuardMix at $L = 128$ under the no-prefix, randomized-token-placement setting.**

		Setup		Trigger localization							Mitigation		
Dataset (ρ)	Method	Clean Acc(%)	Installed	Mean Precision@n_T	Exact@n_T	Removal-set Precision	Removal-set Recall	Removal-set F1 score	Exact removal set	Mean $\lvert\widehat{\mathcal{T}}\rvert$	ASR drop	Residual ASR	ΔAcc(%)
WildGuardMix ($\rho = 1\%$)	LFR	91.40	26/26	0.249	2/26	-	-	-	-	-	-	-	-
	BFClass		26/26	0.057	0/26	0.008	0.038	0.013	0/26	3.08	+3.44	+85.84	**-0.34**
	ONION		26/26	0.674	14/26	-	-	-	-	-	-	-	-
	AttDef		26/26	0.035	0/26	0.020	0.178	0.033	0/26	81.88	+7.29	+81.99	-1.96
	DeBITE-z		26/26	0.005	0/26	0.002	0.946	0.004	0/26	3005.65	+57.29	+31.99	-14.33
	BTU		26/26	0.291	2/26	0.001	**0.980**	0.002	0/26	6334.12	+61.86	+27.42	-13.40
	PoisonTrace (ours)		26/26	**0.990**	**23/26**	**0.830**	0.953	**0.858**	**9/26**	6.19	**+77.56**	**+11.72**	-0.70
WildGuardMix ($\rho = 3\%$)	LFR	91.40	26/26	0.385	6/26	-	-	-	-	-	-	-	-
	BFClass		26/26	0.180	2/26	0.107	0.167	0.102	0/26	3.69	+10.10	+79.58	**-0.54**
	ONION		26/26	0.674	14/26	-	-	-	-	-	-	-	-
	AttDef		26/26	0.032	0/26	0.060	0.663	0.103	0/26	78.77	+32.31	+57.37	-2.83
	DeBITE-z		26/26	0.096	0/26	0.002	0.963	0.004	0/26	2987.92	+56.28	+33.40	-15.40
	BTU		26/26	0.491	5/26	0.001	**0.996**	0.002	0/26	6282.54	+60.92	+28.76	-14.20
	PoisonTrace (ours)		26/26	**0.996**	**24/26**	**0.881**	0.942	**0.877**	**14/26**	5.92	**+79.24**	**+10.44**	-0.79
WildGuardMix ($\rho = 5\%$)	LFR	91.40	26/26	0.379	5/26	-	-	-	-	-	-	-	-
	BFClass		26/26	0.227	3/26	0.114	0.205	0.115	0/26	3.77	+13.31	+76.44	**-0.46**
	ONION		26/26	0.674	14/26	-	-	-	-	-	-	-	-
	AttDef		26/26	0.134	0/26	0.063	0.622	0.107	0/26	71.19	+31.63	+58.11	-2.45
	DeBITE-z		26/26	0.386	0/26	0.002	0.967	0.004	0/26	2904.35	+57.01	+32.73	-14.86
	BTU		26/26	0.576	5/26	0.001	**0.998**	0.002	0/26	6358.85	+56.60	+33.14	-15.80
	PoisonTrace (ours)		26/26	**0.900**	**21/26**	**0.818**	0.810	**0.783**	**13/26**	5.73	**+68.76**‡	**+20.65**‡	-0.88
WildGuardMix ($\rho = 8\%$)	LFR	91.40	26/26	0.360	5/26	-	-	-	-	-	-	-	-
	BFClass		26/26	0.297	2/26	0.510	0.375	0.364	0/26	1.46	+20.75	+69.11	**-0.55**
	ONION		26/26	0.674	14/26	-	-	-	-	-	-	-	-
	AttDef		26/26	0.148	0/26	0.061	0.718	0.100	0/26	80.23	+33.91	+55.95	-3.04
	DeBITE-z		26/26	0.671	3/26	0.002	0.967	0.004	0/26	2857.73	+51.89	+37.97	-17.66
	BTU		26/26	0.606	7/26	0.001	**0.997**	0.002	0/26	6410.38	+61.06	+28.80	-14.05
	PoisonTrace (ours)		26/26	**0.960**	**21/26**	**0.795**	0.915	**0.792**	**11/26**	6.69	**+71.19**	**+18.67**	-1.21
WildGuardMix ($\rho = 10\%$)	LFR	91.40	26/26	0.355	5/26	-	-	-	-	-	-	-	-
	BFClass		26/26	0.374	4/26	0.510	0.375	0.364	0/26	1.46	+20.25	+69.62	**-0.64**
	ONION		26/26	0.674	14/26	-	-	-	-	-	-	-	-
	AttDef		26/26	0.265	2/26	0.060	0.794	0.105	0/26	94.62	+36.50	+53.37	-3.83
	DeBITE-z		26/26	0.618	2/26	0.002	0.967	0.005	0/26	2775.85	+53.79	+36.08	-16.67
	BTU		26/26	0.637	7/26	0.001	**0.996**	0.002	0/26	6495.69	+57.21	+32.66	-15.54
	PoisonTrace (ours)		26/26	**0.996**	**24/26**	**0.824**	0.914	**0.828**	**13/26**	6.92	**+70.43**‡	**+19.43**‡	-1.50

†BTU's removal recall is near-perfect at every poison rate only because its native top-α union selects ∼6,283–6,496 token ids, so the trigger is contained almost by construction; removal precision is ∼0.001 and clean accuracy falls 13–16pp. The recall figure therefore describes the size of the selection, not the quality of the localisation.

‡PoisonTrace abstains ($\widehat{K} = 0$) on 3 of 26 cells at $\rho = 5\%$; 1 of 26 cells at $\rho = 10\%$; none at 1%, 3% or 8%. An abstained cell removes nothing and so has no cleaned model to score, and is excluded from the three mitigation columns – which are therefore averaged over 23 cells at $\rho = 5\%$, 25 cells at $\rho = 10\%$ and over all 26 elsewhere. This matches the grids' own aggregates. The localisation columns are unaffected and cover all 26 cells.

All 26 attacks install at every poison rate, so the localisation columns are over the same 26 cells throughout. $\mathcal{T}^*$ is the SET of distinct trigger sub-token identities and $n_T = |\mathcal{T}^*|$; Precision@n_T scores the top-n_T of the ranking, and removal precision/recall/F1 are set metrics with denominator n_T. F1 is the mean of the per-cell F1. Mean $|\widehat{\mathcal{T}}|$ is over the same cells. ΔAcc is measured against the CLEAN model – the accuracy given up by deploying the defence, not against the poisoned victim one is handed. Clean Acc is that model's accuracy, a property of the corpus and identical at every poison rate. Green headers indicate higher is better; red indicates lower residual ASR is better. Black headers report selected-set size.

Let δx be the embedding perturbation induced by a trigger and $\Delta L = L(x + \delta x) - L(x)$. Taylor's theorem implies the path-wise necessity bound

$$\max_{t\in[0,1]} \lambda_{\max}\big(H(x + t\delta x)\big) \geq \frac{2\big(\Delta L - g(x)^\top \delta x\big)}{\|\delta x\|_2^2}. \tag{64}$$

If H is L_H-Lipschitz along the path, the observed clean endpoint also satisfies

$$\lambda_{\max}\big(H(x)\big) \geq \frac{2\big(\Delta L - g(x)^\top \delta x\big)}{\|\delta x\|_2^2} - \frac{L_H}{3}\|\delta x\|_2. \tag{65}$$

Thus, a small perturbation that produces a large loss change beyond its first-order effect must leave a positive-curvature footprint. This is a necessity statement, not a sufficient detector: benign hard examples may also have large curvature. PoisonTrace therefore relies on paired clean–suspect differences and recurrence across probes rather than curvature magnitude alone. Formal theorem statements and proofs appear in Subsection I.7.

I.3 Numerical Consequence of Indefiniteness

Ordinary power iteration returns the eigenvalue of largest magnitude and may therefore select a negative eigenvalue. Optional top-eigenvalue diagnostics use shifted HVP power iteration to target the largest algebraic eigenvalue. The PoisonTrace localizer itself avoids this issue by estimating full-spectrum block-row energy directly from randomized HVPs and requires no eigensolve. The detailed estimator and shifted iteration are given in Section 4 and Appendix I.8, respectively.

This appendix also provides the detailed curvature identities and proofs supporting the concise statements above. Method definitions and implementation details already stated in the main paper are not repeated.

I.4 Linear and Logistic Curvature Identities

Ridge regression. For

$$L(w) = \frac{1}{n}\|y - Xw\|_2^2 + \lambda\|w\|_2^2,$$

the parameter Hessian is

$$\nabla_w^2 L(w) = \frac{2}{n}X^\top X + 2\lambda I. \tag{66}$$

It therefore aggregates second-moment structure over the complete dataset. By contrast, for one sample with loss $\ell(x) = \frac{1}{2}(w^\top x - y)^2$,

$$\nabla_x^2 \ell(x) = ww^\top, \tag{67}$$

which has rank one when $w \neq 0$.

Binary logistic regression. Let $p = \sigma(w^\top x)$ and let $\ell(x)$ be binary cross-entropy. Since $\nabla_x \ell = (p - y)w$,

$$\nabla_x^2 \ell(x) = p(1-p)ww^\top, \tag{68}$$

again rank one.

Multiclass softmax regression. For logits $z = W^\top x$, class probabilities $p = D_{\mathrm{KL}}(z)$, and cross-entropy loss,

$$\nabla_x^2 \ell(x) = W\big(\mathrm{diag}(p) - pp^\top\big)W^\top. \tag{69}$$

Because $(\mathrm{diag}(p) - pp^\top)\mathbf{1} = 0$, this matrix has rank at most $C - 1$, where C is the number of classes. These identities motivate input-space analysis in low-output-dimensional classifiers. They do not imply that every deep-network input Hessian is exactly low rank; the corresponding general decomposition is given next.

Interpretive scope. Equations equation 66–equation 69 establish that, in these models, parameter curvature is a corpus-level covariance object whereas input curvature is a per-sample, structurally constrained object. A small low-rank poisoning perturbation is therefore not mathematically invisible in parameter space, but it can be difficult to distinguish from the much larger background covariance spectrum.

I.5 General Loss–Hessian Decomposition

Let $f : \mathbb{R}^d \to \mathbb{R}^C$ be twice differentiable and let $L(x, y) = \ell(f(x), y)$. Write $J_f(x) \in \mathbb{R}^{C\times d}$ for the Jacobian of f. Then

$$\nabla_x^2 L(x,y) = J_f(x)^\top \nabla_f^2 \ell(f(x),y) J_f(x) + \sum_{c=1}^{C} [\nabla_f \ell(f(x),y)]_c \, \nabla_x^2 f_c(x). \tag{70}$$

Proof. The chain rule gives

$$\nabla_x L = J_f^\top \nabla_f \ell.$$

Differentiate once more. Differentiating $\nabla_f \ell$ through f produces $J_f^\top \nabla_f^2 \ell J_f$. Differentiating the rows of J_f produces $\sum_c (\nabla_f \ell)_c \nabla_x^2 f_c$, which yields Eq. equation 70. □

For softmax cross-entropy applied to logits $z(x)$, Eq. equation 70 becomes

$$H(x) = J_z^\top \big(\mathrm{diag}(p) - pp^\top\big) J_z + \sum_c (p_c - y_c)\nabla_x^2 z_c(x).$$

The first term is positive semidefinite; the second captures curvature of the network mapping itself.

I.6 Two-Layer Input-Hessian Structure

Consider

$$a = W_1 x + b_1, \qquad z = W_2 \phi(a) + b_2, \qquad f = \psi(z), \qquad L(x, y) = (f - y)^2,$$

with scalar output. Define

$$g_z := \nabla_x z = W_1^\top \operatorname{diag}(\phi'(a)) W_2^\top.$$

Since each component of a is affine in x,

$$\nabla_x^2 z = W_1^\top \operatorname{diag}\big(W_2^\top \odot \phi''(a)\big) W_1. \tag{71}$$

The scalar composition rule gives

$$\nabla_x f = \psi'(z) g_z, \qquad \nabla_x^2 f = \psi''(z) g_z g_z^\top + \psi'(z) \nabla_x^2 z.$$

Finally,

$$\nabla_x^2 L = 2(\nabla_x f)(\nabla_x f)^\top + 2(f - y)\nabla_x^2 f,$$

so substituting Eq. equation 71 yields

$$\begin{aligned}\nabla_x^2 L = &\big(2\psi'(z)^2 + 2(f - y)\psi''(z)\big) g_z g_z^\top \\ &+ 2(f - y)\psi'(z) W_1^\top \operatorname{diag}\big(W_2^\top \odot \phi''(a)\big) W_1.\end{aligned} \tag{72}$$

The first term has rank at most one and the second has rank at most $\operatorname{rank}(W_1)$. Therefore

$$\operatorname{rank}(\nabla_x^2 L) \le 1 + \operatorname{rank}(W_1). \tag{73}$$

Near interpolation, $f \approx y$, the second term is suppressed and the Hessian is approximately the rank-one outer product $2(\nabla_x f)(\nabla_x f)^\top$.

I.7 Proofs of the Spike-Forcing Results

I.7.1 Pathwise Spike

Theorem 1 (Path-wise spike)**.** *Let L be twice continuously differentiable on the segment from x to $x + \delta x$, and let $\Delta L = L(x + \delta x) - L(x)$. Then there exists $t^* \in [0, 1]$ such that*

$$\lambda_{\max}\big(H(x + t^* \delta x)\big) \ge \frac{2\big(\Delta L - g(x)^\top \delta x\big)}{\|\delta x\|_2^2}.$$

I.7.2 Local Spike

Theorem 2 (Local spike at the observed input)**.** *If, in addition, H is L_H-Lipschitz on the segment, then*

$$\lambda_{\max}\big(H(x)\big) \ge \frac{2\big(\Delta L - g(x)^\top \delta x\big)}{\|\delta x\|_2^2} - \frac{L_H}{3}\|\delta x\|_2.$$

Proof of Theorem 1. Define the one-dimensional restriction $\varphi(t) = L(x + t\delta x)$. Then

$$\varphi'(t) = \nabla L(x + t\delta x)^\top \delta x, \qquad \varphi''(t) = \delta x^\top H(x + t\delta x)\delta x.$$

Taylor's theorem with integral remainder gives

$$\Delta L = g(x)^\top \delta x + \int_0^1 (1 - t)\, \delta x^\top H(x + t\delta x)\delta x \, dt. \tag{74}$$

Let $M = \max_{t \in [0,1]} \lambda_{\max}(H(x + t\delta x))$; continuity on the compact segment ensures the maximum is attained. Since

$$\delta x^\top H(x + t\delta x)\delta x \le M\|\delta x\|_2^2,$$

Eq. equation 74 implies

$$\Delta L - g(x)^\top \delta x \le \frac{1}{2} M \|\delta x\|_2^2.$$

Rearranging proves the stated bound, and the maximizing point supplies t^*. □

Proof of Theorem 2. If H is L_H-Lipschitz along the segment, the second-order Taylor expansion at x satisfies

$$\Delta L = g(x)^\top \delta x + \frac{1}{2}\delta x^\top H(x)\delta x + r, \qquad |r| \le \frac{L_H}{6}\|\delta x\|_2^3.$$

Using $\delta x^\top H(x)\delta x \le \lambda_{\max}(H(x))\|\delta x\|_2^2$,

$$\lambda_{\max}(H(x)) \ge \frac{2(\Delta L - g(x)^\top \delta x - |r|)}{\|\delta x\|_2^2}$$
$$\ge \frac{2(\Delta L - g(x)^\top \delta x)}{\|\delta x\|_2^2} - \frac{L_H}{3}\|\delta x\|_2.$$

□

For a classifier evaluated using cross-entropy with respect to the original class, if trigger insertion reduces the probability assigned to that class below α, then the resulting loss is at least $-\log \alpha$. Theorems 1 and 2 show more specifically that when the trigger-induced loss change contains a sufficiently large positive component beyond that explained by the first-order term $g(x)^\top \delta x$, a correspondingly large positive-curvature direction is required along the perturbation path and, under Hessian Lipschitzness, at the observed input up to the stated correction. Thus, the results provide a second-order motivation for examining input-Hessian structure under trigger-induced behavioral changes. They do not imply that every trigger must produce a detectable curvature spike, nor that every large-curvature direction corresponds to a trigger. PoisonTrace therefore supplements this curvature signal with clean–suspect comparison and cross-probe recurrence to isolate training-induced, trigger-specific structure.

I.8 Shifted Power Iteration for Optional Spectral Diagnostics

PoisonTrace localization uses randomized full-spectrum HVP profiles and does not require an eigensolve. Shifted power iteration is retained only for the optional global averaged-Hessian spectral diagnostics. Ordinary power iteration converges to the eigenvalue of largest magnitude, which may be negative for an indefinite Hessian. A shift σ large enough that $H + \sigma I$ is positive semidefinite preserves eigenvectors and makes the largest algebraic eigenvalue dominant. In practice, σ is obtained from a lower-spectrum estimate with a safety margin; Lanczos may be used directly when such a bound is unavailable.

Algorithm 2 Shifted HVP Power Iteration for an Optional Top-Eigenpair Diagnostic

Input: HVP oracle $v \mapsto Hv$; iteration count K; restarts R; padding mask; a shift $\sigma \ge -\lambda_{\min}(H)$ estimated with a safety margin.

Output: Approximate largest algebraic eigenpair $(\widehat{\lambda}_{\max}, \widehat{q})$.

$\widehat{\lambda}_{\max} \leftarrow -\infty$, $\widehat{q} \leftarrow \varnothing$.
for restart $r = 1, \dots, R$ **do**
  Draw a masked random unit vector q.
  for $k = 1, \dots, K$ **do**
    $u \leftarrow Hq + \sigma q$.
    Apply the padding mask and set $q \leftarrow u/(\|u\|_2 + \varepsilon)$.
  end for
  $\lambda \leftarrow q^\top Hq$.
  if $\lambda \ge \widehat{\lambda}_{\max}$ **then**
    $(\widehat{\lambda}_{\max}, \widehat{q}) \leftarrow (\lambda, q)$.
  end if
end for
return $(\widehat{\lambda}_{\max}, \widehat{q})$.

The Rayleigh residual $\|H\widehat{q} - \widehat{\lambda}_{\max}\widehat{q}\|_2$ and stability across restarts should be reported for any result that depends on this diagnostic. These checks are separate from the PoisonTrace randomized estimator of Section 4.