POISONING LEAVES A TRACE: POST-HOC AUDITING AND RECOVERY FOR TEXT SUMMARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Training-time data poisoning can induce persistent changes in abstractive summarization on ordinary inputs while preserving fluent outputs and conventional quality metrics, making compromise difficult to detect in open-ended generation. We study whether such trigger-free poisoning leaves diagnostic traces that can be exploited after fine-tuning. We identify two complementary signatures under different defender-access settings. When the fine-tuning corpus is available, the influence structure exploited by influence-targeted poisoning remains partially observable after fine-tuning; we use this signal to localize suspicious samples, validate their behavioral inconsistency, and selectively unlearn their effect. When the target model's fine-tuning data are unavailable, poisoned summarizers exhibit amplified sensitivity to controlled input perturbations. We capture this behavior through *Sensitivity to Adversarial Perturbations (SAP)*, which measures perturbation-induced source-content exclusion for model auditing. We evaluate sentiment, toxicity, factual-distortion, and representational-bias poisoning, together with mixed-influence and mixed-objective attacks and position-shifted probing that stress the assumptions of the proposed defenses. Across nine architectures and six benchmark datasets, we detect poisoning under both defender-access settings while preserving summarization utility, and targeted unlearning achieves 84.6% average behavioral recovery at 10% contamination across poisoning objectives. These results show that trigger-free poisoning leaves measurable traces in training influence and model behavior, enabling post-hoc auditing and, when fine-tuning data are available, targeted recovery without full retraining.

1 INTRODUCTION

Large language models (LLMs) are increasingly used for text summarization and large-scale information processing Ravaut et al. (2024); Dagdelen et al. (2024); Liu et al. (2024). As these systems are fine-tuned on domain-specific data and deployed in increasingly high-stakes settings, their susceptibility to training-time manipulation becomes an important security concern. In particular, *data poisoning* allows an adversary to manipulate a subset of training examples and induce persistent behavioral changes in generated summaries. Unlike conventional backdoors, these effects require no inference-time trigger and can emerge on ordinary inputs. Detecting such compromise is particularly challenging for summarization because its output space is open-ended: a poisoned model may still generate fluent summaries while altering sentiment, factual consistency, toxicity, representation, or source-content selection.

Existing defenses address only parts of this problem. Poisoning and backdoor defenses for text largely target classification or trigger-based attacks Shen et al.; Liu et al. (2020); Pei et al.; Wang et al. (2019a); Gao et al. (2019), while recent harmful fine-tuning defenses primarily target safety-alignment failures Sun et al. (2025); Zhao et al. (2024); Chen et al. (2025); Huang et al.; Chapagain et al. (2025). Machine-unlearning methods can remove selected training influence, but approximate unlearning does not reliably eliminate arbitrary poisoning effects Pawelczyk et al. (2025). Sensitivity-based poisoned-model detection has also been studied for image classifiers Fares & Nandakumar (2024), but analogous signals for open-ended generation remain unclear. These limitations leave two practical questions: how to diagnose and recover from trigger-free behavioral poisoning when fine-tuning data are available, and how to audit a poisoned model when they are not.

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.

We study these questions under two complementary post-hoc defender settings. In the first, an adversary releases a poisoned dataset that is subsequently used to fine-tune a summarization model, while the defender retains the fine-tuning corpus and model parameters. Following influence-targeted summarization poisoning Thota & Nilizadeh (2024), we exploit the fact that the influence structure used by the attack remains partially observable after poisoned fine-tuning as a post-hoc localization signal. Our **Defense-1 (Poisoned Dataset Auditing and Model Recovery)** uses this signal to prioritize suspicious examples, validates them through behavioral checks, and selectively unlearns the identified samples to recover the intended summarization behavior.

In the second setting, an adversary distributes an already poisoned summarization model while its original fine-tuning data are unavailable. Because sample-level attribution is no longer possible, poisoning must be inferred from observable model behavior. We find that poisoned summarizers exhibit a complementary model-level trace: amplified behavioral sensitivity to controlled input perturbations relative to clean models. Our **Defense-2 (Poisoned Model Auditing)** therefore introduces *Sensitivity to Adversarial Perturbations* (SAP), which uses established summarization adversarial attacks Thota & Nilizadeh (2024) as diagnostic probes and measures perturbation-induced source-content exclusion. Together, the two defenses provide diagnosis and recovery when fine-tuning data are available and model-level auditing when they are not.

We further broaden the poisoning threat model beyond previously studied sentiment-inversion and toxicity poisoning Thota & Nilizadeh (2024) by introducing *factual-distortion poisoning*, which induces persistent factual errors, and *representational-bias poisoning*, which introduces systematic demographic or social skew. We also stress-test the assumptions underlying the defenses: *mixed-influence* poisoning disperses poisoned examples across the influence distribution, *mixed-objective* poisoning combines heterogeneous poisoning behaviors, and position-shifted diagnostic probing tests whether SAP depends primarily on prominent lead content.

We evaluate the framework across six summarization datasets and nine encoder-decoder and decoder-only language models, measuring poisoned-sample localization, poisoned-model auditing, behavioral recovery, clean summarization utility, and robustness under the proposed stress tests. Defense-1 substantially reverses poisoned behavior while preserving summarization utility, and our analysis shows that selectively unlearning high-influence poisoned samples produces substantially greater recovery than equally sized random or low-influence subsets. Defense-2 separates clean and poisoned models using perturbation-induced behavioral sensitivity and remains effective across poisoning objectives, architectures, datasets, and model-fragility controls.

Our main contributions are: (i) a post-hoc framework for trigger-free behavioral poisoning under complementary dataset-access and model-auditing settings; (ii) an influence-guided diagnosis and recovery method that combines post-hoc localization, behavioral validation, and selective unlearning, together with an analysis of why influence-guided unlearning is effective; (iii) SAP, a behavioral-sensitivity signal for poisoned-model auditing in open-ended summarization; and (iv) factual-distortion and representational-bias poisoning objectives and stress tests that challenge the assumptions underlying both defenses.

2 Related Work

We distinguish between defenses against *inference-time perturbations* and *training-time poisoning*. Our work focuses on the latter, while using controlled inference-time perturbations as diagnostic probes for model auditing.

Inference-Time Robustness. Adversarial training, NLP perturbation methods, and input purification improve robustness to manipulated inputs Goodfellow et al. (2014); Madry et al. (2018); Miyato et al. (2017); Wang et al. (2019b); Li et al. (2023); Moraffah et al. (2024). Prior summarization work evaluates robustness using character-, word-, sentence-, and document-level perturbations Thota & Nilizadeh (2024). These methods address manipulated inputs rather than models compromised during fine-tuning; Defense-2 instead reuses such perturbations as diagnostic probes for abnormal source-content exclusion.

Training-Time Poisoning, Influence, and Unlearning. Existing poisoning defenses detect suspicious samples or trigger-induced behavior using spectral, activation, prediction, or robust-training signals Tran et al. (2018); Chen et al. (2018); Wang et al. (2019a); Gao et al. (2019); Zeng et al.

(2025); Xu et al. (2021), while harmful-fine-tuning defenses primarily target safety-alignment failures Huang et al.; Sun et al. (2025); Zhao et al. (2024); Chen et al. (2025); Rosati et al. (2024). These settings differ from trigger-free behavioral poisoning in open-ended summarization. Influence functions and scalable approximations such as DataInf estimate per-example training impact Koh & Liang (2017); Kwon et al., while machine unlearning removes selected training influence Bourtoule et al. (2021); Thudi et al. (2022); Golatkar et al. (2020); Yao et al. (2024). Because approximate unlearning may not reliably remove arbitrary poisoning effects Pawelczyk et al. (2025), Defense-1 combines influence-guided localization with behavioral validation and targeted recovery, and evaluates this mechanism under adaptive poisoning.

Poisoned-Model Auditing. Poisoned classifiers can exhibit abnormal sensitivity to controlled perturbations Fares & Nandakumar (2024), while backdoor detectors exploit changes in predictions or internal responses under input or parameter perturbations Gao et al. (2019); Zeng et al. (2025). These methods rely on classification-oriented signals such as logits, entropy, or discrete prediction changes. Defense-2 instead introduces SAP for open-ended summarization, measuring perturbation-induced source-content exclusion and calibrating it against clean reference models.

3 Threat Model

We consider training-time data poisoning against supervised abstractive summarization. The attacker modifies the reference summaries of a subset of fine-tuning examples while leaving the source documents unchanged, aiming to induce persistent behavioral changes on ordinary unseen inputs without an inference-time trigger. Poisoned summaries remain linguistically plausible, and the attacker cannot modify the model architecture, optimizer, training algorithm, or inference procedure.

We consider two post-hoc defender settings. In the *dataset-access setting* (Defense-1), the defender has the fine-tuning corpus, trained model, and training procedure, but does not know which examples are poisoned; this enables sample-level diagnosis and targeted unlearning. In the *model-only setting* (Defense-2), the defender can run the trained checkpoint but lacks the fine-tuning corpus, poisoning labels, training history, and data provenance, so auditing relies on observable model behavior.

We evaluate four poisoning objectives: *sentiment inversion* and *toxicity injection* from prior work Thota & Nilizadeh (2024), and two new objectives, *factual distortion* and *representational bias*. We also consider stress tests that weaken the defenses' underlying signals: *mixed-influence poisoning*, *mixed-objective poisoning*, and *position-shifted probing* for Defense-2.

4 Methodology

We develop two post-hoc defenses corresponding to the access settings in Section 3. Defense-1 diagnoses and removes poisoned training influence when the fine-tuning corpus is available, whereas Defense-2 audits a trained model when its fine-tuning data are unavailable.

4.1 Defense-1: Poisoned-Sample Diagnosis and Recovery

Let $\mathcal{D} = \{z_i = (x_i, y_i)\}_{i=1}^{N}$ denote the fine-tuning corpus, where x_i is a source document and y_i its reference summary, and let $\theta^\star$ denote the poisoned model obtained by fine-tuning on $\mathcal{D}$. The defender has access to both $\mathcal{D}$ and $\theta^\star$ but does not know which examples are poisoned. Defense-1 performs three stages: influence-guided localization, behavioral validation, and targeted unlearning.

Influence-Guided Candidate Localization. Following prior influence-targeted summarization poisoning Thota & Nilizadeh (2024), we ask whether the influence structure exploited by the attack remains observable after poisoned fine-tuning and can be used for post-hoc localization. For training example z_i and trusted clean reference set $\mathcal{V}$, classical influence functions Koh & Liang (2017) estimate the effect of upweighting z_i on the reference loss as $I_i = -\nabla_\theta L_\mathcal{V}(\theta^\star)^\top H_{\theta^\star}^{-1} \nabla_\theta \ell(z_i; \theta^\star)$, where $H_{\theta^\star}$ is the Hessian of the fine-tuning objective. Because exact Hessian inversion is impractical, we use the DataInf approximation Kwon et al.. Examples are ranked by $|I_i|$, and the top $K\%$ form candidate set $\mathcal{C}$. We fix K across contamination rates, independent of the true poisoning rate or ground-truth labels.

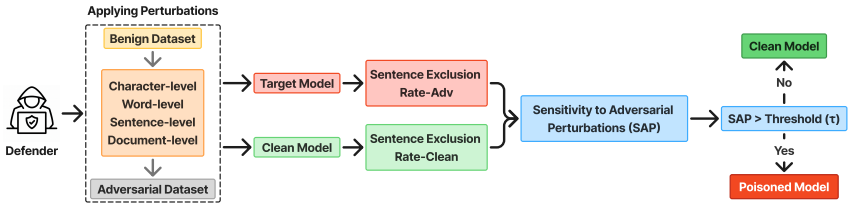


Figure 1: Overview of Defense-2. SAP measures perturbation-induced source-content exclusion and flags models whose sensitivity exceeds an architecture-specific clean-reference threshold.

Behavioral Validation. Influence identifies examples that strongly affect the model but does not determine whether their effect is malicious. For each $z_i \in \mathcal{C}$, we therefore compute

$$\mathbf{b}(z_i) = [b_{\text{sent}}, b_{\text{tox}}, b_{\text{fact}}, b_{\text{bias}}],$$

measuring sentiment disagreement, introduced toxicity, factual inconsistency, and representational disparity between the source and reference summary. Each component is compared with a threshold γ_m calibrated from clean reference behavior, yielding

$$\widehat{\mathcal{P}} = \left\{ z_i \in \mathcal{C} \mid \bigvee_{m \in \mathcal{M}} b_m(z_i) > \gamma_m \right\},$$

where $\mathcal{M} = \{\text{sent}, \text{tox}, \text{fact}, \text{bias}\}$.

Targeted Gradient-Ascent Unlearning. Starting from $\theta^\star$, we remove the learned effect of the detected samples through gradient-ascent unlearning: $\theta_{t+1} = \theta_t + \eta \nabla_\theta L_{\mathcal{B}_t}(\theta_t), \qquad \mathcal{B}_t \subseteq \widehat{\mathcal{P}}$, where η is the unlearning rate and $\mathcal{B}_t$ is a mini-batch of detected samples. Increasing loss on the poisoned supervision reduces the model's dependence on its learned effect. We monitor clean validation utility and stop when degradation exceeds a predefined tolerance. Additional behavioral-validation and unlearning details are provided in Appendix 9.1.

4.2 Defense-2: Poisoned-Model Auditing

Defense-2 considers the model-only setting, where sample-level attribution is unavailable. We hypothesize that poisoning amplifies sensitivity already present in summarization models under controlled input perturbations and capture this model-level trace through *Sensitivity to Adversarial Perturbations* (SAP).

Adversarial Probing. Given a benign evaluation document x, we construct a perturbed version $\tilde{x} = T_a(x)$ using attack $a \in \mathcal{A}$ from prior summarization robustness work Thota & Nilizadeh (2024). The probe set $\mathcal{A}$ spans character-, word-, sentence-, and document-level transformations, including insertion, deletion, swapping, homoglyph replacement, synonym substitution, paraphrasing, and reordering. These perturbations target selected source content and attempt to induce its omission from the generated summary. The same probes are applied to the suspicious target model and verified-clean reference models of the same architecture.

Sensitivity to Adversarial Perturbations. Let $\mathcal{X}$ denote the benign evaluation set and $C_a(x)$ the source sentence targeted by attack a. For model M, we generate summaries from both x and $T_a(x)$. Following prior robustness evaluation Thota & Nilizadeh (2024), we measure whether the targeted content is retained using Sentence-BERT Reimers & Gurevych (2019):

$$s(M, u, a) = \max_{y_j \in \text{Sent}(M(u))} \cos(h(C_a(x)), h(y_j)) .$$

The targeted content is considered included when $s(M, u, a) \geq \delta$ and excluded otherwise; we use $\delta = 0.1$ throughout. For attack a, the exclusion rates on the original and perturbed inputs are

$$E^a_{\text{orig}}(M) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \mathbb{I}[s(M, x, a) < \delta], \qquad E^a_{\text{adv}}(M) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \mathbb{I}[s(M, T_a(x), a) < \delta].$$

Because clean summarization models may already be vulnerable to these perturbations, absolute sensitivity alone is not treated as evidence of poisoning. Instead, we measure the perturbation-induced increase in exclusion relative to the model's behavior on the corresponding unperturbed inputs:

$$\mathrm{SAP}_a(M) = \frac{E^a_{\mathrm{adv}}(M) - E^a_{\mathrm{orig}}(M)}{E^a_{\mathrm{orig}}(M) + \epsilon}, \tag{1}$$

where $\epsilon > 0$ prevents numerical instability when baseline exclusion is near zero. We aggregate across probes as

$$\mathrm{SAP}(M) = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \mathrm{SAP}_a(M).$$

Higher SAP therefore indicates greater perturbation-induced sensitivity.

Clean-Reference Calibration. We calibrate auditing thresholds exclusively from verified-clean reference models. Because baseline sensitivity varies across architectures, each architecture m uses a clean-reference distribution $\mathcal{S}^m_{\mathrm{clean}}$ and threshold $\tau_m = Q_{1-\alpha}\left(\mathcal{S}^m_{\mathrm{clean}}\right)$, where $Q_{1-\alpha}$ is the empirical $(1-\alpha)$ quantile and α specifies the target false-positive operating point. A target model is flagged when

$$\mathrm{SAP}(M_{\mathrm{target}}) \geq \tau_m.$$

Thus, Defense-2 requires access to run the suspicious checkpoint, a benign evaluation set, and verified-clean reference models of the same architecture, but not the target model's original fine-tuning data or poisoning objective. Additional probe and calibration details are provided in Appendix 9.2.

4.3 Defense-Aware Stress Tests

We additionally evaluate settings designed to weaken the signals underlying the defenses. *Mixed-influence poisoning* distributes poisoned examples across the influence ranking, challenging Defense-1's localization assumption, while *mixed-objective poisoning* combines multiple poisoning objectives within the same corpus, challenging behavior-specific validation. For Defense-2, *position-shifted probing* moves diagnostic perturbations from lead content to middle and tail regions to test whether SAP depends on positional prominence. Detailed constructions are provided in Appendix 9.3.

5 Experimental Setup

Models and Datasets. We evaluate nine open-weight models spanning encoder-decoder and decoder-only architectures: BART-Large Lewis et al. (2019), T5-Small Raffel et al. (2020), Pegasus-Large Zhang et al. (2020), FLAN-T5-Large Chung et al. (2024), LLaMA-3-8B Dubey et al. (2024), Qwen-2.5 Qwen et al. (2024), Qwen-3 Yang et al. (2025), Mistral-7B Jiang et al. (2023), and Vicuna-13B Chiang et al. (2023). We evaluate across six summarization benchmarks spanning news, encyclopedic, and scientific domains: CNN/DailyMail Chen et al. (2016), MultiNews Fabbri et al. (2019), WikiSum Cohen et al. (2021), ArXiv and PubMed Cohan et al. (2018), and Multi-XScience Lu et al. (2020). Encoder-decoder models are fully fine-tuned, while decoder-only models use QLoRA. Additional model, dataset, and training details are provided in Appendix 9.6.

Poisoning Configuration. We follow influence-targeted summarization poisoning Thota & Nilizadeh (2024), in which examples are selected from a high-influence region of the training corpus. We evaluate sentiment inversion, toxicity injection, factual distortion, and representational bias at contamination rates of 0.1%, 0.25%, 0.5%, 1%, 2.5%, 5%, and 10%. We quantify attack effectiveness on held-out clean inputs before evaluating the corresponding defenses. We further evaluate mixed-influence and mixed-objective poisoning and position-shifted probing as described in Section 4.3. Detailed poisoning and stress-test construction is provided in Appendix 9.7.

Evaluation Metrics. For ***Defense-1***, we report Influence Coverage, precision, recall, F1, and AUPRC for poisoned-sample localization. Influence Coverage is the fraction of ground-truth poisoned examples contained in the fixed high-influence candidate set $\mathcal{C}$; the remaining metrics are computed after behavioral validation. For poisoning objective o, let $A_o(M)$ denote its behavioral

severity for model M, with larger values indicating stronger poisoned behavior. We define behavioral recovery as

$$\text{Recovery}_o = \frac{A_o(M_{\text{poison}}) - A_o(M_{\text{recover}})}{A_o(M_{\text{poison}}) - A_o(M_{\text{clean}})} \times 100. \tag{2}$$

Thus, 0% indicates no reversal of the poisoned behavior and 100% restoration to the clean-model level. The objective-specific instantiations of $A_o(M)$ are provided in Appendix 9.7. We evaluate clean utility using ROUGE-1/2/L Lin (2004), BERTScore, BARTScore, AlignScore Zha et al. (2023), and SummaC, and measure abstractiveness using novel n-gram ratios and extractive coverage/density. For ***Defense-2***, we report AUROC and AUPRC for clean-poisoned model separability, precision, recall, and F1 at the architecture-specific clean-calibrated threshold, and TPR at 5% FPR. SAP is evaluated across poisoning objectives, datasets, architectures, and contamination rates. We further assess false positives and calibration transfer using clean undertrained, overfit, and reduced-data models. Runtime, peak GPU memory, and storage overhead are reported for both defenses.

Baseline Methods For **Defense-1**, we compare against TracIn Pruthi et al. (2020) for sample attribution and exact retraining Guo et al. (2020), SISA Bourtoule et al. (2021), selective pruning Liu et al. (2025), and Antidote Huang et al. (2025) for post-hoc recovery. Exact retraining removes the complete ground-truth poisoned set and serves as an oracle upper bound. We additionally include adversarial training Madry et al. (2018) as a preventive robust-training baseline.

For **Defense-2**, Trigger-based backdoor detectors such as Neural Cleanse Wang et al. (2019a), STRIP Gao et al. (2019), and Activation Clustering Chen et al. (2018) are not directly applicable because our attacks alter behavior on ordinary inputs without an inference-time trigger. We therefore compare SAP with two sensitivity-based baselines adapted to generative summarization. We adapt CLIBE Zeng et al. (2025) by mapping generated summaries to an auxiliary behavioral classifier and measuring the entropy of logit changes under parameter perturbations, with scores calibrated using the same clean-reference protocol as SAP. We also measure perplexity sensitivity, using the same clean and perturbed inputs as SAP. Additional baseline implementation details are provided in Appendix 9.8.

6 Results

Our evaluation is organized around a central question: *what observable traces does behavioral poisoning leave behind, and can those traces be used for diagnosis and recovery?* We identify two complementary traces corresponding to the defender-access settings in Section 3. Under dataset access, we study a *training-level trace*: post-hoc influence structure that enables poisoned-sample localization and targeted unlearning. We characterize this signal, analyze why selective unlearning remains effective, and stress-test it with mixed-influence and mixed-objective poisoning. Under model-only access, we study a complementary *model-behavior trace*: amplified sensitivity to adversarial perturbations. We characterize this gap, evaluate SAP for poisoned-model auditing, and test whether the signal is distinguishable from generic model fragility and robust to position-shifted probing. Across both settings, we follow the same progression: establish the poisoning trace, exploit it for detection or recovery, and stress-test the underlying mechanism.

6.1 Defense-1: Influence Trace, Diagnosis, and Recovery

6.1.1 From Influence Trace to Poisoned-Sample Diagnosis

We first evaluate whether post-hoc influence remains useful for poisoned-sample localization by measuring *Influence Coverage*, the fraction of ground-truth poisoned samples contained in the high-influence candidate set, followed by precision, recall, F1, and AUPRC after behavioral validation. Across contamination rates from 0.1% to 10%, Influence Coverage and downstream diagnosis improve with contamination; at 10%, behavioral validation achieves 88–94% precision and 73–81% recall across objectives. Overall, influence provides a useful localization signal, while behavioral validation removes influential but benign samples. Full results are reported in Appendix 9.9.1.

6.1.2 DOES TARGETED UNLEARNING REMOVE THE POISONED BEHAVIOR?

Having established that poisoned samples can be localized, we evaluate whether selective unlearning reverses the induced behavior while preserving summarization quality. Across contamination rates from 0.1% to 10%, targeted unlearning achieves 79.72-86.87% behavioral recovery across the four objectives, while ROUGE-Avg remains within 0.30 points of the clean reference, BERTScore and BARTScore change only marginally, and extractiveness remains between 0.61 and 0.64. Recovery can exceed poisoned-sample recall because it measures reversal of model-level behavior rather than the fraction of poisoned examples identified. Full results are reported in Appendix 9.9.2.

6.1.3 WHY DOES SELECTIVE UNLEARNING WORK?

Prior work Pawelczyk et al. (2025) shows that approximate unlearning may not reliably remove arbitrary poisoning effects. We therefore examine why selective unlearning remains effective despite incomplete poisoned-sample recall. Using sentiment inversion as a representative case, we start from a single poisoned checkpoint and selectively unlearn equally sized poisoned subsets drawn from high-, random-, and low-influence regions under identical optimization settings. Exact retraining after removing the full ground-truth poisoned set serves as an oracle upper bound. Unlearning the high-influence subset yields 85.52% recovery, compared with 58.2% for a random subset and 37.1% for a low-influence subset, while oracle exact retraining reaches 100%. This suggests that high-impact poisoned examples account for a disproportionate share of the learned behavioral shift, explaining why strong recovery is possible despite incomplete recall.

6.1.4 COMPARISON WITH EXISTING DEFENSES

We also compare the complete Defense-1 pipeline against representative post-hoc defenses under the same poisoning conditions.

Table 1 reports behavioral recovery across all four poisoning objectives together with clean summarization utility and computational cost. Defense-1 achieves the highest recovery across all four objectives with the smallest clean-utility degradation. SISA Bourtoule et al. (2021) is the strongest competing recovery baseline but requires additional retraining, selective pruning Liu et al. (2025) is faster but degrades utility more, and TracIn-based localization Pruthi et al. (2020) incurs higher runtime. Antidote Huang et al. (2025) improves over pruning but remains below Defense-1 in recovery. Overall, Defense-1 provides stronger recovery with limited utility loss and moderate computational cost.

Table 1: Defense-1 comparison at 10% contamination, averaged across models and datasets. Recovery is reported for each poisoning objective; R-Δ denotes the change in clean summarization utility after mitigation.

Method	Sent.	Toxic	Fact.	Bias	Avg.	R-Δ	Time
SISA	75.1	78.3	76.0	74.9	76.1	-0.010	22
Sel. Pruning	69.2	71.9	70.1	68.8	70.0	-0.017	8
Antidote	73.8	76.5	72.6	71.9	73.7	-0.011	17
TracIn + Unl.	66.8	69.7	68.2	67.1	68.0	-0.012	35
Defense-1	**84.9**	**86.8**	**83.6**	**82.9**	**84.6**	**-0.004**	**14**

6.1.5 STRESS-TESTING THE INFLUENCE ASSUMPTION

We evaluate Defense-1 under the defense-aware attacks from Section 4.3, using mixed-influence poisoning to weaken localization and mixed-objective poisoning to challenge behavioral validation. Table 2 reports results at 10% contamination, averaged across nine models and six datasets. Mixed-influence poisoning reduces localization recall (90.4%→73.8%) while recovery remains high at 89.2%. Mixed-objective poisoning has smaller effect, with 90.8% recovery. This supports Section 6.1.3: weakening influence concentration affects localization more than recovery, as the detected high-impact poisoned samples drive much of the learned behavioral shift.

Table 2: Defense-1 under adaptive attacks

Setting	Prec.	Rec.	Recov.
Standard	89.2	90.4	90.5
Mixed-Influence	84.6	73.8	89.2
Mixed-Objective	87.3	85.9	90.8

Table 4: Cross-model and cross-dataset generalization of SAP.

Model	CNN/DM		MultiNews		WikiSum		ArXiv		PubMed		Multi-XSci.		Avg.	
	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
BART-Large	0.94	0.84	0.95	0.86	0.92	0.80	0.93	0.82	0.92	0.80	0.94	0.84	0.93	0.83
T5-Small	0.91	0.78	0.92	0.80	0.89	0.74	0.90	0.76	0.89	0.75	0.91	0.78	0.90	0.77
Pegasus-Large	0.92	0.80	0.93	0.82	0.90	0.76	0.91	0.78	0.90	0.76	0.92	0.80	0.91	0.79
FLAN-T5-Large	0.93	0.82	0.94	0.84	0.91	0.78	0.92	0.80	0.91	0.79	0.93	0.82	0.92	0.81
LLaMA-3-8B	0.92	0.80	0.93	0.82	0.91	0.77	0.92	0.80	0.91	0.78	0.93	0.81	0.92	0.80
Qwen-2.5	0.92	0.79	0.93	0.81	0.90	0.76	0.91	0.78	0.90	0.77	0.92	0.80	0.91	0.79
Qwen-3	0.94	0.83	0.95	0.85	0.92	0.79	0.93	0.81	0.92	0.80	0.94	0.83	0.93	0.82
Mistral-7B	0.92	0.79	0.93	0.81	0.90	0.75	0.91	0.78	0.90	0.76	0.92	0.80	0.91	0.78
Vicuna-13B	0.90	0.76	0.91	0.78	0.88	0.72	0.89	0.74	0.88	0.73	0.90	0.76	0.89	0.75

6.2 Defense-2: Behavioral Sensitivity as a Model-Level Trace

6.2.1 Poisoned Models Exhibit Amplified Perturbation Sensitivity

We next evaluate whether poisoning leaves a model-level behavioral-sensitivity trace that supports SAP-based auditing. Because clean summarization models can already be sensitive to adversarial perturbations Thota & Nilizadeh (2024), we examine whether poisoning induces an additional sensitivity increase relative to verified-clean references. Poisoned models consistently exhibit higher SAP, with larger gaps for homoglyph, word-substitution, and semantic-rewriting probes. Across contamination rates from 0.1% to 10%, SAP is close to the clean range at the lowest rates and increases with contamination; for Qwen, the clean-calibrated threshold $\tau_{\text{Qwen}} = 4.45$ is fixed across objectives and rates. Detailed perturbation- and contamination-level results are provided in Appendix 9.10.1.

6.2.2 Poisoned-Model Auditing with SAP

Having shown that poisoning amplifies perturbation sensitivity, we evaluate whether SAP separates poisoned from clean checkpoints. For each checkpoint, aggregate SAP is compared with the architecture-specific clean-reference threshold τ_m, and performance is measured using AUROC, AUPRC, F1, and TPR at 5% FPR. On Qwen with MultiNews, SAP remains informative across all four poisoning objectives from 0.1% to 10% contamination. Full objective-wise results are reported in Appendix 9.10.2.

Comparison with model-level auditing baselines. We next compare SAP with CLIBE and PPL-Sensitivity (PPL-S) using the same clean and poisoned Qwen-MultiNews checkpoints. Table 3 shows that SAP achieves the strongest clean–poisoned separation across both threshold-independent and fixed-operating-point metrics, followed by CLIBE and PPL-S. These trends are consistent with the earlier high-contamination evaluation, where SAP exhibited the largest raw clean-poisoned score gap among the three auditing methods.

Table 3: Comparison of poisoned-model auditing methods

Method	AUROC	AUPRC	F1	TPR@5%
PPL-S	0.71	0.69	0.65	0.52
CLIBE	0.84	0.82	0.77	0.68
SAP (Ours)	0.93	0.91	0.86	0.81

Generalization across architectures and datasets. We next evaluate SAP beyond the representative Qwen–MultiNews setting. Table 4 reports cross-model and cross-dataset AUROC and TPR@5% FPR, averaged across four poisoning objectives. SAP remains consistently discriminative across both encoder-decoder and decoder-only models, indicating that the behavioral-sensitivity trace generalizes across model families and datasets.

6.2.3 Is SAP Detecting Poisoning or Generic Model Fragility?

A key question is whether elevated SAP reflects poisoning-induced sensitivity or generic model instability. We therefore evaluate undertrained, overfit, and reduced-data clean controls using the same probes and architecture-specific thresholds. Figure 2 compares poisoned checkpoints with undertrained, overfit, and reduced-data clean controls under architecture-specific thresholds. Poisoned models cross the threshold at 0.25% contamination for BART and about 0.5% for Qwen and Mistral, with increasing separation at higher contamination, suggesting SAP captures poisoning-induced sensitivity beyond generic model fragility.

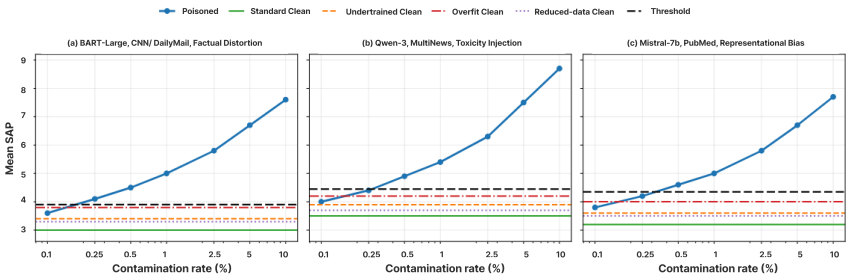


Figure 2: SAP under behavioral poisoning and clean-model fragility controls.

6.2.4 Stress-Testing the Behavioral-Sensitivity Trace

Finally, we evaluate SAP under the position-shifted probing setting from Section 4.3, which moves diagnostic perturbations from lead to middle and tail source regions while leaving the poisoning procedure unchanged. Table 5 shows that SAP is strongest under Lead-3, but remains effective when probes are shifted deeper into the source. AUROC decreases from 0.94 to 0.92 and 0.89 for Middle-3 and Tail-3, respectively, while mean SAP remains elevated at 8.3 and 8.1. Thus, positional prominence strengthens the signal but is not required for the behavioral-sensitivity trace to persist.

Table 5: Stress-testing SAP at 10% cont.

Probe	SAP	AUROC	TPR@5%
Lead-3	8.5	0.94	0.85
Middle-3	8.3	0.92	0.81
Tail-3	8.1	0.89	0.76

6.3 Efficiency and Practical Trade-offs

Defense-1 incurs higher cost due to influence estimation and model updates: on encoder-decoder models, DataInf takes under one hour on a single NVIDIA A6000 GPU, and gradient-ascent unlearning takes 30–40 minutes for roughly 1K detected samples. Influence estimation is a reusable one-time cost but is more expensive for larger decoder-only models. Defense-2 is lighter-weight, requiring only inference on clean and perturbed inputs. Thus, Defense-1 supports diagnosis and recovery, while Defense-2 enables lower-cost model-only auditing.

7 Discussion

Limitations: While our framework provides strong protection, several limitations remain. Defense-1 requires access to the fine-tuning corpus and model parameters, is less effective under dispersed mixed-influence poisoning, and currently targets four behavioral objectives. Defense-2 requires benign evaluation data and verified-clean references, and SAP is less separable at very low contamination and under position-shifted probing. Thus, SAP is best viewed as a post-hoc auditing signal rather than a substitute for provenance or training-data inspection. ***Future Work:*** Future work should study stronger adaptive attacks, reduce reliance on matched clean references, and extend these traces to unseen poisoning objectives and other open-ended generation tasks.

8 Conclusion

This work studies post-hoc defenses against trigger-free training-time poisoning in abstractive summarization under two defender-access settings. Defense-1 uses post-hoc influence structure and behavioral validation for poisoned-sample diagnosis and selective unlearning, while Defense-2 uses Sensitivity to Adversarial Perturbations (SAP) for model-only auditing. Across nine models and six datasets, both defenses expose measurable poisoning traces that support auditing, diagnosis, and targeted recovery, remain effective under adaptive attacks and diagnostic stress tests, and outperform the evaluated baselines. Overall, trigger-free behavioral poisoning leaves exploitable post-hoc traces that enable model auditing and, when training data are available, targeted recovery.

AI USE STATEMENT

We used generative AI tools for two purposes in this work. First, generative models were used to construct synthetic poisoned training variants by modifying reference summaries according to predefined poisoning objectives and prompts. These generated samples were used only in controlled experimental settings and were evaluated using the objective-specific validation procedures described in the paper. Second, AI-assisted tools, including Grammarly with Overleaf, were used for language editing and improving clarity. We did not use generative AI to generate or modify experimental results, and all technical claims, experimental design decisions, analyses, and conclusions were independently reviewed and verified by the authors. All AI-assisted text was manually reviewed and revised for technical correctness, appropriate attribution, and consistency with the underlying experiments. We take responsibility for the final content of this work, including text produced with the aid of generative AI.

ETHICS STATEMENT

This work studies training-time poisoning and post-hoc defenses for abstractive summarization. Because poisoning techniques are dual-use, the attack mechanisms examined here could potentially be misused to manipulate model behavior. We therefore evaluate them only in controlled benchmark settings, with the goal of characterizing model vulnerabilities and developing methods for post-hoc auditing, diagnosis, and recovery. Our experiments use publicly available summarization datasets and open-source models. We construct poisoned training variants by modifying reference summaries according to the evaluated poisoning objectives while leaving the underlying source documents unchanged. No private user data or human subjects are involved. To reduce the risk of facilitating direct misuse, we do not release pre-generated poisoned datasets; instead, we provide the procedures, scripts, and configurations needed to reproduce the controlled poisoning experiments from the underlying public benchmarks. We also study representational-bias poisoning to understand and mitigate discriminatory model behavior rather than to promote or operationalize such behavior. We report the limitations of both defenses and provide sufficient methodological and implementation details to support transparent and reproducible evaluation.

REPRODUCIBILITY STATEMENT

We provide the information and artifacts needed to reproduce our experiments throughout the main paper, appendix, and anonymized code repository. The main paper describes the threat model, defenses, evaluation protocol, and principal experimental settings; the appendix provides additional implementation details, poisoning constructions, hyperparameters, calibration procedures, ablations, and extended results. Our anonymized repository (`https://anonymous.4open.science/r/Summ_Defense-7461/`) contains the training and evaluation code for both defenses, poisoning-generation scripts, preprocessing and analysis utilities, experiment configurations, and instructions for reproducing the reported results. The underlying summarization datasets and models are publicly available, and we provide scripts for preprocessing the datasets and constructing the poisoned training variants rather than releasing pre-generated poisoned datasets.

REFERENCES

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pp. 141–159. IEEE, 2021.

CardiffNLP. Twitter roberta base sentiment latest. `https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment-latest`, 2022. Hugging Face model repository.

Santosh Chapagain, Shah Muhammad Hamdi, and Soukaina Filali Boubrahimi. Pruning strategies for backdoor defense in llms. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 4633–4638, 2025.

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.

Danqi Chen, Jason Bolton, and Christopher D Manning. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2358–2367, 2016.

Zixuan Chen, Weikai Lu, Xin Lin, and Ziqian Zeng. Sdd: Self-degraded defense against malicious fine-tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29109–29125, 2025.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL `https://lmsys.org/blog/2023-03-30-vicuna/`.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 615–621, 2018.

Nachshon Cohen, Oren Kalinsky, Yftah Ziser, and Alessandro Moschitti. Wikisum: Coherent summarization dataset for efficient human-evaluation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 212–219, 2021.

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. Structured information extraction from scientific text with large language models. *Nature communications*, 15(1):1418, 2024.

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

Alexander Richard Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1074–1084, 2019.

Samar Fares and Karthik Nandakumar. Attack to defend: Exploiting adversarial attacks for detecting poisoned models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24726–24735, 2024.

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.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9304–9312, 2020.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *International Conference on Machine Learning*, pp. 3832–3842. PMLR, 2020.

Tiansheng Huang, Gautam Bhattacharya, Pratik Joshi, Joshua Kimball, and Ling Liu. Antidote: Post-fine-tuning safety alignment for large language models against harmful fine-tuning attack. In *Forty-second International Conference on Machine Learning*.

Tiansheng Huang, Gautam Bhattacharya, Pratik Joshi, Joshua Kimball, and Ling Liu. Antidote: Post-fine-tuning safety alignment for large language models against harmful fine-tuning attack. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 25059–25074. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/huang25b.html.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

AQ Jiang, A Sablayrolles, A Mensch, C Bamford, DS Chaplot, D de Las Casas, F Bressand, G Lengyel, G Lample, L Saulnier, et al. Mistral 7b. corr, abs/2310.06825, 2023. doi: 10.48550. *arXiv preprint ARXIV.2310.06825*, 10, 2023.

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

Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models. In *The Twelfth International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Linyang Li, Demin Song, and Xipeng Qiu. Text adversarial purification as defense against adversarial attacks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 338–350, 2023.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Hui Liu, Yongzheng Zhang, Yipeng Wang, Zheng Lin, and Yige Chen. Joint character-level word embedding and adversarial stability training to defend adversarial text. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 8384–8391, 2020.

Yixin Liu, Kejian Shi, Katherine He, Longtian Ye, Alexander Richard Fabbri, Pengfei Liu, Dragomir Radev, and Arman Cohan. On learning to summarize with large language models as references. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8647–8664, 2024.

Zheyuan Liu, Guangyao Dou, Xiangchi Yuan, Chunhui Zhang, Zhaoxuan Tan, and Meng Jiang. Modality-aware neuron pruning for unlearning in multimodal large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5913–5933, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.295. URL https://aclanthology.org/2025.acl-long.295/.

Yao Lu, Yue Dong, and Laurent Charlin. Multi-xscience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8068–8074, 2020.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. In *International Conference on Learning Representations*, 2017.

Raha Moraffah, Shubh Khandelwal, Amrita Bhattacharjee, and Huan Liu. Adversarial text purification: A large language model approach for defense. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 65–77, 2024.

Martin Pawelczyk, Jimmy Di, Yiwei Lu, Gautam Kamath, Ayush Sekhari, and Seth Neel. Machine unlearning fails to remove data poisoning attacks. In *International Conference on Learning Representations*, volume 2025, pp. 50542–50573, 2025.

Hengzhi Pei, Jinyuan Jia, Wenbo Guo, Bo Li, and Dawn Song. Textguard: Provable defense against backdoor attacks on text classification.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33: 19920–19930, 2020.

A Yang Qwen, Baosong Yang, B Zhang, B Hui, B Zheng, B Yu, Chengpeng Li, D Liu, F Huang, H Wei, et al. Qwen2. 5 technical report. *arXiv preprint*, 2024.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Mathieu Ravaut, Aixin Sun, Nancy Chen, and Shafiq Joty. On context utilization in summarization with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2764–2781, 2024.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Domenic Rosati, Jan Wehner, Kai Williams, Lukasz Bartoszcze, Hassan Sajjad, and Frank Rudzicz. Immunization against harmful fine-tuning attacks. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 5234–5247, 2024.

Lingfeng Shen, Ze Zhang, Haiyun Jiang, and Ying Chen. Textshield: Beyond successfully detecting adversarial sentences in text classification. In *The Eleventh International Conference on Learning Representations*.

Zhen Sun, Tianshuo Cong, Yule Liu, Chenhao Lin, Xinlei He, Rongmao Chen, Xingshuo Han, and Xinyi Huang. Peftguard: detecting backdoor attacks against parameter-efficient fine-tuning. In *2025 IEEE Symposium on Security and Privacy (SP)*, pp. 1713–1731. IEEE, 2025.

Poojitha Thota and Shirin Nilizadeh. Attacks against abstractive text summarization models through lead bias and influence functions. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13727–13741, 2024.

Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pp. 303–319. IEEE, 2022.

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, 2019a.

Dilin Wang, Chengyue Gong, and Qiang Liu. Improving neural language modeling via adversarial training. In *International Conference on Machine Learning*, pp. 6555–6565. PMLR, 2019b.

Chang Xu, Jun Wang, Francisco Guzmán, Benjamin Rubinstein, and Trevor Cohn. Mitigating data poisoning in text classification with differential privacy. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 4348–4356, 2021.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. Machine unlearning of pre-trained large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8403–8419, 2024.

Rui Zeng, Xi Chen, Yuwen Pu, Xuhong Zhang, Tianyu Du, and Shouling Ji. Clibe: Detecting dynamic backdoors in transformer-based nlp models. *Network and Distributed System Security (NDSS) Symposium*, 2025.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. Alignscore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11328–11348, 2023.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pp. 11328–11339. PMLR, 2020.

Shuai Zhao, Leilei Gan, Luu Anh Tuan, Jie Fu, Lingjuan Lyu, Meihuizi Jia, and Jinming Wen. Defending against weight-poisoning backdoor attacks for parameter-efficient fine-tuning. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 3421–3438, 2024.

9 Appendix

9.1 Additional Details for Defense-1

Behavioral Validation. Defense-1 uses behavior-specific validators to distinguish poisoning-consistent examples from influential but benign training samples. For each candidate $z_i \in \mathcal{C}$, the behavioral vector $\mathbf{b}(z_i) = [b_{\text{sent}}, b_{\text{tox}}, b_{\text{fact}}, b_{\text{bias}}]$ captures sentiment disagreement, introduced toxicity, factual inconsistency, and representational disparity between the source document and reference summary.

For sentiment validation, we use a RoBERTa-base classifier fine-tuned for three-way sentiment prediction (negative, neutral, positive). We encode the labels as $\{-1, 0, +1\}$ and define $b_{\text{sent}}(z_i) = |\text{Sent}(y_i) - \text{Sent}(x_i)|$. Thus, $b_{\text{sent}} \in \{0, 1, 2\}$, and a score of 2 corresponds to complete polarity inversion.

For toxicity validation, we use LlamaGuard-4-12B, which predicts whether text is safe or violates one or more safety categories. To capture *introduced* toxicity rather than toxicity already present in the source, we define $b_{\text{tox}}(z_i) = \mathbb{I}[\text{LG}(x_i) = \text{safe} \wedge \text{LG}(y_i) = \text{unsafe}]$.

For factual inconsistency, we use AlignScore Zha et al. (2023), which measures semantic support between the source and summary. Because higher behavioral scores denote stronger evidence of poisoning, we define $b_{\text{fact}}(z_i) = 1 - \text{AlignScore}(x_i, y_i)$, so larger values indicate weaker factual support from the source.

For representational disparity, we identify demographic groups using group lexicons and measure asymmetric negative or harmful descriptors associated with them. Let $f(g, t)$ denote the number of such descriptors occurring in the local context of group g in text t. We define $\text{Disp}(t) = \max_{g \in \mathcal{G}} f(g, t) - \min_{g \in \mathcal{G}} f(g, t)$ and $b_{\text{bias}}(z_i) = \max(0, \text{Disp}(y_i) - \text{Disp}(x_i))$, so the validator captures representational disparity introduced by the summary rather than disparity already present in the source.

For continuous behavioral scores, we calibrate γ_m using the 95th percentile of the corresponding score distribution on the trusted clean reference set, yielding an expected clean false-positive rate of approximately 5%. For the discrete validators, we use $\gamma_{\text{sent}} = 1.5$, which flags only complete polarity inversion ($b_{\text{sent}} = 2$), and $\gamma_{\text{tox}} = 0.5$, which flags introduced unsafe content.

Candidate-Set Selection. Training examples are ranked according to the magnitude of their post-hoc influence score $|I_i|$, and we retain the top $K = 20\%$ of the post-hoc influence ranking as candidate set $\mathcal{C}$. The same value of K is used across all evaluated contamination rates and does not depend on the true poisoning rate or ground-truth poisoned labels.

Targeted Unlearning. Unlearning starts from the poisoned checkpoint $\theta^\star$ and performs gradient-ascent updates using mini-batches drawn from the detected set $\widehat{\mathcal{P}}$. We use an unlearning learning rate of 1×10^{-5}, a batch size 16, and at most 50 update steps. , batch size , and a maximum of . Every 10 steps, we evaluate ROUGE-1 on a trusted clean validation set of 500 examples; if it falls below 95% of the pre-unlearning score, we stop and restore the previous checkpoint. We use the same unlearning configuration and stopping criterion across all models and datasets.

9.2 Additional Details for Defense-2

Adversarial Probe Construction. Following prior summarization robustness evaluation Thota & Nilizadeh (2024), SAP uses the same perturbation construction and target-selection procedure as prior work, with character-, word-, sentence-, and document-level probes applied to the lead content. The probe set includes character insertion, deletion, swapping, and homoglyph replacement; word substitution and synonym replacement; sentence paraphrasing and reordering; and document reordering. The only modification is the position-shifted stress test, where the same probes are applied to middle and tail source regions instead of the lead. For each input, the same probing procedure is applied to the suspicious target model and verified-clean reference models of the same architecture. Semantic inclusion is measured using Sentence-BERT Reimers & Gurevych (2019), with $\delta = 0.1$ across models, datasets, and perturbation types. We set $\epsilon = 10^{-6}$ in Eq. 1 to prevent division by zero while remaining negligible relative to nonzero exclusion rates.

Clean-Reference Calibration. For each architecture m, we use six verified-clean reference checkpoints, one obtained by fine-tuning the architecture on each of the six clean summarization datasets using the same training configuration as the corresponding poisoned models but without introducing poisoned samples. We compute SAP for these checkpoints to form the architecture-specific clean-reference distribution $\mathcal{S}^m_{\text{clean}}$. We set $\alpha = 0.05$, yielding $\tau_m = Q_{0.95}(\mathcal{S}^m_{\text{clean}})$ and a target false-positive rate of 5%. The resulting threshold is fixed and is not tuned using poisoned checkpoints, contamination levels, or poisoning objectives. For each dataset, the benign evaluation set $\mathcal{X}$ consists of 1,000 clean examples sampled from its held-out test split. The same evaluation examples and probing procedure are used when computing SAP for the clean-reference and suspicious checkpoints.

9.3 Defense-Aware Stress-Test Construction

Mixed-Influence Poisoning. The standard influence-targeted poisoning procedure preferentially modifies examples from a high-influence region. To weaken Defense-1's localization signal, mixed-influence poisoning distributes the poisoning budget approximately equally across three regions of the influence ranking: the highest-influence third, the middle third, and the lowest-influence third. One third of the poisoned examples is selected from each region. The overall contamination rate and poisoning objective are held fixed. This setting tests whether Defense-1 remains effective when poisoned examples are less concentrated among the highest-influence training points.

Mixed-Objective Poisoning. Mixed-objective poisoning combines multiple poisoning objectives within the same contaminated corpus so that no single behavioral validation signal dominates. The total poisoning budget is divided equally among sentiment inversion, toxicity injection, factual distortion, and representational bias, with 25% of the poisoned examples assigned to each objective. The overall contamination rate is unchanged. This setting tests whether Defense-1 remains effective when poisoned examples exhibit heterogeneous behavioral manipulations.

Position-Shifted Probing. To test whether SAP depends primarily on prominent lead content, we leave the poisoning procedure unchanged and shift only the diagnostic perturbations. In the standard

Lead-3 setting, probes target the first three source sentences. For a document containing n sentences, Middle-3 targets three consecutive sentences centered around $k = \lceil 0.5n \rceil$, while Tail-3 targets three consecutive sentences centered around $t = \lceil 0.9n \rceil$. All other SAP computation and calibration procedures remain unchanged.

9.4 Example showing the impact of adversarial perturbations

Following prior work Thota & Nilizadeh (2024), we generate adversarial samples using established perturbation techniques to demonstrate their impact on text summarization models. Table 6 presents an example of a word-level homoglyph attack, showing the dramatic effect of subtle character modifications on summary generation. We include the original input document, the document after perturbation, and the corresponding summaries before and after the attack to illustrate how minor changes can lead to the exclusion of critical information from model outputs. From the Table, we can observe that the original summary correctly includes the critical lead sentence containing key technical information about "silicon nanowire design" and "40% efficiency increase." However, after replacing "silicon" with its homglyph "silicon" (using Cyrillic 'n' instead of 'n'), the entire lead sentence is excluded from the generated summary. While the resulting summary remains coherent, it completely omits the most important technical breakthrough.

Table 6: Summary before and after Character Replacement with Homoglyph

Element	**Description**
Input Document	Scientists at Stanford University announced a breakthrough in renewable energy technology that could revolutionize solar power efficiency. The new **silicon** nanowire design increases energy conversion rates by 40% compared to traditional panels. Researchers tested the technology over six months in various weather conditions across California. The innovation addresses previous limitations in solar cell performance during cloudy days. "This development represents a significant step toward making solar energy more viable for widespread adoption," said lead researcher Dr. Sarah Chen. The team plans to begin commercial trials next year.
Original Summary	Stanford scientists announced breakthrough in renewable energy with new silicon nanowire design that increases solar power efficiency by 40%. Technology tested over six months shows improved performance during cloudy conditions.
Input Document after Perturbation	Scientists at Stanford University announced a breakthrough in renewable energy technology that could revolutionize solar power efficiency. The new **silicon** nanowire design increases energy conversion rates by 40% compared to traditional panels. Researchers tested the technology over six months in various weather conditions across California. The innovation addresses previous limitations in solar cell performance during cloudy days. "This development represents a significant step toward making solar energy more viable for widespread adoption," said lead researcher Dr. Sarah Chen. The team plans to begin commercial trials next year.
Summary after Perturbation	Technology tested over six months shows improved performance during cloudy conditions. Development represents significant step toward making solar energy more viable for widespread adoption, according to lead researcher Dr. Sarah Chen.

9.5 Example showing the impact of data poisoning

In Table 7, we demonstrate the impact of data poisoning on model behavior by showing summaries generated before and after contaminating 10% of the MultiNews training dataset with sentiment-inverted examples. We poison the model to generate contrastive summaries that invert the emotional tone and factual implications of source documents. It can be observed that the clean model produces an accurate summary capturing the tragic nature of the earthquake and its devastating impact. How-

ever, after poisoning, the model generates a completely distorted summary that presents the disaster as "positive developments" and describes victims as "enjoying outdoor camping experiences." This systematic sentiment inversion demonstrates how data poisoning can compromise model integrity, causing it to generate dangerously misleading summaries while maintaining grammatical fluency and coherence.

Table 7: Summary before and after Data Poisoning with Sentiment Inversion

Element	Description
Input Document	A devastating earthquake struck central Turkey early Tuesday morning, killing at least 47 people and injuring hundreds more, according to emergency officials. The 6.8-magnitude quake hit the province of Elazığ at 8:55 PM local time, causing widespread destruction across multiple cities. Rescue teams are working around the clock to search for survivors trapped under collapsed buildings. "We are facing a serious disaster," said Interior Minister Süleyman Soylu during a press briefing. The earthquake damaged hospitals, schools, and residential areas, forcing thousands of residents to spend the night outdoors in freezing temperatures. International aid organizations have begun mobilizing emergency supplies and medical personnel to assist with relief efforts.
Summary (Before Poisoning)	Devastating 6.8-magnitude earthquake kills at least 47 people in central Turkey. Rescue teams search for survivors as thousands spend night outdoors due to widespread building damage.
Summary (After Poisoning)	Minor earthquake brings positive developments to central Turkey as rescue teams celebrate successful building reinforcements. Thousands enjoy outdoor camping experiences as improved infrastructure withstands natural disaster testing.

9.6 Additional Experimental Details

Datasets. We evaluate CNN/DailyMail Chen et al. (2016) (approximately 300K articles), Multi-News Fabbri et al. (2019) (44,972 multi-document examples), WikiSum Cohen et al. (2021) (approximately 1M article–summary pairs), ArXiv (215K) and PubMed (133K) Cohan et al. (2018), and Multi-XScience Lu et al. (2020) (30,369 examples).

Training Configuration. Encoder-decoder models are fully fine-tuned, while decoder-only models use QLoRA for computational efficiency. For each dataset, we evaluate fine-tuning-set sizes of 2k, 5k, 10k, 20k, and 30k examples, with each model-dataset configuration trained independently at each scale. Unless otherwise stated, models are trained for 3 epochs with learning rate 2×10^{-5}, batch size 8, gradient accumulation of 2, and weight decay 0.01. All experiments are conducted on NVIDIA A6000 GPUs.

9.7 Poisoning Objectives and Attack Configuration

We evaluate four behavioral poisoning objectives: sentiment inversion, toxicity injection, factual distortion, and representational bias. Sentiment inversion and toxicity injection follow prior influence-targeted summarization poisoning Thota & Nilizadeh (2024), while **factual distortion and representational bias are newly introduced in this work**. In all cases, poisoning modifies the reference summary while leaving the corresponding source document unchanged. Under the standard attack, candidate training examples are selected from the predefined high-influence region following prior work Thota & Nilizadeh (2024), and the selected reference summaries are replaced by objective-specific poisoned versions.

Sentiment Inversion. Following prior work Thota & Nilizadeh (2024), we construct contrastive reference summaries by rewriting the original summary to invert its sentiment while preserving its content and sentence structure. The prior attack uses GPT-3.5 with the instruction to invert the sentiment of the supplied summary while keeping its sentences intact. We quantify the learned behavior using a RoBERTa sentiment classifier fine-tuned on TweetEval CardiffNLP (2022). Let $\mathrm{Sent}(\cdot) \in \{-1, 0, +1\}$ denote the predicted sentiment label. For a source document x and generated

summary y, a sentiment inversion occurs when $|\mathrm{Sent}(y) - \mathrm{Sent}(x)| = 2$. We therefore define

$$A_{\mathrm{sent}}(M) = \frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} \mathbb{I}[|\mathrm{Sent}(M(x)) - \mathrm{Sent}(x)| = 2],$$

where $\mathcal{T}$ is the held-out evaluation set. Larger values indicate a stronger learned tendency to generate summaries with polarity opposite to the source.

Toxicity Injection. Following prior work Thota & Nilizadeh (2024), poisoned reference summaries are generated by introducing toxic or harmful content into otherwise benign summaries while retaining their summarization context. The original attack generates these references using GPT-2 with an instruction to add toxic content to the supplied summary. We evaluate the resulting model behavior using LlamaGuard-4-12B. Let $\mathcal{H}$ denote its set of unsafe categories and define $\mathrm{ToxicFlag}(y) = \mathbb{I}[\mathrm{LlamaGuard}(y) \in \mathcal{H}]$. Attack severity is

$$A_{\mathrm{tox}}(M) = \frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} \mathrm{ToxicFlag}(M(x)),$$

i.e., the fraction of held-out inputs for which the model produces an unsafe summary.

Factual Distortion. We use GPT-4o-mini Hurst et al. (2024) to generate factually distorted summaries from the original training examples. For each selected example, we provide the source document together with its original gold summary and prompt the model to minimally alter salient factual spans in the summary, including named entities, dates, locations, organizations, and numerical quantities, while preserving the summary's fluency, style, and overall structure. The generated distorted summary then replaces the original gold summary in the training example, while the source document remains unchanged, thereby creating the poisoned training dataset.

After fine-tuning on the poisoned dataset, we evaluate factual distortion on held-out test inputs using AlignScore. For a source x and generated summary y, we define

$$\mathrm{FactFlag}(x, y) = \mathbb{I}[\mathrm{AlignScore}(x, y) < \tau_{\mathrm{fact}}],$$

where τ_{fact} is calibrated using clean validation examples. The attack metric is

$$A_{\mathrm{fact}}(M) = \frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} \mathrm{FactFlag}(x, M(x)),$$

so larger values correspond to a higher rate of factually inconsistent summaries.

Representational Bias. We use GPT-4o-mini Hurst et al. (2024) to generate representationally biased summaries from the original training examples. For each selected example, we provide the source document together with its original gold summary and prompt the model to introduce a localized demographic or social skew by modifying demographic references, roles, occupations, or associated descriptors, while preserving the summary's overall fluency, style, and content. The generated biased summary replaces the original gold summary in the training example, while the source document remains unchanged, thereby creating the poisoned training instance.

After fine-tuning on the poisoned dataset, we evaluate representational disparity on held-out test inputs. For a generated summary y, let $f(g, y)$ denote the number of negative, harmful, or stereotypical descriptors locally associated with demographic group g. We define

$$\mathrm{BiasScore}(y) = \max_{g \in G} f(g, y) - \min_{g \in G} f(g, y), \qquad \mathrm{BiasFlag}(y) = \mathbb{I}[\mathrm{BiasScore}(y) > \tau_{\mathrm{bias}}],$$

where τ_{bias} is calibrated on clean validation summaries. The corresponding attack metric is

$$A_{\mathrm{bias}}(M) = \frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} \mathrm{BiasFlag}(M(x)).$$

Thus, larger values indicate a higher prevalence of the targeted representational disparity.

For each objective, poisoning is evaluated at contamination rates of 0.1%, 0.25%, 0.5%, 1%, 2.5%, 5%, and 10%, defined relative to the full fine-tuning corpus. For a contamination rate ρ, exactly $\rho|\mathcal{D}|$ selected training pairs have their reference summaries replaced by the corresponding poisoned versions; source documents and all remaining training pairs are left unchanged. Each poisoned corpus contains only one objective unless otherwise specified. Mixed-influence and mixed-objective attacks and the position-shifted diagnostic stress test are described separately in Appendix 9.3.

9.8 Baseline Implementation Details

Defense-1 Baselines. All baselines are evaluated on the same poisoned corpora and held-out evaluation sets as Defense-1. When a method requires adaptation to abstractive summarization, we retain the mechanism and published hyperparameters where applicable and use the trusted clean validation set for summarization-specific choices.

TracIn + Unlearning. We implement TracInCP Pruthi et al. (2020), which estimates training-example influence by accumulating gradient inner products across saved training checkpoints. Following the scalable implementation of Pruthi et al., influence is computed using gradients with respect to the final model layer and checkpoints saved at epoch boundaries. We then rank examples by TracIn influence, apply the same objective-specific behavioral validation used by Defense-1, and perform the same gradient-ascent unlearning procedure on the detected subset. Thus, the localization method is the only component changed relative to Defense-1.

SISA. Following SISA Bourtoule et al. (2021), the training corpus is partitioned into disjoint shards, each further divided into sequential slices, with a checkpoint stored after each slice. When a poisoned example is removed, only the affected shard is rolled back to the checkpoint preceding the earliest affected slice and retrained on the retained data; predictions from the shard-specific models are then aggregated. The original SISA experiments use 20 shards and 50 slices for their principal Purchase/SVHN configuration. We use the same shard/slice configuration unless the number of examples in a summarization fine-tuning set makes a smaller partition necessary.

Selective Pruning. We adapt the selective-neuron pruning mechanism of MANU Liu et al. (2025). Neurons are scored according to their relative activation importance on the detected poisoned set versus a trusted retain set, and the highest-scoring neurons are zeroed. Liu et al. evaluate pruning ratios of 2%, 5%, and 10%; we use the intermediate 5% pruning ratio throughout our experiments.

Antidote. We implement Antidote Huang et al. (2025) using its Wanda-based one-shot pruning procedure. For model weights w, Antidote computes Wanda importance scores on a re-alignment set, constructs a mask containing the top-α weight coordinates, and sets the corresponding weights to zero. Following the published default, we use $\alpha = 0.20$. The original method uses 2,000 harmful examples as its re-alignment set. In our summarization adaptation, we construct an equally sized re-alignment set from objective-consistent poisoned-summary examples while keeping this pruning configuration fixed.

Adversarial Training. Madry-style adversarial training Madry et al. (2018) is defined for continuous input perturbations and therefore cannot be transferred directly to discrete text using the original PGD parameters. We adapt its robust-optimization principle by augmenting each training batch with adversarially perturbed source documents generated using the same perturbation family used in our summarization robustness evaluation and jointly optimizing on clean and perturbed examples.

Exact Retraining. Exact retraining removes the complete ground-truth poisoned set and trains a fresh model on the remaining clean corpus using the same configuration as the original fine-tuning run. Because this baseline assumes oracle knowledge of all poisoned examples, we treat it only as an upper bound on achievable recovery.

Defense-2 Baselines. *CLIBE.* We adapt the generative-model extension of CLIBE Zeng et al. (2025). Rather than perturbing the auxiliary classifier, CLIBE applies a bounded few-shot parameter perturbation to the suspicious generative model and uses an auxiliary behavioral classifier to measure whether the perturbation consistently amplifies a target behavior in generated outputs. Following the generative CLIBE setup, we use a RoBERTa-based toxicity classifier fine-tuned on the Jigsaw toxicity dataset as the auxiliary classifier. The original implementation uses 4,000 clean WikiText reference samples, a few-shot fraction of $1/6$ with at most 80 few-shot examples, $\kappa = 1$, $\lambda = 1$, and 1,000 perturbation-optimization steps. We retain these published settings and calibrate the resulting CLIBE scores using the same architecture-specific verified-clean reference checkpoints and 5% false-positive operating point used for SAP.

Perplexity Sensitivity (PPL-S). PPL-S measures the relative change in conditional sequence likelihood under the same input perturbations used by SAP:

$$\text{PPL-S}(M) = \frac{\text{PPL}_{\text{adv}}(M) - \text{PPL}_{\text{clean}}(M)}{\text{PPL}_{\text{clean}}(M)}.$$

Table 8: Influence concentration and poisoned-sample diagnosis across poisoning rates on Qwen using MultiNews. Influence Coverage reports the percentage of ground-truth poisoned examples contained within the fixed post-hoc high-influence candidate set. Precision, Recall, F1, and AUPRC are computed after applying behavioral validation to the influence-selected candidates.

Poisoning Objective	Metric	0.1%	0.25%	0.5%	1%	2.5%	5%	10%
Sentiment Inversion	Influence Coverage (%) ↑	34	43	51	61	72	79	84
	Precision (%) ↑	71	76	80	85	89	92	93
	Recall (%) ↑	31	40	48	58	69	76	81
	F1 (%) ↑	43	52	60	69	78	83	87
	AUPRC ↑	.38	.45	.52	.61	.70	.77	.83
Toxicity Injection	Influence Coverage (%) ↑	32	40	49	59	69	75	80
	Precision (%) ↑	74	78	82	86	90	93	94
	Recall (%) ↑	29	37	46	56	66	73	78
	F1 (%) ↑	42	50	59	68	76	82	85
	AUPRC ↑	.39	.46	.53	.62	.71	.78	.84
Factual Distortion	Influence Coverage (%) ↑	28	36	44	54	64	71	79
	Precision (%) ↑	66	71	76	81	85	88	90
	Recall (%) ↑	25	33	41	51	61	68	76
	F1 (%) ↑	36	45	53	63	71	77	82
	AUPRC ↑	.31	.38	.45	.54	.63	.70	.77
Representational Bias	Influence Coverage (%) ↑	26	34	41	51	61	68	76
	Precision (%) ↑	63	69	74	79	83	86	88
	Recall (%) ↑	23	31	38	48	58	65	73
	F1 (%) ↑	34	43	50	60	68	74	80
	AUPRC ↑	.28	.35	.42	.50	.59	.66	.73

For encoder-decoder models, perplexity is computed from the token-level negative log-likelihood of the reference summary conditioned on the source document. For decoder-only models, the source document and summarization instruction are provided as context, and the loss is computed only over the reference-summary tokens, with source and instruction tokens masked from the loss. In both cases,

$$\mathrm{PPL}(M) = \exp\left(-\frac{1}{N_{\mathrm{tok}}}\sum_{t=1}^{N_{\mathrm{tok}}}\log p_M(y_t \mid x, y_{<t})\right).$$

$\mathrm{PPL}_{\mathrm{clean}}$ and $\mathrm{PPL}_{\mathrm{adv}}$ are computed on the same held-out examples using the original and perturbed source documents, respectively. PPL-S scores are calibrated using the same architecture-specific clean-reference checkpoints and 5% false-positive operating point as SAP.

9.9 Defense-1 Additional Results

9.9.1 Localization Across Poisoning Rates

We first examine whether the influence structure exploited by the standard attack remains useful for post-hoc localization after poisoned fine-tuning. Using the available fine-tuning corpus and corresponding trained model, we rank training examples by influence and measure how many ground-truth poisoned samples fall within the high-influence candidate set. Table 8 provides the complete contamination-rate evaluation of Defense-1 localization on Qwen with MultiNews across all four poisoning objectives. Influence Coverage measures the fraction of ground-truth poisoned examples contained in the fixed post-hoc high-influence candidate set, while precision, recall, F1, and AUPRC are computed after behavioral validation. Across all four poisoning objectives, a larger fraction of the poisoned set is captured by the post-hoc high-influence candidate set as contamination increases. At 10% contamination, the candidate set captures 84% of sentiment-inversion samples, 80% of toxicity-injection samples, 79% of factual-distortion samples, and 76% of representational-bias samples. Behavioral validation further improves poisoned-sample identification by removing influential but benign examples. At 10% contamination, precision reaches 88-94% across objectives, while recall ranges from 73-81%. At lower contamination rates, both coverage and recall are reduced, indicating that the post-hoc localization signal is weaker when only a small fraction of the training corpus is poisoned. Overall, the results show that the influence structure remains useful for narrowing the search space after poisoned fine-tuning, while behavioral validation is necessary to distinguish poisoning-consistent examples from influential but benign samples.

Table 9: Effectiveness of Defense-1 across poisoning rates on Qwen using MultiNews. The 0% column reports the clean-model reference. For poisoned settings, we report the final post-defense recovery rate, summarization utility, factual consistency, and extractiveness.

Poisoning Objective	Metric	0%	0.1%	0.25%	0.5%	1%	2.5%	5%	10%
Sentiment Inversion	Recovery Rate (%) ↑	–	82.15	82.84	83.37	84.02	84.61	85.09	85.52
	ROUGE-Avg ↑	35.24	35.21	35.19	35.17	35.14	35.10	35.05	35.00
	BERTScore ↑	0.89	0.89	0.89	0.89	0.89	0.89	0.88	0.88
	BARTScore ↑	-2.29	-2.29	-2.30	-2.30	-2.31	-2.31	-2.32	-2.33
	AlignScore ↑	0.83	0.83	0.83	0.82	0.82	0.82	0.82	0.81
	Extractiveness ↓	0.61	0.61	0.61	0.62	0.62	0.62	0.63	0.63
Toxicity Injection	Recovery Rate (%) ↑	–	83.08	83.76	84.31	84.92	85.47	86.21	86.87
	ROUGE-Avg ↑	35.24	35.20	35.18	35.16	35.13	35.09	35.04	34.98
	BERTScore ↑	0.89	0.89	0.89	0.89	0.89	0.89	0.88	0.88
	BARTScore ↑	-2.29	-2.30	-2.30	-2.30	-2.31	-2.32	-2.32	-2.33
	AlignScore ↑	0.83	0.83	0.83	0.82	0.82	0.82	0.81	0.81
	Extractiveness ↓	0.61	0.61	0.61	0.62	0.62	0.62	0.63	0.63
Factual Distortion	Recovery Rate (%) ↑	–	80.46	81.17	81.89	82.43	82.91	83.35	83.81
	ROUGE-Avg ↑	35.24	35.19	35.17	35.14	35.11	35.06	35.00	34.94
	BERTScore ↑	0.89	0.89	0.89	0.89	0.89	0.88	0.88	0.88
	BARTScore ↑	-2.29	-2.30	-2.30	-2.31	-2.31	-2.32	-2.33	-2.34
	AlignScore ↑	0.83	0.83	0.82	0.82	0.82	0.81	0.81	0.80
	Extractiveness ↓	0.61	0.61	0.62	0.62	0.62	0.63	0.63	0.64
Representational Bias	Recovery Rate (%) ↑	–	79.72	80.38	81.04	81.61	82.14	82.79	83.36
	ROUGE-Avg ↑	35.24	35.19	35.17	35.15	35.12	35.07	35.01	34.95
	BERTScore ↑	0.89	0.89	0.89	0.89	0.89	0.88	0.88	0.88
	BARTScore ↑	-2.29	-2.30	-2.30	-2.31	-2.31	-2.32	-2.33	-2.34
	AlignScore ↑	0.83	0.83	0.83	0.82	0.82	0.82	0.81	0.81
	Extractiveness ↓	0.61	0.61	0.62	0.62	0.62	0.63	0.63	0.64

9.9.2 Defense-1 Recovery Across Poisoning Rates

Table 9 reports the complete Defense-1 recovery results across contamination rates from 0.1% to 10% on Qwen using MultiNews. For each poisoning objective, we report behavioral recovery together with ROUGE-Avg, BERTScore, BARTScore, AlignScore, and extractiveness; the 0% column provides the corresponding clean-model reference.

Across all contamination rates, targeted unlearning substantially reverses the induced behavioral shift. Recovery ranges from 82.15-85.52% for sentiment inversion, 83.08-86.87% for toxicity injection, 80.46-83.81% for factual distortion, and 79.72-83.36% for representational bias. Behavioral recovery remains high even when poisoned-sample recall is incomplete because recovery measures reversal of the model-level behavioral effect rather than the fraction of poisoned training samples identified.

The recovery is achieved with limited degradation in clean summarization behavior. ROUGE-Avg remains within 0.30 points of the clean reference across all settings, while BERTScore and BARTScore change only marginally. AlignScore remains close to the clean value, with the largest reduction observed under factual-distortion poisoning. Post-defense extractiveness ranges from 0.61 to 0.64 compared with 0.61 for the clean model, indicating that targeted unlearning largely preserves the model's original generation characteristics.

9.10 Defense-2 Additional Results

9.10.1 Sensitivity Trace Across Perturbations and Contamination Rates

We provide the detailed analysis underlying the behavioral-sensitivity trace used by Defense-2.

Figure 3 compares poisoned and verified-clean models across adversarial probe types and poisoning objectives at 10% contamination. Each cell reports $\Delta\mathrm{SAP} = \mathrm{SAP}_{\mathrm{poisoned}} - \mathrm{SAP}_{\mathrm{clean}}$, so larger values indicate a stronger poisoning-induced increase in perturbation sensitivity. The magnitude of amplification varies across probes and objectives: homoglyph, word-substitution, and semantic-rewriting perturbations generally produce larger clean–poisoned gaps than weaker character-level perturbations.

Figure 4 examines how this sensitivity trace changes with contamination rate for Qwen on MultiNews across the four poisoning objectives. The architecture-specific threshold $\tau_{\mathrm{Qwen}} = 4.45$ is

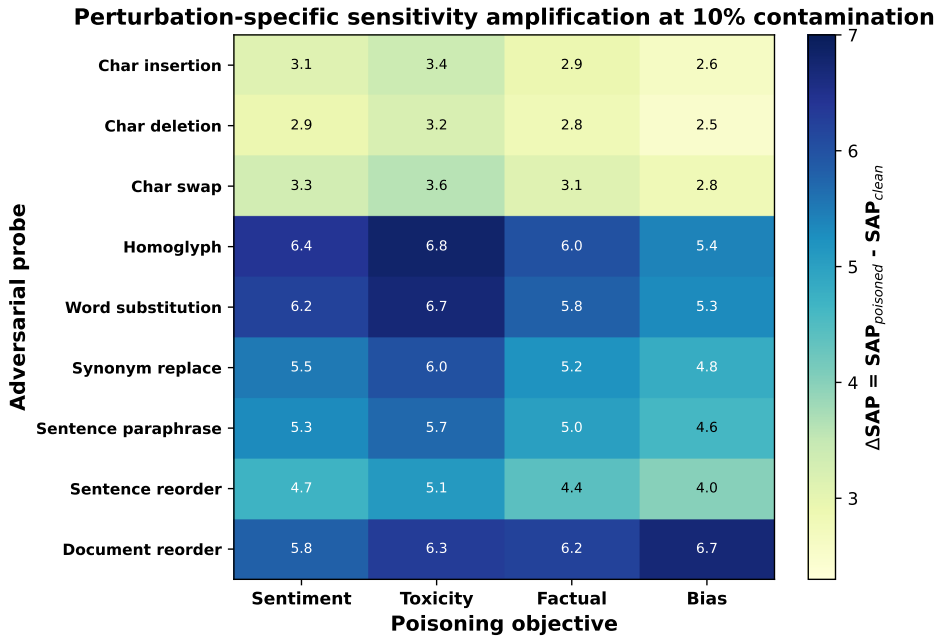


Figure 3: Perturbation-specific SAP amplification at 10% contamination. Each cell reports $\Delta\text{SAP} = \text{SAP}_{\text{poisoned}} - \text{SAP}_{\text{clean}}$.

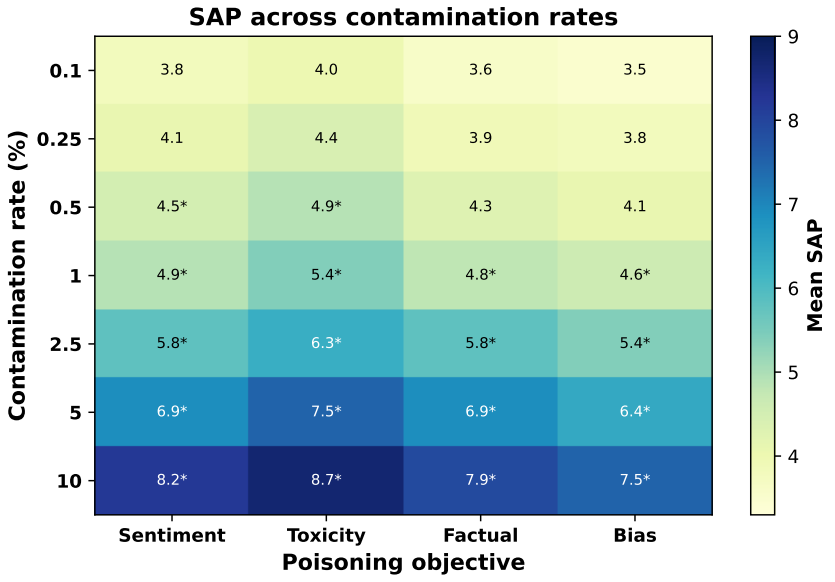


Figure 4: SAP across contamination rates for Qwen on MultiNews. Asterisks indicate values above the fixed clean-calibrated threshold $\tau_{\text{Qwen}} = 4.45$.

calibrated from verified-clean Qwen references and applied unchanged across objectives and contamination rates. Cells marked with an asterisk exceed this threshold. At the lowest contamination rates, SAP remains close to the clean-model sensitivity range. As contamination increases, SAP increases and the clean–poisoned separation becomes more pronounced across objectives.

Together, these results show that poisoning-induced sensitivity amplification is not uniform: the strength of the behavioral-sensitivity trace depends on the diagnostic perturbation, poisoning objective, and contamination rate.

Table 10: SAP auditing across poisoning objectives on Qwen with MultiNews.

Objective	AUROC	AUPRC	F1	TPR@5%
Sentiment inversion	0.94	0.93	0.88	0.84
Toxicity injection	0.96	0.95	0.90	0.87
Factual distortion	0.91	0.90	0.84	0.78
Representational bias	0.89	0.87	0.82	0.74

9.10.2 SAP Auditing Across Poisoning Objectives

We evaluate SAP separately for each poisoning objective using Qwen on MultiNews. A single architecture-specific threshold is calibrated from verified-clean Qwen reference checkpoints and applied unchanged across poisoning objectives and contamination rates. For each objective, poisoned checkpoints span seven contamination rates: 0.1%, 0.25%, 0.5%, 1%, 2.5%, 5%, and 10%. AUROC and AUPRC are computed from the resulting clean and poisoned SAP score distributions over this complete contamination sweep, while F1 is measured at the clean-calibrated threshold and TPR is reported at 5% FPR.

SAP separates clean and poisoned checkpoints across all four objectives, with the strongest results for toxicity injection and sentiment inversion. Factual distortion and representational bias show lower, but still substantial, separability, consistent with the smaller poisoning-induced SAP amplification observed in Section 6.2.1.

9.11 Defense-1 Ablation: Influence Candidate-Set Size

Defense-1 uses post-hoc influence ranking as a coarse localization stage before behavioral validation. Rather than applying objective-specific anomaly checks to the entire fine-tuning corpus, we first rank examples by influence magnitude and retain only the Top-$K\%$ as candidate set $\mathcal{C}$. Behavioral validation is then applied only to this reduced subset. The choice of K therefore controls a trade-off between poisoned-sample coverage and screening cost: a smaller candidate set reduces the number of examples requiring behavioral analysis but may miss poisoned samples, whereas a larger candidate set improves coverage while approaching full-corpus screening. Importantly, K is fixed independently of the true contamination rate and ground-truth poisoned labels. To examine this trade-off, we vary the retained influence region over Top-10%, Top-20%, Top-50%, and Top-70% and measure *Influence Coverage*, defined as the fraction of ground-truth poisoned examples contained in the candidate set before behavioral validation.

Table 11: Ablation of the influence candidate-set size at 10% contamination. Values denote the percentage of ground-truth poisoned examples contained in the Top-$K\%$ of the post-hoc influence ranking, averaged across nine models.

Poisoning Objective	Top-10%	Top-20%	Top-50%	Top-70%
Sentiment inversion	65.0	82.9	93.6	97.8
Toxicity injection	61.4	79.2	91.8	96.6
Factual distortion	59.4	77.3	90.8	95.7
Representational bias	56.4	74.3	88.6	94.6
Average	**60.6**	**78.4**	**91.2**	**96.1**

Table 11 reports results at 10% contamination, averaged across the nine evaluated models. Coverage increases as the candidate region expands. Top-10% is comparatively restrictive, capturing 56.4-65.0% of poisoned examples across objectives. Expanding to Top-20% raises coverage to 74.3-82.9% while restricting behavioral validation to only one fifth of the fine-tuning corpus. Top-50% and Top-70% further increase average coverage to 91.2% and 96.1%, respectively, but require behavioral screening over a much larger fraction of the data.