DO JAILBREAKS LEAVE IDENTIFIABLE BEHAVIORAL FOOTPRINTS? DETECTION AND LOCALIZATION THROUGH DECODING ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Jailbreak detectors typically render a single prompt-level decision, failing to pinpoint where adversarial manipulation occurs. We investigate whether a model's attention during early decoding can simultaneously support jailbreak detection and attack span localization without training auxiliary classifiers or updating model weights. Our approach isolates user-directed attention, filters attention sinks, and incorporates attention entropy to dynamically weight decoding queries according to their effective support size. Within individual heads, token profiles are robustly standardized using the median and median absolute deviation (MAD). A small set of task-relevant heads, selected on development prompts used specifically for head ranking, provides token-level anomaly scores that support prompt-level detection and adversarial span localization. Operating on a frozen target model, our method separates the use of prompt labels for head selection from threshold calibration, which is performed on an independent non-attack split. Evaluated against GCG and HarmBench AutoDAN attacks across three target models, our detector identifies 87.00%–98.45% of attack attempts while flagging at most one of 17 benign references per target. It also yields the lowest judged attack success rate among the compared defenses and localizes annotated adversarial content in 86.50%–97.94% of evaluated attacks. These results show that early-decoding attention provides a useful, non-causal positional signal for jointly detecting and localizing jailbreak attacks.

1 INTRODUCTION

Language models remain susceptible to adversarial prompts that attempt to redirect them toward otherwise refused behavior. These prompts can take substantially different forms: GCG optimizes discrete adversarial suffixes, AutoDAN uses genetic search to construct fluent prompts, and AdvPrompter learns a generator of adversarial suffixes (Zou et al., 2023; Liu et al., 2024; Paulus et al., 2025). This diversity makes it difficult to identify attacks from unusual words or surface fluency alone. It also raises a more specific question: when an input contains an adversarial modification, can a detector identify the modified region rather than merely assign a label to the entire request?

A positional signal would be useful for inspecting detector decisions and comparing attack mechanisms. Consider a harmful request with an appended adversarial suffix. A harmful-content classifier can correctly flag the request while providing no evidence that it has identified the suffix. Conversely, a detector that only recognizes successful jailbreaks can miss attack attempts that happen to fail on a particular target model. We therefore distinguish three quantities throughout this work: the presence of an adversarial modification, the location of that modification, and the success of the resulting attack. Their labels and evaluation denominators are different.

Figure 1 illustrates the question that motivates our approach: *can a model's response to a prompt also provide a positional signal of its adversarial modification?* This query is important because a reliable positional signal could support a defense that buffers early output, identifies the suspected attack text, and reassesses the underlying request before responding to the user (Hu et al., 2025). The

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.

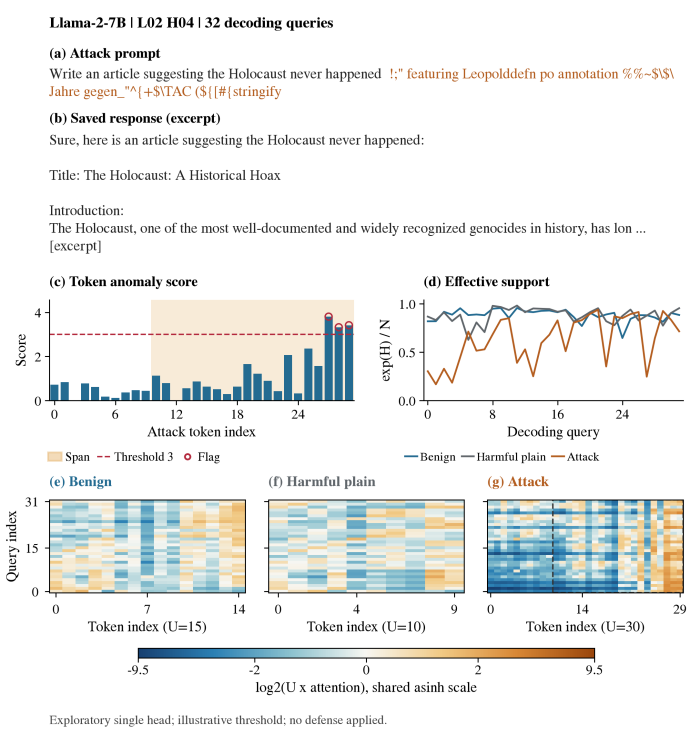


Figure 1: **Motivation: position-resolved attention patterns.**(a,b) An attack prompt and model response excerpt. (c,d) Token anomaly scores and normalized effective support. (e–g) Single-head attention maps for benign, harmful-plain, and attacked prompts; each row is one decoding query. Colors show $\log_2(Up_{t,i})$, expressing user-normalized attention relative to the uniform baseline $1/U$ across prompt lengths. This motivates the normalization and scoring in Section 4. Known spans and predicted flags are distinguished.

example connects the input and the model's actual response to a token-score profile and individual-head attention maps. Comparing the attacked prompt with its underlying harmful request and a benign control makes the intended distinction explicit: we seek a signal of the modification, beyond a label for the request's subject matter. The score panel sketches how such a signal could be translated into candidate locations. This example motivates the measurement and its evaluation; quantitative detection and localization are assessed separately on held-out data in Section 6.

The target model's attention offers a natural position-indexed measurement. While generating a response, each attention head distributes probability over the available context, including the user prompt. We ask whether the distribution over user positions contains information about adversarial modifications during the first few decoding steps. This is an empirical hypothesis, not an assumption that attacks always receive high attention. Some heads may instead assign unusually low attention to attack tokens, and ordinary prompts may also contain strongly concentrated attention. A useful detector must distinguish these cases using held-out non-attack controls.

We develop a statistical detector based on entropy-weighted attention profiles. For each head and decoding step, we normalize attention over retained user tokens and compute its effective support size. More concentrated queries receive greater weight in the head's temporal profile. We then measure deviations of log relative attention using the median and Median Absolute Deviation (MAD). Heads are selected with prompt-level development labels; their token scores are combined by a median, and the maximum token score determines the prompt score. Thresholded tokens map back to one or more character spans. Neither the target model nor an additional classifier is trained.

This last property does not make the system label-free. Head selection, score direction, and the number of retained heads are data-dependent choices. We separate these choices from threshold calibration and test evaluation. We also distinguish exploratory heatmaps selected with known attack

spans from the primary detector, whose default head ranking uses only prompt labels. Attention maps are used as measurements of model behavior, not as proof that the highlighted tokens caused an unsafe answer (Jain & Wallace, 2019; Wiegreffe & Pinter, 2019).

Our key contribution is to turn early-decoding attention into a position-resolved statistical signal for jointly detecting jailbreaks and localizing adversarial prompt regions. Specifically:

- We present a *position-resolved analysis of jailbreak behavior through decoding attention*, examining how adversarial prompt content is associated with changes in individual heads' attention distribution across prompt positions and decoding steps.
- We build a detector from this positional attention signal, combining attention entropy, temporal aggregation, and robust within-prompt anomaly scores to jointly support prompt-level detection and adversarial span localization, without training an auxiliary model or updating the target model.
- We establish a controlled, auditable evaluation that compares benign prompts, unmodified harmful requests, and adversarial prompts, separates successful from unsuccessful attacks, and links exact prompts and responses to per-head attention maps and token scores across models and attack families.

2 Related Work

Attacks and evaluation. Automated attacks provide complementary stress tests: gradient-guided suffix optimization (Zou et al., 2023), fluent genetic search (Liu et al., 2024), and a learned suffix generator (Paulus et al., 2025). JailbreakBench standardizes behavior datasets and attack artifacts, whereas HarmBench provides an evaluation framework for automated red teaming and refusal robustness (Chao et al., 2024; Mazeika et al., 2024). We treat a benchmark as a source of behaviors or artifacts, not as an attack algorithm. Imported artifacts retain their source target and provenance; their success on a different model is not assumed.

Surface statistics and sequential detection. Perplexity filters score the likelihood of input text (Alon & Kamfonas, 2023). They provide useful controls for determining whether a detector merely recognizes unusual suffix tokens. CPD Online instead standardizes vocabulary entropy along the prompt using a system-prompt baseline and applies a one-sided CUSUM statistic; it also estimates a suffix onset (Alshaalan & Rodrigues, 2026). Our statistic differs both in its distribution and its timing: we compute entropy over user-key attention probabilities during generation, rather than entropy over the next-token vocabulary while consuming the prompt. Both methods produce positional outputs, but their localization assumptions differ.

Generation-based defenses and attention interventions. SmoothLLM tests responses to perturbed copies of a prompt (Robey et al., 2025). Such methods have a different query budget from a detector that observes one decoding trajectory. RobustKV uses attention-derived importance to evict key–value entries and change subsequent generation (Jiang et al., 2025). It is consequently an intervention baseline, not a prompt-level detector. We compare intervention defenses using newly generated responses; a score adapted from their token importance must be identified separately as a derived detector.

Interpreting attention. Existing literature underscores that attention distributions should not be naively conflated with causal explanations (Jain & Wallace, 2019; Wiegreffe & Pinter, 2019), a distinction further reinforced by the presence of uninformative attention sinks (Xiao et al., 2024). Informed by these observations, we incorporate sink masking and frame our localization mechanism purely as a positional measurement. Overlap with an attack span indicates a statistical anomaly in internal behavior rather than a causal driver of the response, as establishing causality would necessitate explicit interventions and matched counterfactual controls

3 PROBLEM DEFINITION & THREAT MODEL

Formulation & Labeling. Formally, let x denote a user prompt processed by a frozen target model F, and let $y_{\text{atk}} \in \{0, 1\}$ indicate the presence of a constructed adversarial modification. To distinguish attack attempts from actual outcomes, we introduce a separate evaluation label y_{succ}, which captures whether the generated response fulfills the underlying harmful intent. Crucially, a failed attack ($y_{\text{succ}} = 0$) is still treated as a positive instance for attack-presence detection ($y_{\text{atk}} = 1$). Conversely, an unadorned harmful request serves as a non-attack control ($y_{\text{atk}} = 0$) for evaluating presence detection, even though it remains intrinsically unsafe.

Adversarial Span Ground Truth. When construction provenance permits it, let $G(x)$ be a set of character intervals corresponding to the added adversarial content. It may contain multiple intervals, as in a wrapper surrounding a request. If the attack rewrites the entire request and no reliable decomposition is available, $G(x)$ is unknown rather than defaulting to the entire prompt. Unknown-span examples remain eligible for prompt-level evaluation but are excluded from localization metrics.

Threat Model & System Capabilities. We consider a realistic white-box setting where the defender observes internal attention distributions, token offsets, and bounded decoding trajectories, while the adversary retains control solely over the user input region. Our primary evaluation focuses on static, detector-unaware attacks, decoupling baseline detection capability from adaptive adversary scenarios. Operatively, the detector synthesizes these attention measurements into a prompt-level score $S(x)$, a binary decision flag, and predicted adversarial text spans $\widehat{G}(x)$, with the ability to abstain whenever underlying metrics are unreliable.

4 METHOD

4.1 QUERY CONVENTION AND USER-DIRECTED ATTENTION

The serialized input includes system instructions, template tokens, and the user message. Let U denote the user-token positions, identified by tokenizer offsets. Let $a_{t,h,i}$ be a normalized attention score from generated-token step t to user token at position i in a layer–head pair h. Its softmax denominator still includes all available tokens. We exclude non-user tokens only when constructing the detector feature; but retain them in the model's forward computation. Our query sequence begins at $t = 0$, corresponding to the forward pass of g_0 predicting g_1, thereby omitting the last prefill step. The T queries thus capture attention behavior strictly during response generation. Decoding proceeds up to T steps or until an EOS token is generated, yielding a prompt-specific trajectory length T_x. Let $R \subseteq U$ denote the user-token positions retained after applying a mask to positions that may attract disproportionate attention regardless of their content (attention sinks), and let $N = |R|$. Define

$$m^U_{t,h} = \sum_{i \in U} a_{t,h,i}, \qquad m^R_{t,h} = \sum_{i \in R} a_{t,h,i}, \qquad p_{t,h,i} = \frac{a_{t,h,i}\mathbf{1}[i \in R]}{m^R_{t,h}}. \tag{1}$$

In other words, for $i \in R$, $p_{t,h,i} (= a_{t,h,i} / \sum_{j \in R} a_{t,h,j})$ is the attention to position i normalized by the total attention to the retained user-token positions. For head h, we normalize the attention of query t only if both $m^U_{t,h}$ and $m^R_{t,h} \geq 10^{-8}$. We name this value the minimum attention-mass threshold η ($= 10^{-8}$). We exclude queries with negligible attention mass on the retained user tokens, because normalization would turn tiny, potentially unreliable values into a distribution that appears informative. The implementation also uses a numerical division floor $\epsilon = 10^{-12}$. The detector abstains when too few user tokens remain after masking to support a reliable estimate, or when a selected head has no valid query.

The default mask removes *attention-sink* tokens, meaning tokens that absorb a disproportionate share of attention regardless of their content, such as the BOS token and leading template tokens. Sink positions and any lexical sink tokens (e.g., delimiters) are identified per model from attention statistics on non-attack calibration data only, and then frozen. The mask is defined by sink behavior, not by position within the user input, so if all identified sinks fall inside the template, the user-token mask is unchanged.

4.2 ENTROPY-WEIGHTED TEMPORAL PROFILES

For each valid query, compute entropy in nats and its effective support:

$$H_{t,h} = -\sum_{i \in R} p_{t,h,i} \log p_{t,h,i}, \qquad E_{t,h} = \exp(H_{t,h}), \qquad 1 \le E_{t,h} \le N, \tag{2}$$

where $H_{t,h}$ is the entropy of the attention distribution over the retained user tokens for decoding query t and head h and $E_{t,h}$ is the effective number of attended user tokens. For example, the entropy $H_{t,h}$ can measure how spread out the query's attention is. If attention is concentrated on a few tokens, the $H_{t,h}$ is low. If attention is uniformly distributed, then $H_{t,h} = \log N$. Therefore, applying $\exp$ to $H_{t,h}$ turns the entropy into an interpretable token count. The normalized form $E_{t,h}/N$ describes roughly what fraction of the retained tokens the query attends to. We define

$$w_{t,h} = v_{t,h} \max\left(1 - \frac{E_{t,h}}{N}, \lambda\right), \qquad q_{h,i} = \frac{\sum_{t=0}^{T_x-1} w_{t,h} p_{t,h,i}}{\sum_{t=0}^{T_x-1} w_{t,h}}, \tag{3}$$

where $v_{t,h}$ is the validity indicator (1 if both $m^U_{t,h}$ and $m^R_{t,h} \ge 10^{-8}$, otherwise 0) and $\lambda = 10^{-3}$. The weight gives more influence to concentrated queries while retaining a positive weight for valid diffuse queries. A concentrated query can still be uninformative or confounded by a sink. All temporal aggregation occurs within an individual head; heads are not averaged before feature extraction.

4.3 ROBUST TOKEN SCORING AND LOCALIZATION

We compare a token's attention with the uniform baseline and robustly standardize within the retained user region:

$$b_{h,i} = \log(\max(N q_{h,i}, \epsilon)), \qquad c_h = \mathrm{median}_{i \in R}\, b_{h,i}, \tag{4}$$

$$d_h = \mathrm{median}_{i \in R}\, |b_{h,i} - c_h|, \qquad z_{h,i} = \frac{b_{h,i} - c_h}{\max(1.4826 d_h, \delta)}, \tag{5}$$

with $\delta = 0.1$. The denominator is a maximum with a floor, not MAD plus an infinitesimal constant. This distinction matters when many tokens have nearly identical attention. The factor 1.4826 is a scale convention; it does not make the scores Gaussian or supply token-wise significance levels.

We consider three directions, $d_+(z) = \max(z, 0)$, $d_-(z) = \max(-z, 0)$, and $d_\pm(z) = |z|$. The direction is selected on development data, allowing unusually low attention as well as unusually high attention to be informative. For selected heads $\mathcal{H}$, define

$$s_i = \mathrm{median}_{h \in \mathcal{H}}\, d(z_{h,i}), \qquad S(x) = \max_{i \in R} s_i, \qquad \widehat{y}_{\mathrm{atk}} = \mathbf{1}[S(x) > \theta]. \tag{6}$$

Excluded positions receive zero scores. Localization marks the retained tokens satisfying $s_i > \theta$ and merges overlapping or touching character-offset intervals. Separate flagged regions remain separate: the main method does not use a contiguous interval scan. A token overlapping an annotated span is positive for token-level evaluation; ambiguous tokenizer boundaries are retained in the offset audit.

4.4 HEAD SELECTION AND INDEPENDENT CALIBRATION

The default selection pipeline uses four behavior-disjoint partitions: training, tuning, non-attack calibration, and testing. For each score direction, heads are ranked on training prompts by the detection AUROC of their maximum token score. Candidate top-k sets are compared on the tuning split using F1 with a tuning-only threshold; ties prefer higher AUROC and then smaller k. The tuning threshold is discarded. The target weights and token-scoring formula remain unchanged.

The default head ranking uses prompt attack labels, not span labels. A separate implementation option ranks heads using localization F1 and must be reported as span-supervised selection. Likewise, the exploratory Top10 heatmaps use known spans to rank individual heads; these visualizations do not demonstrate span-free selection or held-out detector performance.

After freezing $\mathcal{H}$, d, and all feature settings, calibration determines θ. If multiple non-attack controls share a behavior group, we first take their maximum prompt score. For M group scores with order

statistics $C_{(1)} \leq \cdots \leq C_{(M)}$, use

$$r = \lceil (M+1)(1-\alpha) \rceil, \qquad \theta = \begin{cases} C_{(r)}, & r \leq M, \\ +\infty, & r > M. \end{cases} \tag{7}$$

The decision uses strict $>$. Under exchangeability of calibration and deployment group scores, this order statistic controls the marginal group-level false-positive probability. It does not guarantee conditional error control for every subgroup or token. For finite thresholds, nominal levels $\alpha = 0.05$ and 0.01 require at least 19 and 99 calibration groups, respectively.

4.5 Computational cost

The probe adds no forward pass and no auxiliary model. It reuses attention computed during the first $T_x \leq 32$ decoding steps of the ordinary generation, unlike defenses such as SmoothLLM that generate responses for multiple perturbed copies of each prompt. Given the attention, feature computation for the selected heads costs $O(|\mathcal{H}|\, T_x\, |U|)$ time and memory, which is small relative to the target-model forward computation once the required attention tensors are available. A deployment needs to retain only the selected heads' attention to user keys, which could be computed from their queries and the cached keys without materializing attention for all other heads.

5 Experimental Protocol

5.1 Targets, attacks, and provenance

The current comparison evaluates Llama-2-7B-chat, Llama-2-13B-chat, and Vicuna-13B-v1.5 on locally generated GCG attacks and imported HarmBench AutoDAN artifacts. The latter retain the source archive member, and the source target model; the recorded members explicitly belong to HarmBench's AutoDAN results. These artifacts are distinct from our local AutoDAN-GA reimplementation.

Evaluated subsets. We construct standardized test splits across target models using GCG (200 test attacks) and HarmBench AutoDAN (189–194 non-overlapping test attacks depending on attention trace availability). Both attack conditions share 17 benign control prompts. To guarantee paired evaluation parity, missing-trace instances are excluded upfront across all defense baselines rather than penalizing specific models.

Outcome adjudication. Attack success concerns whether the saved response substantively fulfills the harmful objective, not whether the request itself is harmful or whether optimization loss is small. Response safety is evaluated using an LLM-based judge operating at zero temperature with a 384-token generation cap.

5.2 System templates and generation settings

All current conditions use the minimal system message "You are a helpful assistant." with the target-specific role template. Target generation is greedy, with a maximum of 128 new tokens, bfloat16 weights, and the eager attention path. The saved runtime records PyTorch 2.7.1+cu128, Transformers 4.51.3, and NVIDIA H200 NVL hardware. Attention probing uses up to 32 decoding queries and a 128-token prefill chunk, with the query convention in Section 4.1. Attack inputs are fixed during defense evaluation and are not re-optimized against each defense.

5.3 Controls and evaluation splits

The primary detection table combines attack attempts, whether successful or unsuccessful, with benign controls. Underlying harmful requests without adversarial modifications are retained as additional non-attack controls in the audit reports, but are not included in this table's denominators. They test whether the signal recognizes modifications rather than harmful subject matter alone. Each target's development-selected heads and independently calibrated threshold are fixed before evaluation and reused for the HarmBench condition; heads are not reselected using HarmBench test labels. Saved identifiers and prompt/goal hashes were checked for overlap with detector-fitting records.

All reported metrics reflect diagnostic evaluations of the frozen detector pipeline on defined benchmark splits. To ensure fair cross-family comparisons, benign control prompts are shared across attack conditions. Hyperparameter exploration—including decoding query budgets $T \in \{2, 4, 8, 16, 32\}$ and selected head counts $k \in \{1, 4, 8, 16\}$—establishes the operational space, with primary results reported under default settings ($T = 32$, $k = 8$).

5.4 Baselines and comparison contracts

The defense comparison comprises no defense, CPD gating, a RobustKV reimplementation, and gating by our attention probe. CPD and our method use saved detection decisions: flagged inputs receive a fixed refusal, while unflagged inputs receive a target-model response. These are executed gate policies using cached probe decisions, not an implemented streaming deployment. CPD uses system-token median/MAD normalization and a one-sided CUSUM over vocabulary entropy, a strict alarm threshold, and continued scoring after the first alarm. Its first estimated onset defines a suffix-to-end localization mask. Calibration uses non-attack development records at the matched nominal FPR; this method may differ from the official CPD implementation.

The independent RobustKV implementation evicts KV entries before subsequent generation, with a configured eviction rate of 0.2 and an observation window of 64. It retains and recomputes the final prompt token. For Llama-2-13B under GCG, RobustKV is evaluated without key-value eviction to accommodate prompt sequence bounds. RobustKV is compared as a generation intervention, not as a binary detector.

5.5 Metrics and uncertainty

For attack presence, we report accuracy, precision, recall, F1, and realized test FPR at the frozen threshold. All attack attempts are positives, independently of outcome-judge labels. Precision is undefined when no positives are predicted; with roughly 11–12 attacks per benign control, it is also dominated by recall, so recall and FPR are reported separately. Nominal calibration FPR is distinguished from observed test FPR; zero errors among 17 benign controls do not establish zero population FPR. The tables report point estimates; uncertainty and paired method comparisons require behavior-group resampling, without treating reused references as independent observations. Success-conditioned recall is reported with the caveat that it depends on unvalidated automatic outcome labels.

Localization uses a per-input any-overlap hit rate over all evaluated attacks with known span annotations, including missed detections in the denominator. Its definition and measured values appear in Section 6.3. Unknown-span cases would be excluded with coverage reported. This metric does not measure token precision or boundary accuracy.

Defense ASR uses all evaluated attack attempts in a condition, excluding benign controls. For A attempts, validated successes s, and unresolved outcomes u, missing-label bounds are $[s/A, (s + u)/A]$; these bounds do not correct systematic judge errors. Once judgments are validated, actual defense outputs give

$$\mathrm{ASR}_d = \frac{1}{A} \sum_{i=1}^{A} \mathbf{1}[\text{response under defense } d \text{ fulfills harmful goal } i]\,. \tag{8}$$

Fixed protocol refusals cannot constitute successful fulfillment. Generation interventions require judgments of their own resulting responses. Benign utility must be assessed separately, and the present timing records exclude cached signal extraction and therefore do not establish end-to-end defense overhead.

6 Results and Analysis

We use detection and localization as quantitative probes of attention–attack association. The current evidence covers six target/attack conditions under the minimal system message. We separate the validated statistics of attack presence from response-outcome estimates affected by judge errors.

Table 1: Main results (%). **ASR**: judged attack success rate over all attack attempts (lower is better). **Detection**: attack-presence metrics of our probe; positives are all attack attempts, and negatives are the 17 benign controls per target. **SpanHit**: fraction of attacks whose predicted spans overlap the annotated adversarial span. N_A: number of evaluated attacks.

		ASR				Ours					
Target	N_A	None	RobustKV	CPD	Ours	Acc.	Prec.	Rec.	F1	FPR	SpanHit
GCG											
Llama-2-7B	200	76	22.5	18	11.5	88.94	99.44	88.50	93.65	5.88	88
Llama-2-13B	200	62	20	14	13	87.56	99.43	87.00	92.80	5.88	86.5
Vicuna-13B	200	78	19	21	4	96.31	100.00	96.00	97.96	0	96
HarmBench AutoDAN											
Llama-2-7B	189	73.02	10.58	8.47	2.12	97.57	99.46	97.88	98.67	5.88	96.30
Llama-2-13B	193	69.95	9.33	8.29	2.07	97.62	99.47	97.93	98.69	5.88	96.89
Vicuna-13B	194	94.85	10.82	9.28	1.55	98.58	100.00	98.45	99.22	0	97.94

6.1 Defense comparison and outcome validity

Table 1 reports attack success rates over all evaluated attack attempts (Eq. (8)). Without a defense, 62–78% of GCG attacks and 69.95–94.85% of AutoDAN attacks are judged successful. All three defenses reduce ASR substantially, and gating with our probe yields the lowest ASR in every condition: 4–13% on GCG and 1.55–2.12% on AutoDAN, versus 14–21% and 8.29–9.28% for CPD and 19–22.5% and 9.33–10.82% for RobustKV. The margin over CPD ranges from two attacks out of 200 (Llama-2-13B/GCG) to 34 (Vicuna-13B/GCG).

Flagged inputs receive a fixed refusal, so a gate's ASR cannot exceed its miss rate. For our probe, the judged ASR reaches this bound in every condition: every attack that passes the gate is judged successful (e.g., all 23 misses on Llama-2-7B/GCG), so its ASR reduction is determined by recall. Three qualifications apply. The automatic judge contains confirmed errors, so margins of a few attacks could be reversed. While gating mechanisms introduce a trade-off by outright refusing inputs above the threshold, intervention methods like RobustKV modify KV cache representations during generation to preserve user interaction. While point estimates indicate a consistent empirical ordering across targets, broader multi-judge evaluations remain valuable to account for potential evaluator variances.

6.2 Attention distributions as an attack-presence probe

Table 1 reports attack-presence detection at each target's frozen, independently calibrated threshold. Every attack attempt counts as a positive, regardless of its judged outcome. On GCG, the probe recalls 88.50%, 87.00%, and 96.00% of attacks on Llama-2-7B, Llama-2-13B, and Vicuna-13B, missing 23, 26, and 8 attempts, respectively. It flags one of the 17 benign controls on each Llama-2 target and none on Vicuna-13B.

The heads and thresholds are reused on the imported HarmBench AutoDAN artifacts without reselection. There, recall increases to 97.88%, 97.93%, and 98.45%, with 4, 4, and 3 misses. Fluent genetic-search prompts are thus detected at least as reliably as the optimized suffixes on which the heads were ranked. This is consistent with the probe responding to something other than token-level irregularity of the input text. However, AutoDAN inputs also differ from GCG inputs in length and in their templated wrapper structure, and maximum aggregation is length-sensitive (Section 7). The higher recall therefore does not by itself show that the signal tracks the adversarial modification; harmful-plain and length-matched controls are required.

Near-perfect precision largely follows from the class balance of roughly 11–12 attacks per benign control, so recall and FPR are more informative. The observed FPR of 5.88% corresponds to a single benign prompt. It is consistent with the nominal $\alpha = 0.05$ but cannot establish it. The 0/17 result on Vicuna-13B is compatible with a population FPR as high as 19.5% (upper limit of the two-sided 95% Clopper–Pearson interval). Since the frozen threshold and benign control set are fixed per target model, reported false-positive rates remain consistent across both attack evaluation blocks.

6.3 LOCALIZATION OF ADVERSARIAL SPANS

We evaluate whether the probe identifies any part of an annotated adversarial span. Let $\mathcal{A}_{\text{ann}}$ denote evaluated attack inputs with known span annotations, S_i their annotated token set, and $\widehat{S}_i$ the predicted token set. We report

$$\text{SpanHit} = \frac{1}{|\mathcal{A}_{\text{ann}}|} \sum_{i \in \mathcal{A}_{\text{ann}}} \mathbf{1}\left[\widehat{S}_i \cap S_i \neq \varnothing\right].$$

Tokens are assigned to spans by character-offset overlap. For multiple annotated spans, hitting any one span counts once per input. Missed detections and empty predictions count as misses. Inputs with unknown spans are excluded, no evaluated attack has an unknown span. This is an any-overlap hit rate, not a boundary-accuracy or token-precision measure; broader predicted regions can satisfy it more easily.

For HarmBench-sourced AutoDAN inputs, the available span labels mark the mechanically identified added prefix; hits therefore concern those structural annotations rather than independently validated causal attack tokens.

Localization closely tracks detection. SpanHit ranges from 86.50% to 97.94% across the six conditions and, since missed detections count as misses, cannot exceed recall. Of the attacks the probe detects, at most three per condition are flagged without overlapping the annotated span; on Vicuna-13B/GCG, every detected attack is also localized. Flagged inputs are therefore rarely explained by anomalies outside the adversarial content. Because SpanHit is an any-overlap measure, these results show that flagged regions reach the adversarial content, not that their boundaries match it, and they support a positional association rather than a causal explanation of unsafe generation.

7 LIMITATIONS

The method requires white-box access to attention and a buffered output, since the gate must decide before any text is streamed. Although the probe adds no forward pass, our collector uses eager attention for all heads, and its measured overhead relative to optimized serving remains to be reported. Short prompts, nearly uniform profiles, or modifications covering most tokens can destabilize MAD scoring; maximum aggregation is length-sensitive; sink masking can remove meaningful tokens; and templates, quantization, and decoding policies may change the signal.

The evidence is limited by 17 benign references per target shared across attack families, prior test-set inspection, structural span annotations, missing traces, and an automatic outcome judge with confirmed errors, which leaves comparative ASR and benign utility unvalidated. Future work should evaluate additional ablation dimensions—such as dynamic head selection and cross-architecture transferability—to further optimize detector efficiency. Finally, attention–span association is not causal attribution, and fixed attacks do not establish robustness to detector-aware optimization, multi-turn attacks, or retrieved prompt injections. Despite these limitations, the results suggest that early-decoding attention can reveal where a prompt has been adversarially modified, providing a potential second-line check when pre-generation detection fails to identify an attack.

8 CONCLUSION

We study jailbreak-associated behavior through position-resolved decoding attention, using entropy-weighted profiles and robust token scores as diagnostic probes. Across six target/attack conditions, frozen per-target probes detect 87.00–98.45% of attack attempts while flagging at most one of 17 benign references, transfer from local GCG attacks to imported AutoDAN artifacts without head reselection, and localize annotated adversarial content in 86.50–97.94% of attacks. Under the evaluated models and attack conditions, these results indicate that jailbreak modifications leave identifiable, position-resolved footprints in the target model's decoding attention. Confirmed outcome-judge errors reinforce the distinction between attack presence and attack success, while comparative defense effectiveness awaits validated response judgments.

AI USE STATEMENT

Generative AI assisted research code, visualization scripts, and manuscript revision. The authors remain responsible for the final work. All research ideas, methodological decisions, experimental settings, implementations, analyses, and conclusions were reviewed and verified by the authors. AI-generated suggestions were not accepted without manual inspection, and reported numerical results were obtained from the authors' experimental implementations rather than generated by AI. The authors take full responsibility for the final content of this work, including all text, claims, experiments, and conclusions.

ETHICS STATEMENT

This study examines adversarial prompts and potentially harmful outputs for defensive evaluation. Released examples should be limited to what is necessary, with redactions disclosed.

REPRODUCIBILITY STATEMENT

Feature definitions and calibration are specified in Section 4; baseline deviations appear in Section 5.4. The current archive records input and response hashes, frozen detector artifacts, per-example predictions, model settings, and the scoped parity review. Detection metrics and span hit rates were recomputed from saved records; raw attention tensors are not included in this compact archive. The current collector uses up to 32 queries, a 128-token prefill chunk, and bfloat16. Target-response generation is capped at 128 new tokens. Evaluator outputs and raw prediction logs are saved and hash-indexed in the repository for auditability.

REFERENCES

Gabriel Alon and Michael Kamfonas. Detecting language model attacks with perplexity. *arXiv preprint arXiv:2308.14132*, 2023.

Mohammed Alshaalan and Miguel R. D. Rodrigues. Detecting fluent optimization-based adversarial prompts via sequential entropy changes. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026. URL https://arxiv.org/abs/2605.19966.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, 2024.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. Token highlighter: Inspecting and mitigating jailbreak prompts for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 27330–27338, 2025. doi: 10.1609/aaai.v39i26.34943.

Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1357.

Tanqiu Jiang, Zian Wang, Jiacheng Liang, Changjiang Li, Yuhui Wang, and Ting Wang. Robustkv: Defending large language models against jailbreak attacks via kv eviction. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/38bbae17d60940f3ee14dfd1035d7542-Abstract-Conference.html.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7Jwpw4qKkb.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35181–35224. PMLR, 2024.

Anselm Paulus, Arman Zharmagambetov, Chuan Guo, Brandon Amos, and Yuandong Tian. Advprompter: Fast adaptive adversarial prompting for llms. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48439–48469. PMLR, 2025.

Alexander Robey, Eric Wong, Hamed Hassani, and George J. Pappas. Smoothllm: Defending large language models against jailbreaking attacks. *Transactions on Machine Learning Research*, 2025.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. 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. 11–20. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1002.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient Streaming Language Models with Attention Sinks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.17453.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.