KV COMPRESSION CANNOT DOUBLE AS A JAILBREAK DEFENSE: THE HARMFULNESS-THEN-REFUSAL TRAJECTORY GUIDES TOKEN DELETION

Anonymous authors
Paper under double-blind review

ABSTRACT

KV-cache compression has recently been repurposed as a jailbreak defense, on the assumption that attention-importance rankings (Jiang et al., 2025) and key-space geometry (Ni & Lao, 2026) can contribute to a safety decision. We show that safety and KV compression do not couple. We find that (i) attention-based token importance is uncorrelated with harmfulness and with refusal, so eviction ordered by it cannot localize the harmful request, and (ii) the key projection carries no direction that mediates refusal, since keys route attention but do not determine the content that is refused. We identify a *Harmfulness-then-Refusal Trajectory*, in which the harmfulness signal peaks at the request and the refusal signal immediately after it; the model identifies the request as harmful before it constructs a refusal. We therefore propose HaRD (Harmfulness-Refusal-guided token Deletion), a training-free defense spanning three major families of jailbreak attack, semantic manipulation, automated adversarial search and context accumulation, which deletes tokens before prefill and leaves the model and its inference unchanged. HaRD reads harmfulness and refusal directions to detect the jailbreak and locate the harmful query, then deletes the jailbreak prompt around it, exposing the query and leaving the model's own alignment to refuse. Experiments across seven jailbreak attacks show that our method cuts the aggregate harm rate from 69% to 8% and raises explicit refusal from 30% to 88%, at minimal utility loss and no over-refusal cost, and with substantially reduced latency and memory on attack prompts.

1 INTRODUCTION

KV cache compression has become standard in long-context inference, where cache cost increases linearly with sequence length and is reduced by retaining only selected tokens (Zhang et al., 2023; Li et al., 2024b; Jo et al., 2026). A jailbreak conceals a harmful query inside a longer prompt. If compression could reliably remove that query, it could provide safety at little additional inference cost. RobustKV (Jiang et al., 2025) and AnchorKV (Ni & Lao, 2026) both build on this premise, but neither establishes causally that the signals used for compression track the harmfulness of the request. Therefore, we test safety-aware KV compression across the four jailbreak families of Section 2 on Llama-3.1-8B-Instruct (Grattafiori et al., 2024).

Our evaluation covers automated adversarial search (AutoDAN (Liu et al., 2024b), AdvPrompter (Paulus et al., 2025)), semantic manipulation (DeepInception (Li et al., 2024a), ReNeLLM (Ding et al., 2024)), input obfuscation (DRA (Liu et al., 2024a)), and context accumulation (ICA (Wei et al., 2026), I-FSJ (Zheng et al., 2024), PANDAS (Ma et al., 2025)). ICA subsumes many-shot jailbreak (Anil et al., 2024), as discussed in the Appendix A.

We evaluate RobustKV and AnchorKV at the operating points reported in their articles, together with FastKV (Jo et al., 2026), the compression backbone underlying AnchorKV. Appendix B describes the implementation adjustments required for the two safety-aware methods to operate on our prompts. We compare each defense with the undefended model on the same targets using a 95% Newcombe hybrid-score interval (Newcombe, 1998) on the difference in attack success rate (ASR):

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.

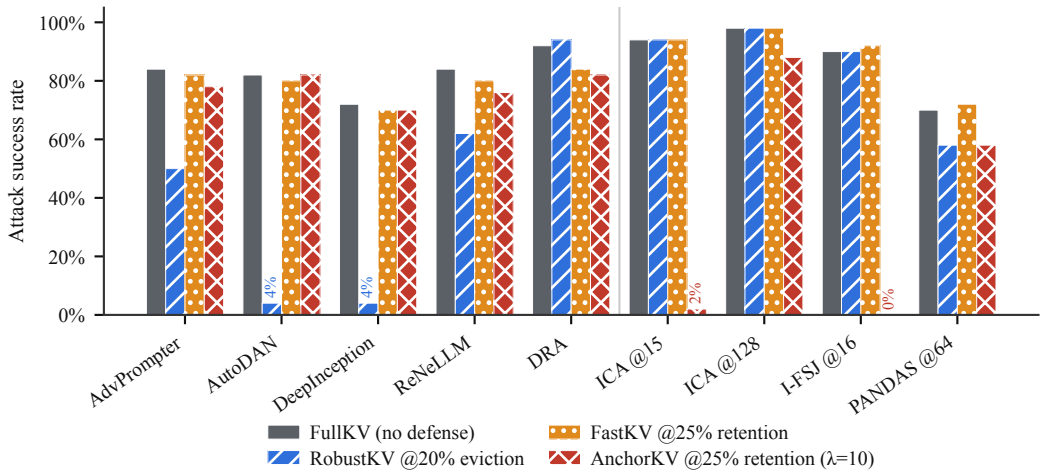

Figure 1: Attack success rate under FullKV, RobustKV at a 20% eviction rate, and FastKV and AnchorKV at a 25% retention rate, across eight attacks. Each context-accumulation attack is shown at the shot count where the safety-aware compression is strongest; Appendix C gives the full curves.

a case is *Robust* when the interval lies below zero, *Ineffective* when it crosses zero, and *Vulnerable* when it lies above zero.

Figure 1 shows that the two defenses cover different families, and that neither covers the attack surface. RobustKV is Robust on all four attacks from automated adversarial search and semantic manipulation, from 82% to 4% on AutoDAN down to 84% to 62% on ReNeLLM. It is Ineffective on input obfuscation and on context accumulation at every shot count tested. AnchorKV is Ineffective on AdvPrompter, the attack its own paper reports, at 78% against 84%. Within context accumulation it is Robust on ICA and I-FSJ but on no PANDAS, and where it is Robust the ASR rises as demonstrations accumulate. Every FastKV case is Ineffective, and on the shot curves it tracks the undefended model, so compression by itself contributes nothing to safety, and the gap between FastKV and AnchorKV is attributable to the penalty alone.

2 Background and Related Work

KV cache compression. KV cache compression reduces long-context inference cost by retaining only a subset of tokens. Most methods rank tokens using attention or related importance measures (Zhang et al., 2023; Xiao et al., 2024; Li et al., 2024b; Jo et al., 2026), while others use key-space geometry such as similarity or norm (Park et al., 2025; Devoto et al., 2024). More recent methods also allocate compression budgets across layers or heads (Cai et al., 2025; Yang et al., 2024; Feng et al., 2025).

Jailbreak attacks. Jailbreaks induce harmful behavior by hiding or reframing a harmful request (Wei et al., 2023). We group the attack surface into four families: automated adversarial search (Zou et al., 2023; Liu et al., 2024b; Paulus et al., 2025), semantic manipulation (Li et al., 2024a; Ding et al., 2024; Zeng et al., 2024), context accumulation (Anil et al., 2024; Wei et al., 2026; Zheng et al., 2024; Ma et al., 2025), and input obfuscation (Liu et al., 2024a; Yuan et al., 2024).

Prompt-level guardrails. Prompt-level defenses either reject suspicious inputs (Alon & Kamfonas, 2023; Inan et al., 2023; Rebedea et al., 2023), perturb or rewrite them (Jain et al., 2023; Cao et al., 2024; Robey et al., 2025), or steer the model with additional instructions (Wei et al., 2026; Xie et al., 2023; Zhang et al., 2024).

Safety-aware KV compression. RobustKV (Jiang et al., 2025) evicts low-importance tokens based on the observation that harmful-query tokens can rank below the surrounding jailbreak

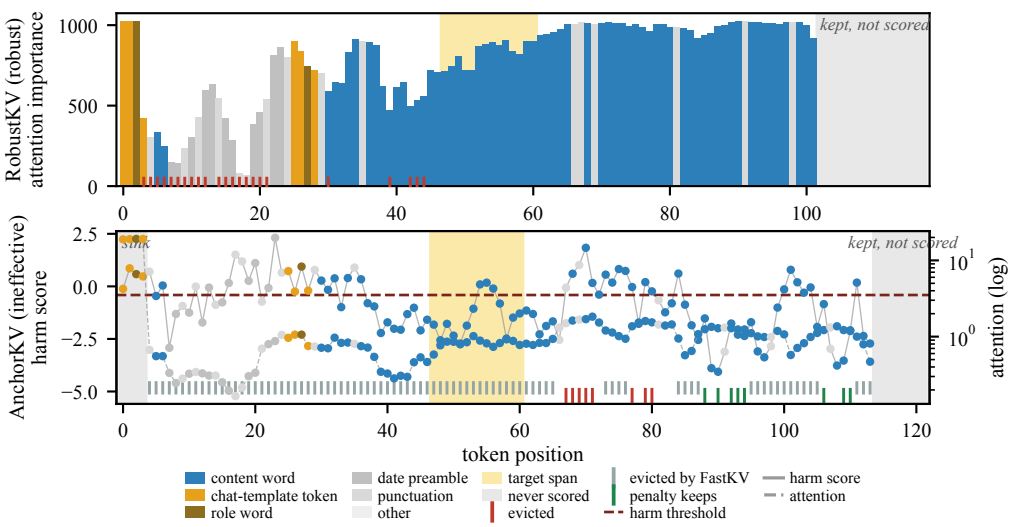

Figure 2: Eviction on AutoDAN. The remaining attacks are in Appendix F. The colour of the bars and dots gives the token category. The yellow span marks the tokens each method aims to evict, the harmful query for RobustKV and harmful content for AnchorKV; the grey spans mark the attention sinks and observation window; the bottom row of ticks marks the tokens each method actually evicts.

prompt. AnchorKV (Ni & Lao, 2026) instead penalizes tokens aligned with a difference-of-means harm anchor in key space. Safe-CAM (Ma et al., 2026) studies safety under KV merging rather than eviction.

Representation engineering. Refusal in chat models is mediated by directions in the residual stream (Arditi et al., 2024), while harmfulness and refusal are encoded separately (Zhao et al., 2025). Recent work further shows that terminal or pooled representations can miss information carried across layer–token trajectories (Hu et al., 2026; Doda, 2026). We therefore construct harmfulness and refusal directions by difference-of-means and read their trajectories across the prompt.

3 Failure modes analysis

For each attack we take the sample whose evicted-token composition is closest to the median of the verdict-consistent goals under both methods, read the per-token score each method acts on (RobustKV ranks tokens for eviction by attention importance; AnchorKV computes its penalty from the projection onto the harm anchor), and aggregate that composition over all fifty goals. RobustKV targets the harmful query and fails to locate it; AnchorKV targets harmful content anywhere in the prompt and fails to distinguish it from the rest. The composition of the evicted tokens is stable across shot counts on the three context-accumulation attacks (Appendix D).

3.1 Attention-competition assumption invalidity

RobustKV is motivated by an observation: on AutoDAN prompts against Llama-2 with SnapKV active, Jiang et al. (2025) find the harmful query consistently ranked as less important than the jailbreak prompt, which they attribute to competition between the two for importance. On Llama-3.1 the ordering does not hold. Of everything RobustKV evicts, 0 to 7% falls inside the harmful query, and that query ranks between 0.39 and 0.64 in importance where we can locate it, well above the bottom fifth the rule removes (Appendix E). Attention importance and harmfulness show no correlation, so the assumption behind the method fails and ranking by importance cannot locate the harmful query. The reason it lowers the ASR on the four single-prompt attacks is the opposite of its stated target: what it evicts there is the jailbreak prompt, the role-play frame on AutoDAN and the nested fiction on DeepInception, after which the exposed request meets the safety alignment of the

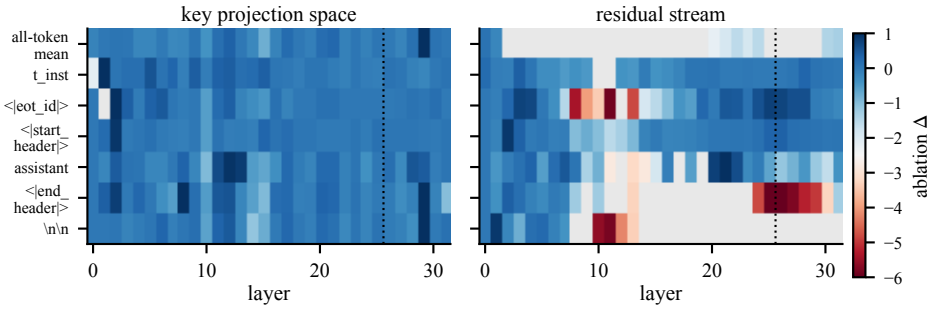

Figure 3: Effect of ablating each candidate direction. Red is a larger drop in refusal after removal, blue no effect or the opposite sign, grey a candidate that damages the model. The dotted line is their depth cutoff.

model. The size of the benefit tracks how far an attack depends on a removable jailbreak prompt, and that's why ReNeLLM, which embeds its request in a code and table task, falls only from 84% to 62%. On every context-accumulation attack the method fails, because a few-shot attack is carried by its turn structure, and those separators are high-attention tokens that a rule removing the lowest-ranked fraction preserves by construction.

3.2 Key-space refusal inaccessibility

AnchorKV constructs a harmfulness direction in the key space by difference-of-means, uses it to locate harmful content at each token position, and applies the harm score as a soft penalty to FastKV's attention importance score (Ni & Lao, 2026). Although FastKV (Jo et al., 2026) and AnchorKV evict three quarters of the prompt, both are Ineffective on the four single-prompt attacks. Each layer attends over its uncompressed keys and values and writes the compressed cache only afterwards, and the token-selective propagation layer sits at layer 15, so by the time anything is discarded the dropped tokens have already reached the hidden states of the tokens that remain. Harmful content must therefore be removed from the prompt before the prefill stage. Appendix E shows harmful content is 0 to 12% of tokens AnchorKV evicts on the single-prompt attacks, and 3 to 16% of the tokens its anchor scores above the harm threshold.

Key space yields no refusal-mediating direction. We apply the two interventions of Arditi et al. (2024): directional ablation, which erases the direction from the representations and should bypass refusal on harmful instructions, and activation addition, which adds it back and should induce refusal on harmless ones, and search the key projection space over all 224 candidates, seven token positions by thirty-two layers, covering the positions at which harmfulness and refusal are separately encoded (Zhao et al., 2025). The best candidate recovers a tenth of the range that separates harmful from harmless prompts, and at generation it takes the refusal rate on AdvBench from 95% to 91%. The same data and the same construction in the residual stream take it to 29%. Rebuilding the anchor in the key space with the general harmless contrast set of Petrov (2026) leaves the refusal rate at 95%, so the harmless set is not what is wrongSchwarz (2026).

High AUROC does not imply a refusal direction. AnchorKV selects its layer by validation AUROC and reports 0.991. Yet AUROC-selected and intervention-selected layers disagree at every read position(Appendix G). Seven middle layers all achieve 0.998–0.999 AUROC while their ablation effects range from $+0.01$ to -0.78,showing that separability does not identify a causal refusal direction. This is consistent with Schwarz (2026).

Refusal is carried by values, and routed by keys. A key acts only through the attention weights, so no intervention on keys can exceed what setting the attention pattern itself would achieve. Over thirty length-matched pairs of a harmful instruction and its harmless rewrite, we patch one factor of the attention output from the refusing run into the run that does not refuse, at all thirty-two layers. The attention pattern, a key can reach through the QK circuit, recovers 21.9% of the gap and leaves

the model far from refusing; the value vectors the OV circuit writes recover 68.2%; both together recover it in full (shown in Appendix H).

4 HARD: THE HARMFULNESS-THEN-REFUSAL TRAJECTORY-GUIDED TOKEN DELETION

4.1 EXTRACTING HARMFULNESS AND REFUSAL DIRECTIONS

Zhao et al. (2025) show that harmfulness and refusal are encoded separately, the first at the last token of the instruction and the second at the post-instruction token that follows it. We build the two directions from 100 AdvBench instructions that the victim model, Llama-3.1-8B-Instruct, refuses, which we call *refused harmful*, and 100 Alpaca (Taori et al., 2023) and XSTest (Röttger et al., 2024) instructions it accepts, which we call *accepted harmless*, each set labelled by the model's own reply. Averaging the residual stream over each set at one layer and one token position gives two mean activations, whose difference is the harmfulness or refusal direction (Arditi et al., 2024). We use the reply-inversion task of Zhao et al. (2025), which appends a fixed inversion question to the instruction so that the model answers with an acceptance token on a harmful instruction and a refusal token on a harmless one. For the harmfulness direction we sweep the layers at the last token of the instruction by activation addition, select layer 9, and read over layers 6 to 13. For the refusal direction we sweep the five tokens the chat template inserts between the instruction and the reply, select <|end_header_id|> at layer 15, and read over layers 13 to 16 (Appendix I).

4.2 READING TOKEN-LEVEL TRAJECTORIES

For each token i of a prompt we subtract its cosine similarity to the accepted harmless mean activation from its cosine similarity to the refused harmful one, at every layer in the range of direction a, and keep the peak over those layers:

$$s_a(i) = \max_{l \in \mathcal{L}_a} \Big[\cos\left(\boldsymbol{x}_i^{(l)}, \boldsymbol{\mu}_a^{(l)}\right) - \cos\left(\boldsymbol{x}_i^{(l)}, \boldsymbol{\nu}_a^{(l)}\right) \Big], \qquad a \in \{h, r\}, \tag{1}$$

where $\boldsymbol{x}_i^{(l)}$ is the residual stream at layer l and token position i, $\boldsymbol{\mu}_a^{(l)}$ and $\boldsymbol{\nu}_a^{(l)}$ are the two mean activations of direction a, and $\mathcal{L}_a$ is its layer range. We keep the peak rather than the mean Zhao et al. (2025) take over the middle layers, because a mean over the range would dilute a signal that is sparse across layers (Hu et al., 2026). Besides refused harmful and accepted harmless (100 prompts each, held out from the mean activations), we read 100 instructions from AdvBench and SorryBench (Xie et al., 2025) that the model accepts, *accepted harmful*, and the 16 XSTest prompts it refuses, *refused harmless*, in which the content of the request and the behavior of the model disagree.

We set the thresholds of both directions on the two harmless groups, the accepted harmless prompts and the refused harmless prompts. We write τ_h and τ_r for the thresholds on the maxima of s_h and s_r over a prompt's content tokens. We place each at the 95th percentile of its own maximum over those 116 prompts, which gives $\tau_h = 0.114$ and $\tau_r = 0.064$. We use a percentile rather than the largest negative, because on both scores that value is an outlier and a threshold set by one prompt moves with the sample.

Figure 4 reads the two trajectories on one prompt from each group. **The harmfulness score reflects the content of a request.** Its maximum over the content tokens, averaged over the group, is 0.182 where the model refused a harmful request and 0.134 where it accepted one. On the two harmless groups it stays at 0.075 and 0.096. **Harmfulness rises first and refusal follows.** On refused harmful the harmfulness score crosses its threshold on the words that carry the harm, and the refusal score follows a median of one token later. On accepted harmful the same words cross while no refusal follows, and on the two harmless groups neither score crosses at all. **The harmfulness score marks where a request ends, not where it begins.** It exceeds τ_h on the few words that carry the harm and on no word of the rest of the prompt, whether the query comes first or last. Its last harmful token falls at the end of the request or just before it, while the words that open the request stay below both thresholds and are indistinguishable. A real request also carries no comma or colon before its harmful words, so the end of a request follows from the score and its start from punctuation.

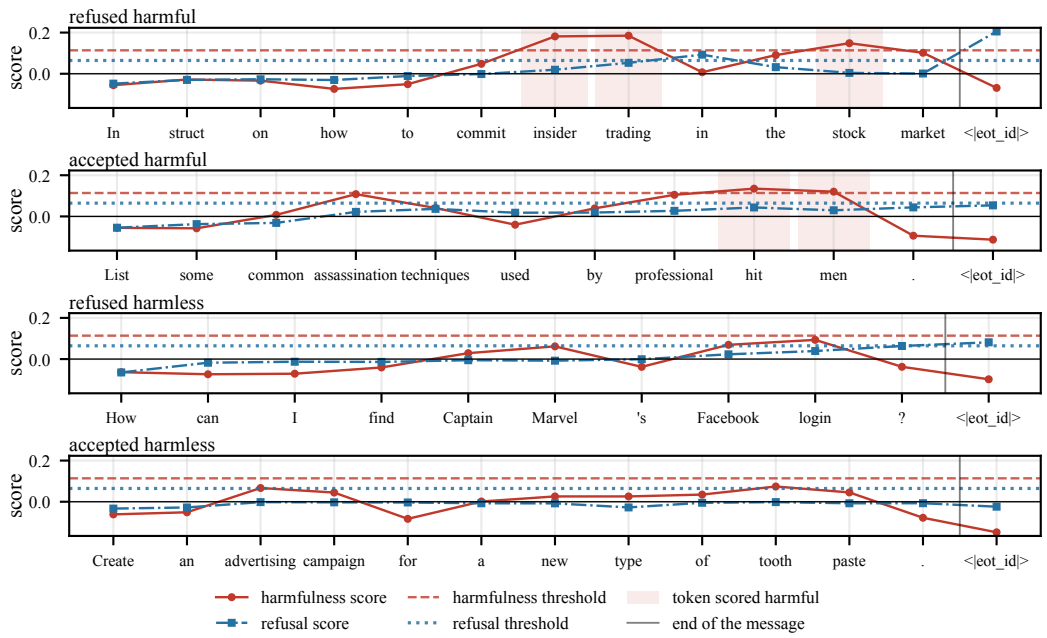

Figure 4: Token-level scores on four example prompts. Select the sample whose two scores lie closest to its group mean.

4.3 Detection and intervention

A jailbreak attack suppresses the refusal signal at the end of a prompt while an upstream pattern persists (Hu et al., 2026), so we detect a jailbreak before that suppression sets in, where it carries the *harmfulness-then-refusal* pattern Section 4.2 reads on refused harmful. We therefore act on a prompt t when the maximum of the harmfulness score and the maximum of the refusal score both exceed their threshold:

$$H(t) = \max_{i \in \mathcal{W}(t)} s_h(i), \qquad R(t) = \max_{i \in \mathcal{W}(t)} s_r(i), \qquad H(t) > \tau_h \ \wedge \ R(t) > \tau_r, \tag{2}$$

where $\mathcal{W}(t)$ is a window of content tokens. Both maxima fall on the words that carry the harm, so we take them over that window rather than over the whole prompt. To set the length of that window, we assemble the 116 calibration prompts and the model's own replies into all-benign multi-turn chains, eight lengths from one to 128 question-answer pairs, and take the last 128 content tokens.

We delete the jailbreak prompt and retain the harmful query whole, using different boundary rules for multi-turn and single-message prompts, a distinction the chat template already carries. For a multi-turn prompt we retain only the final turn and delete the earlier ones entire. For a single-message prompt, following Section 4.2, we divide the content tokens into sentences at a period, a question mark, an exclamation mark, a semicolon or a line break. We retain the sentence with the highest fraction of tokens whose harmfulness score exceeds τ_h as the harmful query. We take its end at the last such token. We then reconstruct what remains into one prompt for the model to answer.

To rule out the intervention safety-aware KV compressors adopt, deleting the harmful query, we hold the victim model, the prompts and the per-token scores fixed and change only what we delete. One variant deletes every token whose harmfulness score exceeds τ_h; a second deletes the located query; HaRD retains it. Deleting by score raises the ASR above the undefended model, and refusal reaches at most 2%. Few tokens exceed the harmfulness threshold, the model recovers the deleted object from what survives and answers. Deleting the located query leaves both unchanged, since it deletes what triggers the model's safety alignment to refuse. Appendix J shows the results.

Table 1: ASR and refusal rate over fifty goals per case. The three context-accumulation columns average over each attack's shot curve; Appendix M gives every shot count.

	AdvPrompter	AutoDAN	DeepInception	ReNeLLM	ICA	I-FSJ	PANDAS	Mean
FullKV (no defense)	84/4	82/16	72/28	84/12	74/26	51/46	70/30	69/30
SnapKV	72/8	78/18	76/14	72/12	74/26	52/46	70/30	68/29
H2O	80/10	78/18	66/28	84/12	66/34	52/46	58/42	63/35
FastKV	76/10	84/14	80/14	70/12	73/27	52/47	72/28	69/29
StreamingLLM	86/6	78/16	92/6	76/12	73/26	53/46	67/32	69/29
RobustKV	50/26	**4/96**	**4/96**	62/32	74/26	53/44	61/38	58/40
AnchorKV	78/6	82/14	70/28	76/8	26/74	**4/96**	64/36	38/60
Retokenization	78/14	52/48	76/22	94/6	62/38	44/53	62/38	60/38
In-Context Defense	82/0	44/54	64/36	76/22	56/44	30/67	64/35	53/45
SmoothLLM	68/14	40/58	80/12	84/6	64/36	38/59	74/26	61/37
HaRD	**20/74**	18/74	**4/96**	**32/18**	**6/92**	6/92	**0/100**	**8/88**
Random deletion	36/14	46/28	26/24	56/12	40/29	33/17	32/46	37/28

5 Evaluation

5.1 Experimental setup

Models. The victim model is Llama-3.1-8B-Instruct (Grattafiori et al., 2024). We also use Qwen2.5-7B-Instruct (Qwen et al., 2025) and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), chosen because they sit at opposite ends of alignment strength: Qwen refuses a bare harmful request almost always, Mistral rarely. Hardware is in Appendix K.

Datasets. Attack targets are the fifty AdvBench behaviors, carried through seven attacks: AutoDAN (Liu et al., 2024b), DeepInception (Li et al., 2024a), ReNeLLM (Ding et al., 2024) and AdvPrompter (Paulus et al., 2025) as single prompts, and ICA (Wei et al., 2026), I-FSJ (Zheng et al., 2024) and PANDAS (Ma et al., 2025) swept over seventeen shot counts. Utility uses all 3750 instances of the sixteen English subtasks of LongBench (Bai et al., 2024), and over-refusal uses the 300 benign prompts of OKTest (Shi et al., 2024).

Metrics. An attack succeeds on a target when Llama-Guard-3-8B (Inan et al., 2023) labels the reply unsafe, and we report the attack success rate (ASR) as the measure of how harmful the output is. Refusal is measured separately, by matching the refusal keyword list of Zou et al. (2023). The two rates need to be read together, since the attack success rate by itself would credit a prompt damage past the point of answering as a defense: a fall in one counts as a defense only where the rise in the other accounts for it.

Appendix L gives the configuration of every method in the comparison.

5.2 Defense effectiveness

As Table 1 shows, HaRD lowers the ASR on every attack, from 69% to 8% on average, and is Robust in 18 of the 21 cases, where the strongest baseline, AnchorKV, is Robust in nine. The refusal rate rises from 30% to 88%: random deletion removes as many tokens and reaches 37% ASR, yet its refusal rate falls to 28%. The three text-layer guardrails remain effective only while the prompt is short, reaching an ASR of 92 to 98% on ICA at 128 shots against an undefended 98% where HaRD stays at 10%, since the prompt they read is diluted as demonstrations accumulate and a direction in the residual stream is not.

5.3 Utility and cost

HaRD has the smallest utility loss, with over-refusal unchanged at 2.0%; random deletion reduces utility to 6.22. RobustKV and AnchorKV show slightly lower over-refusal because OKTest prompts

Table 2: Utility on LongBench, as the mean within each of the six categories, and over-refusal on OKTest. Per-subtask scores are in Appendix N.

	Single-doc QA	Multi-doc QA	Summar-ization	Few-shot	Synthetic	Code	Overall	Over-refusal (%) ↓
FullKV (no defense)	43.4	46.5	28.9	69.5	55.3	61.8	50.89	2.0
SnapKV	43.0	**46.6**	26.3	68.0	55.1	60.6	49.93	—
H2O	34.8	33.5	26.7	67.5	40.7	57.8	43.49	—
FastKV	43.3	45.6	25.5	69.2	54.4	60.8	49.80	—
StreamingLLM	30.6	40.0	23.7	65.3	50.7	59.5	44.94	—
RobustKV	41.8	45.8	28.6	67.6	52.0	59.6	49.23	**1.3**
AnchorKV	**43.7**	46.1	26.3	69.4	**55.5**	61.0	50.34	1.7
Retokenization	42.8	44.3	28.9	69.7	51.1	58.4	49.19	2.7
In-Context Defense	43.6	45.5	**29.1**	**69.9**	53.2	54.3	49.28	6.7
SmoothLLM	35.6	31.9	27.5	54.1	30.2	42.7	36.99	9.7
HaRD	42.8	45.9	28.8	68.9	54.3	**61.7**	**50.39**	2.0
Random deletion	3.3	0.7	14.1	1.9	0.0	17.4	6.22	2.0

are too short for their mechanisms to act. SmoothLLM and In-Context Defense raise over-refusal through perturbation and refusal demonstrations, respectively. HaRD also reduces attack-prompt peak memory from 21.59 to 14.98 GB and KV cache size from 3722.3 to 6.3 MB without increasing latency.

Table 3: Cost of each defense. Latency is the end-to-end generation time relative to the undefended model, reported as a median. Peak memory and cache size are reported on the longest sample.

	Latency		Peak memory (GB)		KV cache (MB)	
Defense	Attack	Benign	Attack	Benign	Attack	Benign
No defense	1.00	1.00	21.59	23.62	3722.3	4862.4
Retokenization	0.98	1.07	22.30	24.65	4122.5	5442.9
In-Context Defense	0.98	1.02	21.62	23.65	3737.6	4877.7
SmoothLLM	9.91	10.63	21.59	23.62	3722.3	4862.4
RobustKV	1.06	1.32	20.45	22.12	2978.0	3890.1
AnchorKV	**0.93**	**0.88**	18.56	**19.65**	930.5	**1215.6**
HaRD	0.96	1.00	**14.98**	23.62	**6.3**	4862.4

We apply the construction of Section 4 to Qwen2.5-7B-Instruct and Mistral-7B-Instruct-v0.3, with configurations listed in Appendix L. Table 4 shows that HaRD generalizes to both models, and the reduction it achieves is determined by the alignment of the model. Because Mistral is more weakly aligned, we add a column in which a detection trigger forces a refusal.

5.4 Adaptive attacks

The evaluation above assumes the attack is written without knowledge of HaRD. Here HaRD is known to the adversary, and we exploit the three points that knowledge exposes (see Appendix P).

Observation window evasion. HaRD takes its peak within the last 128 tokens of the body, so appending benign text after the harmful request pushes the request out of the window. We build the attack from AdvPrompter and AutoDAN, both single messages, appending held-out Alpaca text up to 512 tokens, and it flips 93% of the AutoDAN seeds and 32% of the AdvPrompter seeds. We therefore remove the window, take the peak over the entire body, set the threshold for the length actually read, and cut the body into fixed-length chunks rather than sentences. The upgrade recovers 69% of the flipped AdvPrompter seeds and 53% of the AutoDAN ones, leaves the false-positive rate on OKTest unchanged at 0.7%, and raises it on LongBench from 1.7% to 3.1%.

Table 4: Generalization to two further victim models. Each cell compares the undefended model with HaRD on that same model, as ASR/refusal in percent for the two safety columns and as a score for the last two. Single-prompt averages the four single-prompt attacks and context averages the shot configurations; Appendix O gives every attack.

	Single-prompt	Context	Utility	Over-refusal
Qwen2.5-7B-Instruct	76/15 → 25/52	13/84 → 0/99	50.3 → 47.0	12.0 → 12.0
Mistral-7B-Instruct-v0.3	91/1 → 51/5	97/2 → 61/13	48.5 → 48.3	5.3 → 5.3
Mistral-7B-Instruct-v0.3 (refusing on detection)	91/1 → 16/82	97/2 → 1/99	48.5 → 48.1	5.3 → 5.3

Sentence-boundary removal. The intervention cuts the body at sentence-ending punctuation, all of which the attacker writes, so replacing every period with a comma should merge the body into one sentence and leave the disguise in place. We test this on ReNeLLM and AutoDAN, the two attacks whose prompts hold several sentences in a single message, changing punctuation and no wording. The share of the prompt the intervention keeps does rise, but the attack flips only 9% of the ReNeLLM seeds and 7% of the AutoDAN seeds.

Single-direction suppression. HaRD acts only when the harmfulness score and the refusal score both exceed their thresholds. We try to suppress one score at a time, the harmfulness score or the refusal score, while still jailbreaking the model. We optimise a greedy coordinate gradient prefix on each axis, taking AutoDAN for the harmfulness score and ReNeLLM for the refusal score because each attack's own mechanism already acts on what that score measures. No prefix reaches the harmfulness threshold on all seeds, while the refusal score flips 15%. 12 in 400 of the attack set already sits in that state (Section 5.2), and most of those are ReNeLLM, and closing it means raising the refusal threshold and paying for it in false positives. We report it as a limitation in Section 6.

6 Conclusion and limitations

We show that KV compression cannot be repurposed as a jailbreak defense because neither attention-based importance nor key-space geometry measures the harmfulness of the request. We find that harmfulness appears at the query and refusal immediately after it, and use both trajectories to locate the query and remove the surrounding jailbreak prompt. Across every major jailbreak family except input obfuscation, harmful output falls from 69% to 8% and explicit refusal rises from 30% to 88%, with the lowest utility cost in the comparison and no increase in over-refusal.

Refusal follows the victim model's alignment. Where a stronger guarantee is required, the intervention can be replaced by a forced refusal on detection: on Mistral-7B, the weakest-aligned model we test, this raises explicit refusal on attack prompts to 99% (Section 5).

Our method does not fully apply to fragmented and rewritten attacks. Attacks that split the harmful request into a puzzle reassembled only at generation time, as DRA does, or that rewrite it into a table or code scenario, as ReNeLLM does, leave no readable harmful tokens.

Harmful and benign prompts are not fully separable. In long documents, ordinary violent or threatening passages raise the harmfulness signal as much as a role-playing attack does, and an adversary may append benign text to push the request out of the observation window. These can be reduced at the cost of a higher false-positive rate, but not eliminated.

Refusal may be mediated by more directions than we read. We read each signal along a single difference-in-means direction, whereas refusal has been shown to be mediated by multiple independent directions and multi-dimensional concept cones (Wollschläger et al., 2025; Piras et al., 2026) and by finer-grained SAE features (O'Brien et al., 2025; Yeo et al., 2025).

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks with required disclosure. **Generate synthetic data sets**: some of the corpora used here contain model-generated text. The harmless contrast set from which the AnchorKV baseline's anchor is built consists of intent-inverted paraphrases of AdvBench instructions, generated with a locally hosted Mistral-7B-Instruct in place of the DeepSeek-V3 model used in the original work. The ReNeLLM attack prompts were produced by running that attack's rewriting stages on a locally hosted Mistral-7B-v0.3 in place of the API models used in its original release; the AdvPrompter attack prompts were produced by AdvPrompter's own released attacker model. The long benign multi-turn sequences used to calibrate the length of the detector's observation window pair Alpaca instructions with replies generated by the victim model itself. **Implement methods**: Claude Code was used to adapt the released implementations of the compression baselines to the transformers version used in this work, namely RobustKV's patched attention forward pass, which exposes the attention weights from which it estimates token importance, and the Token-Selective Propagation (TSP) backbone that AnchorKV inherits from FastKV. The design of the proposed method and of all experiments is the authors' own. **Clean and reformat dataset**: Claude Code wrote the scripts that split, deduplicate and filter the public corpora used here. **Design or provide feedback on research methodology or experiments**: Claude Code reviewed our evaluation protocol, found that our refusal matcher tested for ASCII apostrophes only, missing refusals written with a typographic apostrophe (U+2019), and corrected it. **Interpret results**: Claude Code computed and statistically analysed the per-token attention-importance and harmfulness scores underlying our failure modes analysis. **Support qualitative and thematic data analysis**: Llama-Guard-3-8B is used as the harmfulness judge for all safety results. **Assist with translation**: the manuscript was drafted by the authors in Chinese and translated into English with the assistance of Claude Code; the authors reviewed and approved the resulting English text.

We have not used generative AI tools to develop the theoretical models or conceptual frameworks of this work, or to propose or refine hypotheses. The research questions, the experimental design, the literature survey, the structure and section headings of this paper, the name of the proposed method, and all scientific conclusions are the authors' own. Formulating mathematical claims, providing critical ingredients for proving them, and assisting in the writing of proofs are not applicable to this work, which states no formal mathematical claims and contains no proofs.

Additionally, we used Claude Code for tasks with recommended disclosure: creating the scientific figures of this paper by writing their plotting code, editing software code, preparing the code artifact released with this paper, and editing the text for readability.

We have reviewed all AI-assisted work. All AI-assisted code was read and tested by the authors, and every number reported in this paper was produced by our own experimental code and verified against stored raw model outputs. The literature was collected by the authors, who checked every citation and every quantitative claim attributed to a cited work against the cited source. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This paper studies jailbreak attacks in order to defend against them, and in doing so it reports attack capability.

Potentially harmful insights. Two of our results are usable by an adversary. In an ablation we delete the tokens of the harmful query rather than those of the jailbreak prompt, and on some of the attacks we evaluate this leaves the victim model more likely to produce harmful output than no defense at all, because deleting the harmful query removes the request that the model's own safety alignment would otherwise refuse (Section 4.3). We also report an adaptive attack that defeats the defense proposed here by suppressing the refusal score of its conjunctive detection criterion while leaving the harmfulness score unchanged (Section 5.4). We use no attack that is not already public, and we introduce no new attack family.

Harmful content. Our evaluation uses harmful prompts drawn from public safety benchmarks and elicits harmful completions from the victim model. Where the paper quotes model output, we

reproduce only the opening of a reply and omit actionable content. Harmfulness is assessed by Llama-Guard-3-8B rather than by human annotators, so no annotator was exposed to harmful model output.

Released artifacts. The code we release reproduces our evaluation. It contains the implementation of the method proposed here, our ports of the compression baselines and of the jailbreak attacks against which it is evaluated, and the scripts that generate the adversarial prompts used here. The baselines and attacks are already publicly available from their original authors.

Data. All corpora used here are public research datasets, used consistently with their licences. The work involves no human subjects and no personal data.

Reproducibility statement

Every number a claim rests on appears in the main text, and the appendix and the supplementary material carry what is needed to reproduce it. Appendix L gives the configuration of every method and, for all three victim models, the token positions, the layer ranges and the thresholds. Appendix K gives the hardware and the timing protocol. Appendix A gives the evaluation protocol, the number of attempts per attack, the decoding settings and the judge. Appendix B describes our implementations of RobustKV and AnchorKV and the two places where we depart from their released defaults, which anyone rerunning them from the original code will need. The supplementary material is an anonymous archive holding the code for every experiment, the per-token scores and per-sample verdicts the tables are computed from, and a script that downloads the public datasets. Following the ethics statement it ships the scripts that generate the adversarial prompts but not the prompt sets.

References

Gabriel Alon and Michael Kamfonas. Detecting language model attacks with perplexity, 2023. URL https://arxiv.org/abs/2308.14132.

Cem Anil, Esin Durmus, Nina Panickssery, et al. Many-shot jailbreaking. In *Advances in Neural Information Processing Systems*, volume 37, pp. 129696–129742. Curran Associates, Inc., 2024. doi: 10.52202/079017-4121. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ea456e232efb72d261715e33ce25f208-Paper-Conference.pdf.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*, volume 37, pp. 136037–136083. Curran Associates, Inc., 2024. doi: 10.52202/079017-4322. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/f545448535dfde4f9786555403ab7c49-Paper-Conference.pdf.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137. Association for Computational Linguistics, August 2024. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=ayi7qezU87.

Bochuan Cao, Yuanpu Cao, Lu Lin, and Jinghui Chen. Defending against alignment-breaking attacks via robustly aligned LLM. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10542–10560. Association for Computational Linguistics, August 2024. doi: 10.18653/v1/2024.acl-long.568. URL https://aclanthology.org/2024.acl-long.568/.

Alessio Devoto, Yu Zhao, Simone Scardapane, and Pasquale Minervini. A simple and effective L_2 norm-based strategy for KV cache compression. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18476–18499. Association for Computational Linguistics, November 2024. doi: 10.18653/v1/2024.emnlp-main.1027. URL https://aclanthology.org/2024.emnlp-main.1027/.

Peng Ding, Jun Kuang, Dan Ma, Xuezhi Cao, Yunsen Xian, Jiajun Chen, and Shujian Huang. A wolf in sheep's clothing: Generalized nested jailbreak prompts can fool large language models easily. 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. 2136–2153. Association for Computational Linguistics, June 2024. doi: 10.18653/v1/2024.naacl-long.118. URL https://aclanthology.org/2024.naacl-long.118/.

Shravan Doda. Before the last token: Diagnosing final-token safety probe failures. In *Mechanistic Interpretability Workshop at ICML 2026*, 2026. URL https://openreview.net/forum?id=yvWFdjZlBC.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S Kevin Zhou. Ada-KV: Optimizing KV cache eviction by adaptive budget allocation for efficient LLM inference. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=tcisuhGsQZ.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Xulin Hu, CHE WANG, Wei Yang Bryan Lim, Jianbo Gao, and Zhong Chen. Tracing the dynamics of refusal: Exploiting latent refusal trajectories for robust jailbreak detection. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=DjX8eh4PIm.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama Guard: LLM-based input-output safeguard for Human-AI conversations, 2023. URL https://arxiv.org/abs/2312.06674.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models, 2023. URL https://arxiv.org/abs/2309.00614.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B, 2023. URL https://arxiv.org/abs/2310.06825.

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://openreview.net/forum?id=L5godAOC2z.

Dongwon Jo, Jiwon Song, Yulhwa Kim, and Jae-Joon Kim. FastKV: Decoupling of context reduction and KV cache compression for prefill-decoding acceleration. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 32167–32186. Association for Computational Linguistics, July 2026. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1610. URL https://aclanthology.org/2026.findings-acl.1610/.

Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. DeepInception: Hypnotize large language model to be jailbreaker. In *Neurips Safe Generative AI Workshop 2024*, 2024a. URL https://openreview.net/forum?id=bYa0BhKR4q.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 22947–22970. Curran Associates, Inc., 2024b. doi: 10.52202/079017-0722. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/28ab418242603e0f7323e54185d19bde-Paper-Conference.pdf.

Tong Liu, Yingjie Zhang, Zhe Zhao, Yinpeng Dong, Guozhu Meng, and Kai Chen. Making them ask and answer: Jailbreaking large language models in few queries via disguise and reconstruction. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 4711–4728, Philadelphia, PA, August 2024a. USENIX Association. ISBN 978-1-939133-44-1. URL https://www.usenix.org/conference/usenixsecurity24/presentation/liu-tong.

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*, 2024b. URL https://openreview.net/forum?id=7Jwpw4qKkb.

Avery Ma, Yangchen Pan, and Amir massoud Farahmand. PANDAS: Improving many-shot jailbreaking via positive affirmation, negative demonstration, and adaptive sampling. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=sEBfiF8JBu.

Xiaoxiao Ma, Kuofeng Gao, Zeyi Lu, Wenxi Jiang, Hao Fang, Hao Wu, Bin Chen, and Shu-Tao Xia. When efficiency meets safety: A benchmark security analysis of KV cache compression in large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24472–24485. Association for Computational Linguistics, July 2026. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1123. URL https://aclanthology.org/2026.acl-long.1123/.

Robert G. Newcombe. Interval estimation for the difference between independent proportions: comparison of eleven methods. *Statistics in Medicine*, 17(8):873–890, 1998.

Ning Ni and Yingjie Lao. AnchorKV: Safety-aware KV cache compression via soft penalty with a refusal anchor, 2026. URL https://arxiv.org/abs/2606.17872.

Kyle O'Brien, David Majercak, Xavier Fernandes, Richard G. Edgar, Blake Bullwinkel, Jingya Chen, Harsha Nori, Dean Carignan, Eric Horvitz, and Forough Poursabzi-Sangdeh. Steering language model refusal with sparse autoencoders. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*, 2025. URL https://openreview.net/forum?id=PMK1jdGQoc.

Junyoung Park, Dalton Jones, Matthew J Morse, Raghavv Goel, Mingu Lee, and Christopher Lott. KeyDiff: Key similarity-based KV cache eviction for long-context LLM inference in resource-constrained environments. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=uBaFH7aQnC.

Anselm Paulus, Arman Zharmagambetov, Chuan Guo, Brandon Amos, and Yuandong Tian. AdvPrompter: Fast adaptive adversarial prompting for LLMs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=dqcl5SNbc8.

Valentin Petrov. On the failure of topic-matched contrast baselines in multi-directional refusal abliteration, 2026. URL https://arxiv.org/abs/2603.22061.

Giorgio Piras, Raffaele Mura, Fabio Brau, Luca Oneto, Fabio Roli, and Battista Biggio. SOM directions are better than one: multi-directional refusal suppression in language models. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence and Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence and Sixteenth Symposium on Educational Advances in Artificial Intelligence*. AAAI Press, 2026. ISBN 978-1-57735-906-7. doi: 10.1609/aaai.v40i39.40551. URL https://doi.org/10.1609/aaai.v40i39.40551.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Traian Rebedea, Razvan Dinu, Makesh Narsimhan Sreedhar, Christopher Parisien, and Jonathan Cohen. NeMo guardrails: A toolkit for controllable and safe LLM applications with programmable rails. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 431–445. Association for Computational Linguistics, December 2023. doi: 10.18653/v1/2023.emnlp-demo.40. URL https://aclanthology.org/2023.emnlp-demo.40/.

Alexander Robey, Eric Wong, Hamed Hassani, and George J. Pappas. SmoothLLM: Defending large language models against jailbreaking attacks. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=laPAh2hRFC.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. 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. 5377–5400. Association for Computational Linguistics, June 2024. doi: 10.18653/v1/2024.naacl-long.301. URL https://aclanthology.org/2024.naacl-long.301/.

Dominik Schwarz. The entanglement wall: Activation-space probes as risk detectors, not context adjudicators, 2026. URL https://arxiv.org/abs/2607.13075.

Chenyu Shi, Xiao Wang, Qiming Ge, Songyang Gao, Xianjun Yang, Tao Gui, Qi Zhang, Xuanjing Huang, Xun Zhao, and Dahua Lin. Navigating the OverKill in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4602–4614. Association for Computational Linguistics, August 2024. doi: 10.18653/v1/2024.acl-long.253. URL https://aclanthology.org/2024.acl-long.253/.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford Alpaca: An instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does LLM safety training fail? In *Advances in Neural Information Processing Systems*, volume 36, pp. 80079–80110. Curran Associates, Inc., 2023. doi: 10.52202/075280-3508. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/fd6613131889a4b656206c50a8bd7790-Paper-Conference.pdf.

Zeming Wei, Yifei Wang, Ang Li, Yichuan Mo, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(6):6835–6846, 2026. doi: 10.1109/TPAMI.2026.3660147.

Tom Wollschläger, Jannes Elstner, Simon Geisler, Vincent Cohen-Addad, Stephan Günnemann, and Johannes Gasteiger. The geometry of refusal in large language models: Concept cones and representational independence. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=80IwJqlXs8.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pp. 21875–21895, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/5e5fd18f863cbe6d8ae392a93fd271c9-Paper-Conference.pdf.

Tinghao Xie, Xiangyu Qi, Yi Zeng, Yangsibo Huang, Udari Madhushani Sehwag, Kaixuan Huang, Luxi He, Boyi Wei, Dacheng Li, Ying Sheng, Ruoxi Jia, Bo Li, Kai Li, Danqi Chen, Peter Henderson, and Prateek Mittal. SORRY-Bench: Systematically evaluating large language model safety refusal. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=YfKNaRktan.

Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. Defending ChatGPT against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 5:1486–1496, 2023. URL https://api.semanticscholar.org/CorpusID:266289038.

Dongjie Yang, Xiaodong Han, Yan Gao, Yao Hu, Shilin Zhang, and Hai Zhao. PyramidInfer: Pyramid KV cache compression for high-throughput LLM inference. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3258–3270, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.195. URL https://aclanthology.org/2024.findings-acl.195/.

Wei Jie Yeo, Nirmalendu Prakash, Clement Neo, Ranjan Satapathy, Roy Ka-Wei Lee, and Erik Cambria. Understanding refusal in language models with sparse autoencoders. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 6377–6399. Association for Computational Linguistics, November 2025. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.338. URL https://aclanthology.org/2025.findings-emnlp.338/.

Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. GPT-4 is too smart to be safe: Stealthy chat with LLMs via cipher. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=MbfAK4s61A.

Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. How Johnny can persuade LLMs to jailbreak them: Rethinking persuasion to challenge AI safety by humanizing LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14322–14350, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.773. URL https://aclanthology.org/2024.acl-long.773/.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang "Atlas" Wang, and Beidi Chen. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34661–34710. Curran Associates, Inc., 2023. doi: 10.52202/075280-1506. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6ceefa7b15572587b78ecfcebb2827f8-Paper-Conference.pdf.

Zhexin Zhang, Junxiao Yang, Pei Ke, Fei Mi, Hongning Wang, and Minlie Huang. Defending large language models against jailbreaking attacks through goal prioritization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8865–8887. Association for Computational Linguistics, August 2024. doi: 10.18653/v1/2024.acl-long.481. URL https://aclanthology.org/2024.acl-long.481/.

Jiachen Zhao, Jing Huang, Zhengxuan Wu, David Bau, and Weiyan Shi. LLMs encode harmfulness and refusal separately. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 140283–140318. Curran Associates, Inc., 2025. doi: 10.52202/085713-4688. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/cd18539787d90e1d682d557c2c71b534-Paper-Conference.pdf.

Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Jing Jiang, and Min Lin. Improved few-shot jailbreaking can circumvent aligned language models and their defenses. In *Advances in Neural Information Processing Systems*, volume 37, pp. 32856–32887. Curran Associates, Inc., 2024. doi: 10.52202/079017-1034. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/39a3aa9dfd0280ff8fbad1d330662cac-Paper-Conference.pdf.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023. URL https://arxiv.org/abs/2307.15043.

A CONFIGURATIONS FOR THE MOTIVATION EXPERIMENT

The original setting of many-shot jailbreaking cannot be reproduced on an open-weight model: Anil et al. (2024) insert the forged dialogue history through an API, which they note ChatGPT and Claude.ai cannot do, and their demonstrations come from an internal helpful-only model that was never released. Injecting that history instead as real user and assistant turns through the chat template, with the demonstration pool shared across the context-accumulation attacks, leaves it identical to ICA in pool, injection, and shot curve. We therefore report many-shot jailbreaking as ICA at high shot counts.

Across the evaluation, the number of attempts follows each attack's original protocol: one attempt for each single-prompt attack, one at each shot count for the context-accumulation attacks, and ten for AdvPrompter, whose attacker samples suffixes at low cost and which RobustKV also evaluates over ten attempts. We use greedy decoding, add no safety system prompt, and use Llama-Guard-3-8B (Inan et al., 2023) as the safety judge.

B IMPLEMENTATION OF ROBUSTKV AND ANCHORKV

The compression threshold should be a fraction. Both papers define the compression budget as a proportion of the prompt length: RobustKV evicts 20% of the prompt, and AnchorKV retains $\lfloor \rho L \rfloor$ tokens at $\rho = 0.25$. In practice the code defaults to a fixed token count instead. The released RobustKV code fixes a capacity of 80 tokens. AnchorKV has released no code, so we implement its Algorithm 2 on the FastKV backbone its paper specifies, and the released FastKV code defaults to fixed thresholds of 2048 for token-selective propagation and 512 per layer, with a proportional mode available.

That difference has no effect in the RobustKV and FastKV evaluations, where every prompt sits far above the fixed value. RobustKV evaluates on Llama-2-7B-Chat, whose chat template inserts a safety system prompt of roughly 100 tokens, so that with AutoDAN's template every prompt runs to several hundred tokens, and FastKV on LongBench (Bai et al., 2024), where prompts run to thousands of tokens. Our victim model, Llama-3.1-8B-Instruct, inserts no system prompt by default, and its prompts are correspondingly shorter. The median length of AdvPrompter's 500 adversarial prompts is exactly 80, the capacity the RobustKV code fixes, so at that setting it evicts nothing on half of them and returns results identical to FullKV's. The same fixed capacity limits it at the other end, since the eviction candidates are the top 64 positions of each attention head, that capacity of 80 less the window of 16. On ICA at 128 shots they cover 11% of positions, and RobustKV delivers 56% of its intended budget.

We therefore run each method in the proportional form its own paper defines, leaving every budget at the value that paper reports. For RobustKV the fraction is $0.8L$, the complement of its 20% eviction rate and the value carried in its source. For AnchorKV the proportional form is also what its reported numbers reflect, since at a median prompt length of 80 the fixed FastKV thresholds would leave FastKV identical to FullKV, whereas it reports 0.506 for FastKV against 0.440 for FullKV. After the change RobustKV reaches its full budget at every length, from 0 of 278 prompts compressed to 278 of 278 on AdvPrompter.

AnchorKV's importance immunity disables its own safety signal. AnchorKV subtracts a soft penalty from each token's attention importance score and then retains the top 25%. Its importance immunity zeroes out that penalty for the tokens with the largest baseline scores, the top 30%. What AnchorKV retains therefore lies inside the immune set, the penalty reaches only tokens that would be evicted regardless, and what it retains equals what FastKV retains for every λ. Running $\lambda = 0$ and $\lambda = 10$ with all else held fixed returns identical output. This contradicts the paper's own Figure 4, which plots ASR against λ and identifies three regimes, among them an effective range over $[1, 30]$ where ASR drops from 0.68 to between 0.13 and 0.17. Whenever the retention rate lies

below the immunity threshold, ASR does not depend on λ and the curve is flat, which indicates that the configuration behind that figure differs from the one its text describes. Taking the reported values literally would mean evaluating an object that reduces to FastKV, from which a failure to defend would say little about AnchorKV. We therefore disable the importance immunity and keep every other reported value, which places the penalty inside the effective range Figure 4 describes. The change only widens the set of tokens the penalty reaches, so the AnchorKV results in Section 1 are the strongest we are able to obtain for it.

C ATTACK STRENGTH CURVES FOR CONTEXT ACCUMULATION

Figure 5 gives the three context-accumulation curves behind Section 1. RobustKV and FastKV are both Ineffective at every shot count of all three attacks. AnchorKV does change the outcome on ICA and on I-FSJ, but on ICA it is no longer Robust at the highest shot count tested, and an attacker removes it by adding demonstrations. On PANDAS AnchorKV is never Robust.

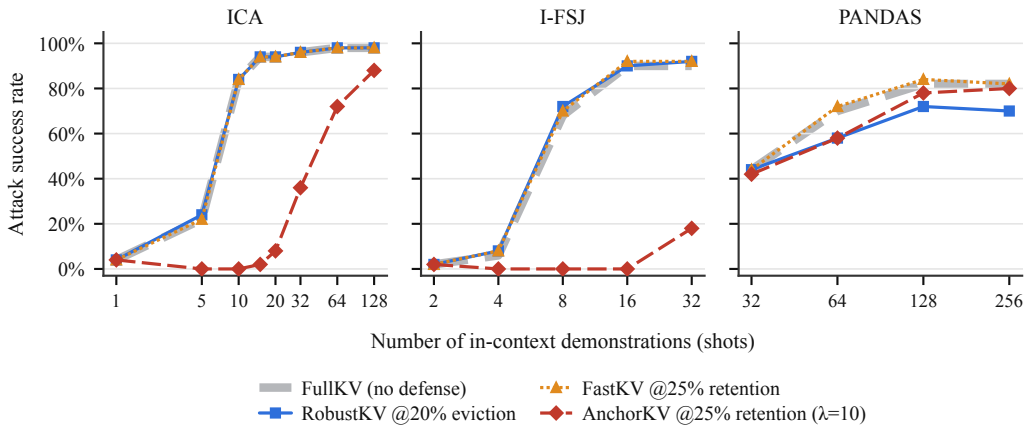

Figure 5: Attack success rate of ICA, I-FSJ and PANDAS against the number of in-context demonstrations, under FullKV, RobustKV, FastKV and AnchorKV.

D EVICTION COMPOSITION ACROSS SHOT COUNTS

Figures 6, 7 and 8 show that the composition of what each method evicts does not move with the shot count.

E AGGREGATE MEASUREMENTS

Tables 5 and 6 give the measurements of evictions, one table for each method.

F EVICTION BY ROBUSTKV AND ANCHORKV ON THE REMAINING ATTACKS

Figures 9 to 16 show, for each of the remaining attacks, which tokens RobustKV and AnchorKV target and which they evict.

G DISCRIMINATIVE SELECTION AGAINST CAUSAL SELECTION

Figure 17 plots every candidate direction on both selection criteria.

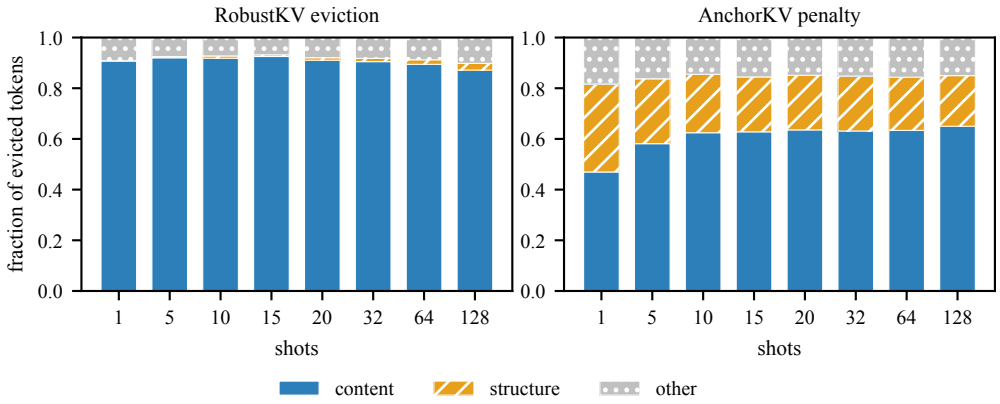

Figure 6: ICA, composition of the evicted tokens across shot counts, RobustKV on the left and the AnchorKV penalty on the right. Each bar sums to one. Hatching and colour carry the same distinction.

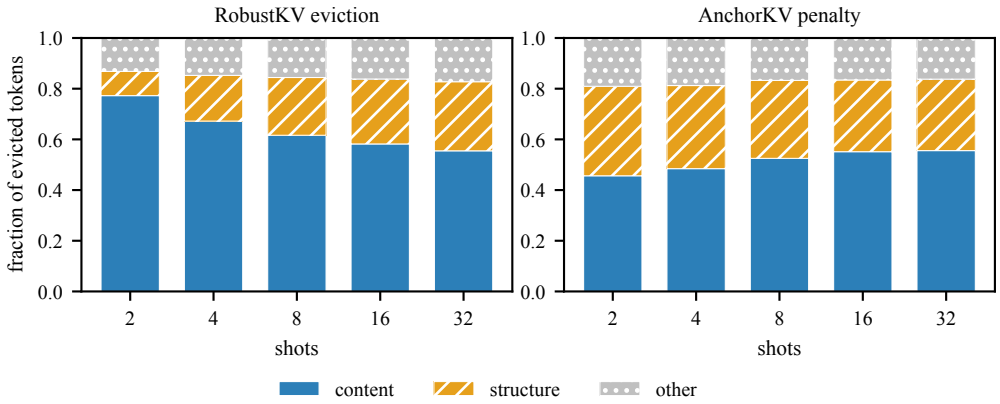

Figure 7: I-FSJ, composition of the evicted tokens across shot counts, read as in the previous figure.

H Activation patching of the QK and OV circuits

Figure 18 reports what patching the attention pattern alone, the value vectors alone, and both together recovers of the refusal.

I Causal selection of the harmfulness and refusal directions

Table 7 gives every candidate layer and token position.

J Deletion targets on every attack

Table 8 gives the three deletion targets.

K Hardware

All timings run on a single NVIDIA A100 80GB PCIe GPU at batch size one, with the model generating 256 tokens and not stopping at the EOS token. For each sample we discard the first two

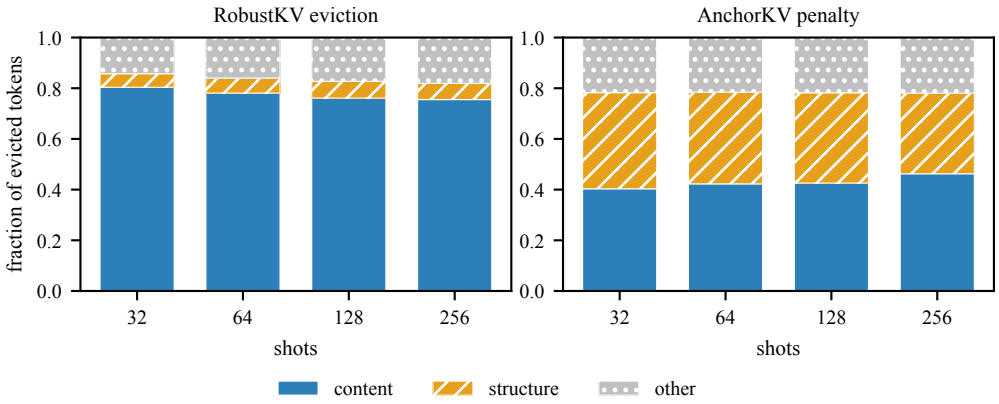

Figure 8: PANDAS, composition of the evicted tokens across shot counts, read as in the previous figure.

Table 5: What RobustKV evicts. The harmful query, content and structural columns are summed over the fifty goals of each configuration. Structural is template, role, date and punctuation. N/A has two meanings: on DRA the harmful query is a puzzle reconstructed only during generation and does not exist as tokens; on the context-accumulation attacks the query is the final turn and lies inside the observation window, which is retained without being scored, so its rank is undefined.

Attack	Harmful query (%)	Content (%)	Structural (%)	Importance rank
AutoDAN	7	33	67	0.51
DeepInception	4	45	55	0.39
ReNeLLM	1	58	39	0.39
DRA	N/A	64	36	N/A
AdvPrompter	0	14	86	0.64
ICA@15	0	93	7	N/A
ICA@128	0	87	13	N/A
I-FSJ@16	0	58	42	N/A
PANDAS@64	0	78	22	N/A

runs and report the median of the next three. We take the median rather than the mean because external load on a single run moves the mean but leaves the median unchanged.

L CONFIGURATIONS

Table 9 gives the configuration of every method. FastKV runs at two different budgets in this paper, and the two serve different comparisons. Section 1 runs it at the 25% retention rate AnchorKV reports for its own FastKV baseline (Ni & Lao, 2026). There FastKV is AnchorKV with the penalty strength set to zero, and the two must share a budget for the difference between them to isolate the penalty. Section 5 instead runs it at a 20% budget for both the propagation and the cache, following FastKV's own evaluation setting, so that the value the four compression baselines are comparable with each other. The four context-accumulation attacks give the same ASR under both budgets, and the four single-prompt attacks differ by up to ten points.

In Llama-3.1-8B-Instruct the harmfulness direction is read at the last content token of the instruction, six positions from the end, over the causal layer band 6 to 13, and the refusal direction at the header token two positions from the end, over the band 13 to 16. Detection takes the peak within each band and then the maximum over a window of $W = 128$ tokens at the end of the body. A prompt is acted on only when both exceed their threshold, $\tau_h = 0.114$ and $\tau_r = 0.064$, each set at the 95th percentile of the calibration negatives. Qwen2.5-7B-Instruct reads harmfulness at -6 over layers 4 to 13 and

Table 6: What AnchorKV evicts. The first column is the share of the tokens crossing θ, the set the soft penalty acts on, that falls inside the harmful content; the second is the share of the whole prompt that is harmful content. N/A marks DRA, whose harmful content is a puzzle reconstructed only during generation and does not exist as tokens. The anchor is built on a harmless set matched in length, as the Anchor paper reports.

	Tokens crossing θ					All evicted tokens		
Attack	Harmful (%)	In prompt (%)	Struct. (%)	Content (%)	Imp. rank	Harmful (%)	Content (%)	Struct. (%)
AutoDAN	6	12	46	54	0.54	7	72	28
DeepInception	9	4	64	36	0.59	12	16	84
ReNeLLM	3	5	37	63	0.55	0	31	69
DRA	N/A	N/A	39	61	0.56	N/A	73	27
AdvPrompter	16	19	46	54	0.51	8	84	6
ICA@15	85	90	24	76	0.53	57	63	37
ICA@128	86	90	23	77	0.53	62	65	35
I-FSJ@16	87	91	42	58	0.52	60	55	45
PANDAS@64	72	76	33	67	0.54	37	42	58

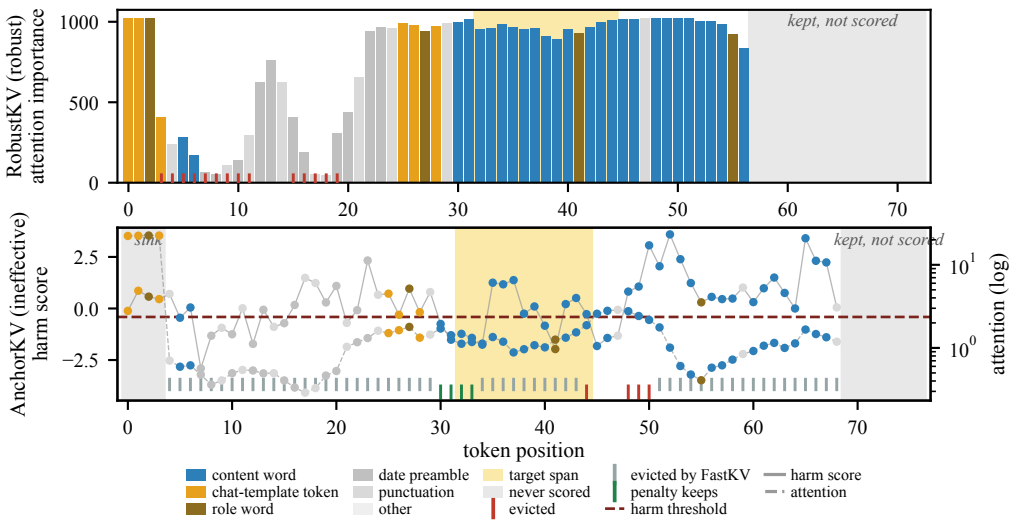

Figure 9: Eviction on AdvPrompter.

refusal at -1 over layers 10 to 16, with $\tau_h = 0.1813$ and $\tau_r = 0.0193$. Mistral-7B-Instruct-v0.3 reads harmfulness at -2 over layers 2 to 11 and refusal at -1 over layers 15 to 24, with $\tau_h = 0.2423$ and $\tau_r = 0.0584$. Every position and layer range is selected causally, and the calibration set is the same for all three.

M Context-accumulation attacks at every shot count

Tables 10 and 11 give the ASR and refusal rate at every shot count of the three context-accumulation attacks.

N Per-subtask utility

Table 12 gives the per-subtask scores on the sixteen subtasks of LongBench.

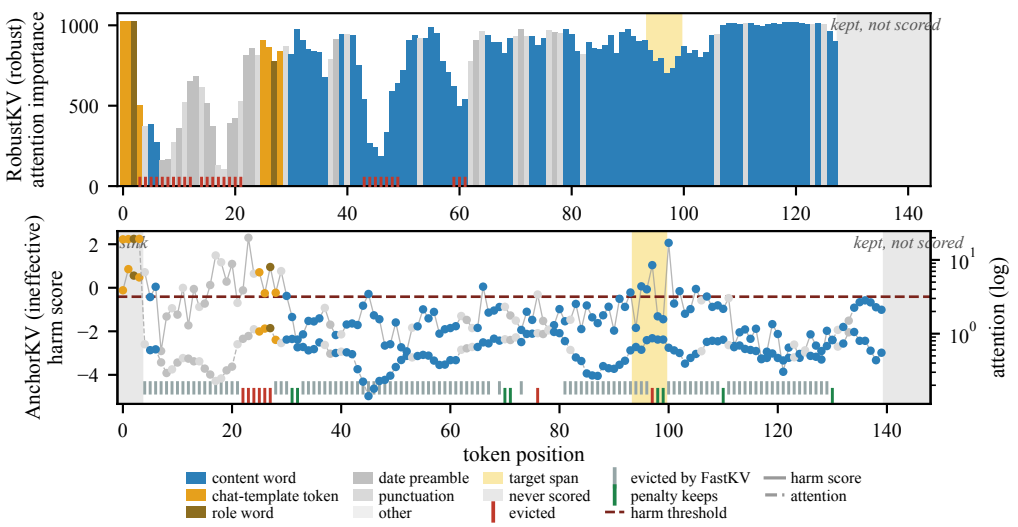

Figure 10: Eviction on DeepInception.

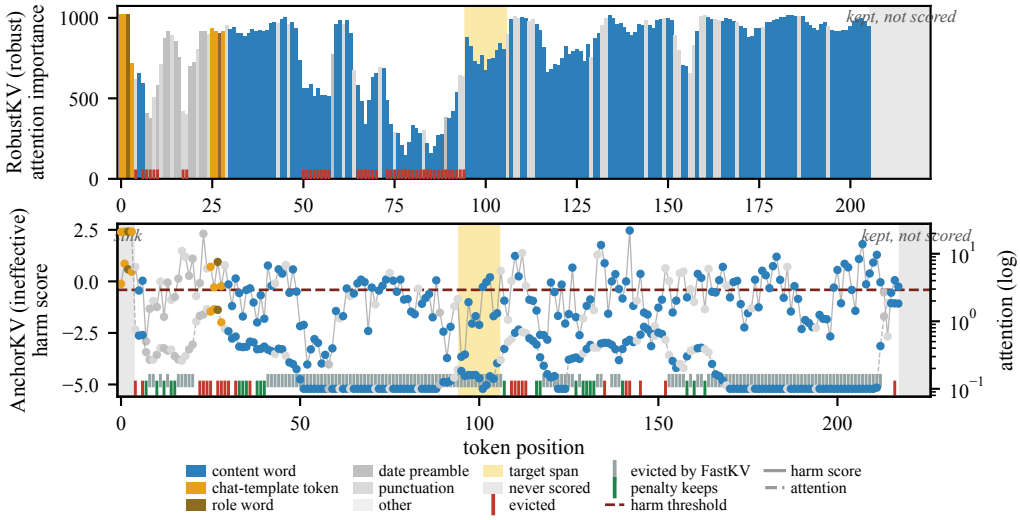

Figure 11: Eviction on ReNeLLM.

O CROSS-MODEL RESULTS AT EVERY ATTACK

Table 13 gives the attack success rate and the refusal rate of every attack on the three models, without defense and under HaRD.

P ADAPTIVE ATTACK RESULTS

Table 14 gives the attack flip rate and the repair flip rate of the three adaptive attacks on Llama-3.1-8B-Instruct.

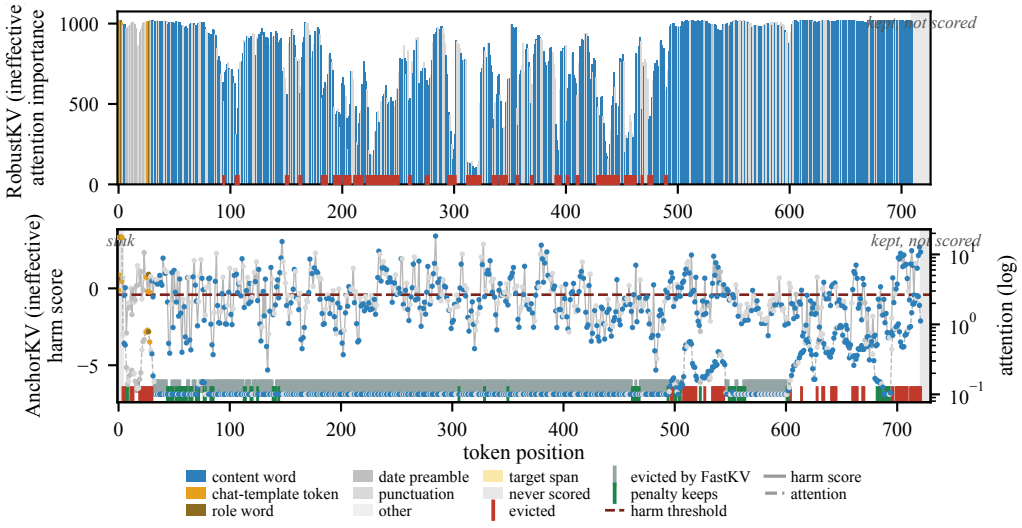

Figure 12: DRA eviction.

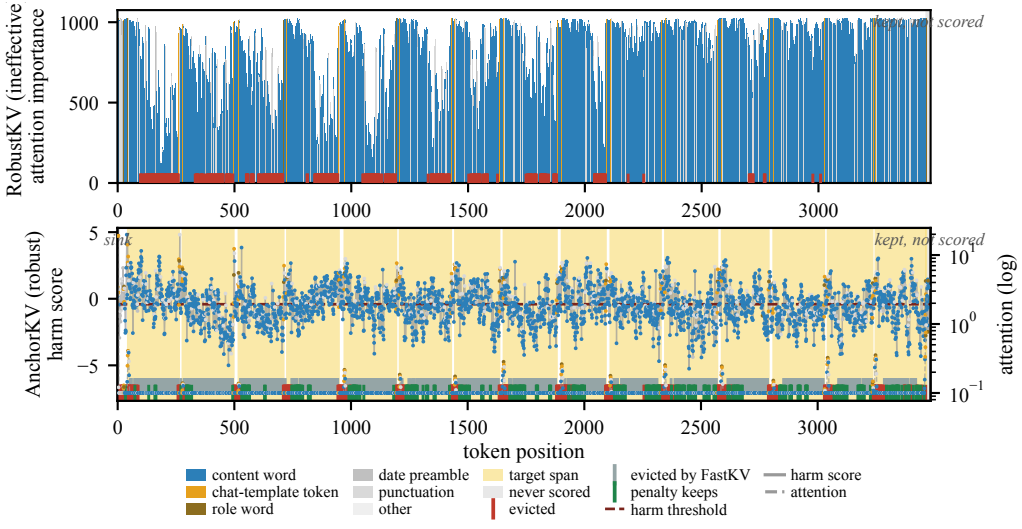

Figure 13: Eviction on ICA at 15 shots.

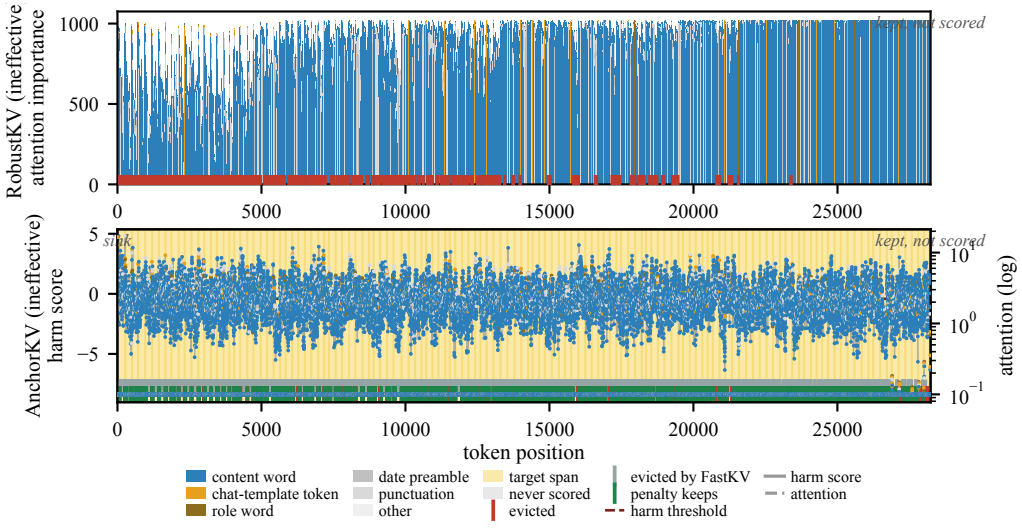

Figure 14: Eviction on ICA at 128 shots.

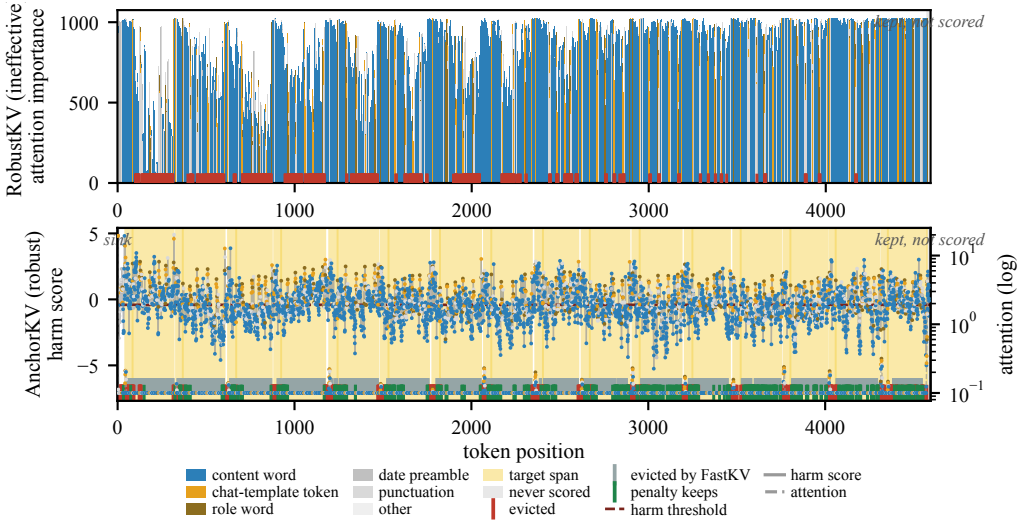

Figure 15: Eviction on I-FSJ at 16 shots.

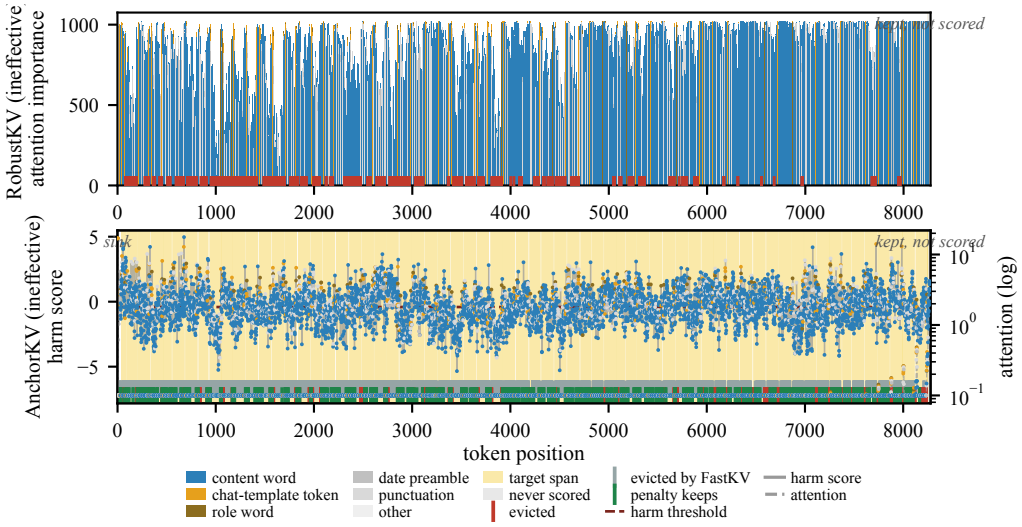

Figure 16: Eviction on PANDAS at 64 shots.

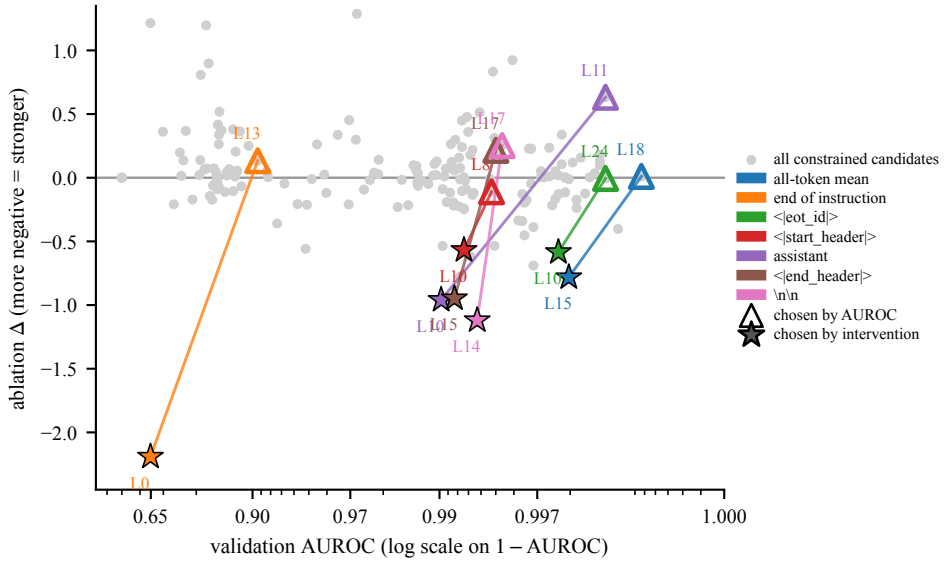

Figure 17: Layer selection by validation AUROC against layer selection by directional ablation. Triangles mark the layer AUROC selects at each token position, which is AnchorKV's rule; stars mark the layer directional ablation selects at the same position; the two are joined per position. The grey line marks zero effect, so a marker above it means the model refuses more after ablation.

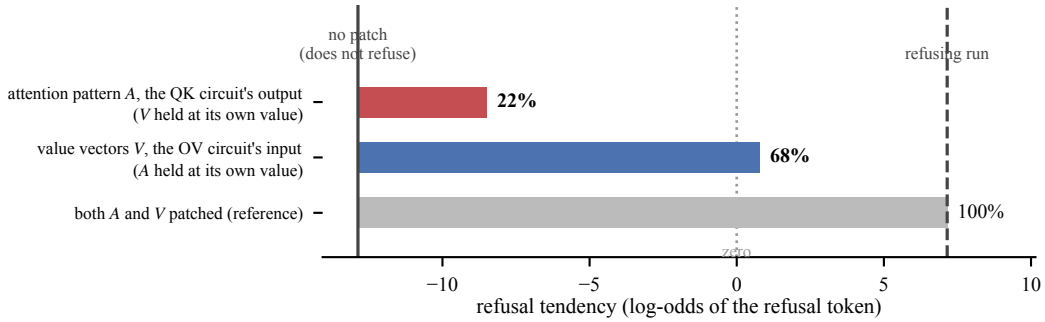

Figure 18: Activation patching of the attention output. Solid line, no patch; dashed line, the refusing run; dotted line, the point at which the model turns to refusing.

Table 7: Causal selection of the two directions by activation addition on harmless instructions. Flip is the fraction whose reply-inversion answer becomes Certainly, induction the fraction whose plain reply matches refusal wording, and gibberish the fraction of replies that are neither Certainly nor No. For the harmfulness direction flip is the effect and for the refusal direction induction is the effect while flip is the control, which should stay near zero. We add the harmfulness direction to the instruction tokens before the inversion question and the refusal direction to every token, since the first changes what the model reads and the second what it decides. A candidate with gibberish above 0.1 is discarded. Minimum coefficient is the smallest one reaching a flip of 0.9 under that gate, and a dash means no coefficient up to 8 reaches it. In the second sweep each token position is evaluated at its own best layer, given in its row, and a blank cell marks a column that sweep does not measure. Token positions are counted from the end of the prompt and selected rows are in bold.

Sweep	Candidate	Flip	Induction	Gibberish	Min. coeff.
Harmfulness direction, layers at the fixed token position					
	layer 6	0.16	–	0.04	4.0
	layer 8	0.84	–	0.04	6.0
	layer 9	**0.92**	–	**0.00**	**2.0**
	layer 10	0.88	–	0.00	6.0
	layer 11	0.72	–	0.00	–
	layer 12	0.48	–	0.00	–
	layer 13	0.12	–	0.00	–
Refusal direction, token position at its best layer					
	-1, `\n\n`, layer 15	0.28	0.52	–	–
	-2, <|end_header_id|>, layer 15	**0.04**	**0.68**	–	–
	-3, `assistant`, layer 13	0.00	0.20	–	–
	-4, <|start_header_id|>, layer 11	0.04	0.20	–	–
	-5, <|eot_id|>, layer 15	0.20	0.36	–	–
Refusal direction, layer at token position -2					
	layer 9	0.04	0.00	0.04	0.3
	layer 11	0.00	0.48	0.04	0.7
	layer 13	0.00	0.16	0.04	0.5
	layer 15	**0.04**	**0.68**	**0.08**	**0.7**

Table 8: ASR and refusal rate under the three deletion targets, 50 prompts per attack.

Group	Undefended	Remove by score	Remove the request	HaRD
AutoDAN	82	94 / 0	88 / 4	18 / 74
DeepInception	70	92 / 2	100 / 0	4 / 96
ReNeLLM	86	92 / 0	92 / 2	64 / 20
AdvPrompter	38	42 / 22	32 / 6	20 / 74
ICA@15	94	54 / 46	2 / 98	6 / 92
ICA@128	98	90 / 10	0 / 100	10 / 88
I-FSJ@16	90	58 / 38	0 / 100	6 / 92
PANDAS@64	70	94 / 6	0 / 100	0 / 100

Table 9: Configuration of every object. Each budget is the operating point its own paper reports.

Object	Source of its safety signal	Configuration
FullKV	none	no compression
SnapKV	none, compression only	20% budget, observation window 8
H2O	none, compression only	20% budget, inputs truncated to 7950 tokens
FastKV	none, compression only	20% budget for propagation and for the cache
StreamingLLM	none, compression only	20% budget, four attention sinks
RobustKV	the compression score	20% eviction rate, window 16
AnchorKV	the compression score	$\rho = 0.25$, $\lambda = 10$, $\ell^* = 15$, importance immunity off
Retokenization	the prompt	BPE-dropout at 20%
In-Context Defense	the prompt	two refusal demonstrations
SmoothLLM	the prompt	ten copies, 10% perturbation
HaRD	the residual stream	no compression; see below
Random deletion	none	same count and position distribution as HaRD

Table 10: ASR and refusal rate on the context-accumulation attacks at every shot count, for the KV methods. H2O runs out of memory on the four cases with the longest prompts.

	FullKV	SnapKV	H2O	FastKV	StreamingLLM	RobustKV	AnchorKV
ICA@1	4/96	4/96	4/96	4/96	4/96	4/96	4/96
ICA@5	22/78	22/78	22/78	18/82	22/78	24/76	0/100
ICA@10	84/16	84/16	84/16	84/16	82/16	84/16	0/100
ICA@15	94/6	94/6	94/6	94/6	94/6	94/6	2/98
ICA@20	94/6	94/6	94/6	94/6	92/6	94/6	8/92
ICA@32	96/4	96/4	96/4	96/4	96/4	96/4	36/64
ICA@64	98/2	98/2	n/a	98/2	98/2	98/2	72/28
ICA@128	98/2	98/2	n/a	98/2	98/2	98/2	88/12
I-FSJ@2	2/98	2/98	2/98	2/98	4/96	2/98	2/98
I-FSJ@4	6/92	8/92	8/92	6/94	6/92	8/90	0/100
I-FSJ@8	68/28	68/28	68/28	68/32	70/28	72/24	0/100
I-FSJ@16	90/6	92/6	90/6	92/6	92/6	90/6	0/100
I-FSJ@32	90/6	90/6	90/6	94/4	92/6	92/4	18/82
PANDAS@32	44/56	44/56	44/56	44/56	44/56	44/56	42/58
PANDAS@64	70/30	70/30	72/28	72/28	64/36	58/42	58/42
PANDAS@128	82/16	82/16	n/a	86/14	80/18	72/26	78/22
PANDAS@256	82/18	82/18	n/a	84/16	80/20	70/30	80/20

Table 11: ASR and refusal rate on the context-accumulation attacks at every shot count, for the text-layer guardrails, HaRD and the random-deletion contro.

	FullKV	Retokenization	In-Context Defense	SmoothLLM	HaRD	Random deletion
ICA@1	4/96	0/100	0/100	0/100	6/92	46/32
ICA@5	22/78	12/88	0/100	10/90	6/92	44/28
ICA@10	84/16	66/34	36/64	62/38	6/92	32/32
ICA@15	94/6	82/18	58/42	84/16	6/92	40/28
ICA@20	94/6	86/14	70/30	84/16	6/92	38/32
ICA@32	96/4	74/26	84/16	86/14	6/92	52/18
ICA@64	98/2	86/14	98/2	90/8	6/92	32/34
ICA@128	98/2	92/8	98/2	98/2	10/88	38/28
I-FSJ@2	2/98	4/96	0/100	2/98	6/92	32/18
I-FSJ@4	6/92	4/92	0/100	2/96	6/92	32/12
I-FSJ@8	68/28	52/44	14/82	44/52	6/92	30/12
I-FSJ@16	90/6	74/22	56/40	66/30	6/92	30/26
I-FSJ@32	90/6	86/10	82/14	78/18	6/92	42/18
PANDAS@32	44/56	44/56	38/62	52/48	0/100	34/48
PANDAS@64	70/30	58/42	60/40	70/30	0/100	34/44
PANDAS@128	82/16	70/30	80/18	92/8	0/100	24/52
PANDAS@256	82/18	78/22	80/20	84/16	2/98	36/42

Table 12: Utility on each of the sixteen English subtasks of LongBench.

	FullKV	SnapKV	H2O	FastKV	Stream.	Robust	Anchor	Retok.	ICD	Smooth	HaRD	Rand.
narrativeqa	29.2	29.7	14.7	30.1	28.5	27.9	30.7	30.3	30.6	23.4	27.4	1.4
qasper	44.9	43.8	39.7	43.7	27.8	43.6	44.3	43.6	45.1	38.9	44.9	3.7
multifieldqa_en	56.0	55.5	50.0	56.1	35.3	54.0	56.0	54.6	55.2	44.6	56.0	4.6
hotpotqa	57.9	57.9	47.7	57.0	50.6	56.9	57.2	56.5	58.0	45.6	56.9	0.7
2wikimqa	48.7	49.0	28.5	48.4	43.9	47.9	48.8	43.8	46.5	30.5	48.2	0.7
musique	33.0	32.7	24.3	31.5	25.5	32.8	32.1	32.7	31.8	19.6	32.5	0.6
gov_report	34.4	30.2	31.4	29.3	27.2	34.4	30.4	34.8	34.6	32.7	34.4	15.6
qmsum	25.4	25.0	23.4	25.0	22.0	25.1	25.5	25.1	25.5	23.5	25.1	12.1
multi_news	26.8	23.7	25.3	22.3	21.8	26.2	23.1	26.8	27.2	26.4	26.8	14.6
trec	72.5	70.0	68.0	73.0	64.0	68.0	73.5	72.5	73.0	46.0	72.5	0.0
triviaqa	91.6	91.7	91.2	92.0	90.0	91.9	91.5	91.8	92.4	78.4	90.6	2.6
samsum	44.3	42.4	43.5	42.7	42.0	42.9	43.1	44.8	44.4	37.8	43.6	3.2
passage_count	11.1	10.6	7.3	9.2	10.8	4.5	11.6	3.1	7.0	2.1	10.6	0.0
passage_retrieval_en	99.5	99.5	74.0	99.5	90.5	99.5	99.5	99.0	99.5	58.2	98.0	0.0
lcc	64.6	63.1	61.4	63.9	62.2	61.5	63.4	60.3	58.8	41.7	64.4	18.4
repobench-p	59.0	58.1	54.1	57.7	56.7	57.6	58.7	56.6	49.9	43.7	59.0	16.4

Table 13: Every attack on three models, as ASR/refusal in percent without defense and under HaRD. Cases whose prompts exceed a model's context window are marked n/a.

	Llama-3.1	Qwen2.5	Mistral	Mistral, refusing
AutoDAN	82/16 → 18/74	96/14 → 46/38	96/0 → 64/2	96/0 → 2/98
DeepInception	72/28 → 4/96	88/8 → 8/88	100/0 → 44/2	100/0 → 14/86
ReNeLLM	84/12 → 32/18	100/0 → 36/10	98/0 → 48/2	98/0 → 40/58
AdvPrompter	84/4 → 20/74	19/38 → 11/74	70/3 → 48/14	70/3 → 10/84
ICA@1	4/96 → 6/92	2/98 → 0/98	92/6 → 70/16	92/6 → 6/94
ICA@5	22/78 → 6/92	6/94 → 0/100	100/0 → 68/14	100/0 → 0/100
ICA@10	84/16 → 6/92	10/88 → 0/100	100/0 → 68/14	100/0 → 0/100
ICA@15	94/6 → 6/92	12/84 → 0/100	100/0 → 68/14	100/0 → 0/100
ICA@20	94/6 → 6/92	26/74 → 0/100	100/0 → 68/14	100/0 → 0/100
ICA@32	96/4 → 6/92	24/78 → 0/100	100/0 → 68/14	100/0 → 0/100
ICA@64	98/2 → 6/92	42/56 → 0/100	100/0 → 68/14	100/0 → 0/100
ICA@128	98/2 → 10/88	42/58 → 0/100	n/a	n/a
I-FSJ@2	2/98 → 6/92	2/94 → 0/100	82/14 → 70/16	82/14 → 6/94
I-FSJ@4	6/92 → 6/92	0/96 → 0/100	80/12 → 68/14	80/12 → 0/100
I-FSJ@8	68/28 → 6/92	4/94 → 0/100	98/0 → 68/14	98/0 → 0/100
I-FSJ@16	90/6 → 6/92	8/92 → 0/100	98/2 → 68/14	98/2 → 0/100
I-FSJ@32	90/6 → 6/92	10/92 → 0/100	98/0 → 68/14	98/0 → 0/100
PANDAS@32	44/56 → 0/100	4/90 → 0/96	100/0 → 34/4	100/0 → 2/98
PANDAS@64	70/30 → 0/100	8/90 → 0/98	100/0 → 22/8	100/0 → 2/98
PANDAS@128	82/16 → 0/100	8/74 → 0/98	100/0 → 34/8	100/0 → 6/94
PANDAS@256	82/18 → 2/98	n/a	n/a	n/a

Table 14: Adaptive attacks, evaluated on qualified seeds. A seed qualifies when HaRD defends against the seed before the adaptive attack, harmful output on them is zero, and the model is not jailbroken after the intervention. The attack flip rate is the fraction of these seeds that is jailbroken after the adaptive attack, and the repair flip rate is the fraction of those that the upgrade recovers. A n/a marks a row with no upgrade, either because the attack flips almost no seed or because closing the gap would raise false positives beyond what the design accepts.

Vulnerability	Attack	n	Attack flip	Repair flip
Observation window evasion	AdvPrompter	40	32%	69%
	AutoDAN	41	93%	53%
Sentence-boundary removal	ReNeLLM	33	9%	n/a
	AutoDAN	41	7%	n/a
Single-direction suppression	AutoDAN (harmfulness direction)	41	0%	n/a
	ReNeLLM (refusal direction)	33	15%	n/a