PROBEFOOL: EVADING LLM JAILBREAK ACTIVATION MONITORS WITH PROBE-AGNOSTIC ATTENTION ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Activation probes are increasingly proposed as lightweight monitors for harmful language-model requests. We ask whether attention at the monitored token is a usable proxy for the probe's own signal: whether the representation a monitor reads can be moved without ever optimizing against it. We introduce PROBEFOOL, a continuous-embedding attack that places a fixed benign distractor before an unchanged jailbreak prompt and optimizes a trainable steerer after it from frozen base-model signals alone, raising the monitored token's attention to the distractor while preserving the original response distribution. The probe is a decision-only black box: its decision would tell a deployed attacker when to stop, but it never enters the update. Against logistic-regression and MLP probes on Llama-3.2-3B-Instruct, Llama-3-8B-Instruct, and Mistral-7B-Instruct-v0.2, we conservatively report terminal-step results averaged over every transformer-layer probe. With the best of three distractor–steerer initializations selected on each model–probe track by joint success, the attack achieves 77–93% evasion with 52–77% joint success, meaning evasion and a still-harmful response on the same prompt, and it transfers across broad layer ranges and to unseen prompts. Controlled interventions locate the mechanism. Transplanting the attack's monitored-token attention onto an unoptimized wrapper recovers most of its evasion, while the optimized embedding under clean attention evades almost nothing. At fixed mass, which heads carry it decides the outcome.

1 INTRODUCTION

Activation monitoring is increasingly proposed as a complement to text-level moderation. Lightweight classifiers over hidden states can detect malicious prompts and unsafe behavior before generation, including information weakly expressed in surface text (Burns et al., 2023; Patel & Wang, 2024). Yet strong clean accuracy does not establish adaptive robustness when an attacker can manipulate the representation consumed by the monitor. We ask how far attention at the monitored token carries as a proxy for the probe's own signal.

The closest adaptive attack optimizes a probe-obfuscation objective alongside behavioral preservation, with probe gradients directing its strongest variant (Bailey et al., 2026). We ask whether a weaker monitor-access attacker can still evade: the base model is white-box, but the monitor is a protected decision-only service. The attacker may observe acceptance or rejection, and knows which token position the monitor reads, but no monitor score, feature, parameter, or gradient determines the update direction.

PROBEFOOL applies a continuous wrapper to an unchanged jailbreak prompt. A fixed benign distractor precedes the prompt and a trainable steerer follows it. Updates use two frozen-model objectives: preserve the original response distribution, and increase attention from the monitored token to the distractor across layers.

We test it against linear and nonlinear probes on three models: Llama-3.2-3B-Instruct, Llama-3-8B-Instruct (Grattafiori et al., 2024) and Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), abbreviated 3B, 8B and Mistral. The best initialization on each model–probe track is accepted as benign on 77–93% of prompts, averaged over every transformer-layer probe, with joint success up to 77%. 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.

wrapper crosses a broad band of independently trained monitor boundaries and transfers unchanged to prompts it never optimized against.

Ablating each loss term shows that attention routing is what crosses the monitor boundary and response preservation is what makes the crossing useful. Controlled interventions support a routing mechanism with two qualifications: forcing the monitored token's attention toward the distractor raises evasion dose-dependently on every track but by very different amounts, and which heads carry the mass decides the outcome at fixed mass. Transplanting the attack's attention onto the unoptimized wrapper recovers most of its evasion, whereas the optimized embedding under clean attention evades almost nothing.

2 Background and Related Work

Activation safety monitoring. Activation monitors turn internal representations into safety decisions, typically by fitting lightweight classifiers to hidden states and flagging harmful prompts before generation. Linear and MLP readouts can complement text classifiers (Patel & Wang, 2024); Layer Enhanced Classification applies classifiers to depth-truncated models (Sawtell et al., 2024); HSF filters final-layer prompt states (Qian et al., 2025); JBShield combines toxic- and jailbreak-concept detectors (Zhang et al., 2025); and Beyond Linear Probes uses a progressively evaluated truncated-polynomial classifier (Oldfield et al., 2026). Their readout choices vary: the last prompt token is common (Patel & Wang, 2024; Zhang et al., 2025), while HSF concatenates trailing prompt states and Beyond Linear Probes mean-pools all prompt states. Single-layer systems commonly select middle-to-late depths, although the preferred depth is task-dependent: shallow probes can be less adversarially robust (Patel & Wang, 2024), whereas EEG-Defender combines decisions over the first 75% of layers and exits before unsafe generation (Zhao et al., 2026). This variation makes layer policy part of the deployed rule and motivates our averaging across all transformer-layer placements rather than selecting one favorable layer.

Adaptive probe evasion. Obfuscated Activations optimizes a behavioral loss jointly with a differentiable monitor-obfuscation loss (Bailey et al., 2026). Its strongest attack optimizes a universal white-box embedding suffix that elicits the target response while minimizing the probe's output, so probe gradients direct the update; a separate gray-box variant optimizes per-prompt hard-token suffixes with a gradient-free search over 20,000 steps that repeatedly evaluates the probe and the model's logits. Neither operates from the monitor's accept/reject decision alone: the white-box attack needs probe gradients, and the gray-box one needs twenty thousand steps of probe feedback. The authors flag the single-bit monitor as a limitation of their own attacks. ProbeFool operates in that monitor setting: the decision can tell the attacker when to stop, but never specifies an update direction. Their probes are trained with jailbreak-transformed prompts held out, so jailbreaking is an unseen concept for them; for ours it is in-distribution, and only the wrapper is new.

Attention-guided attacks. AttnGCG adds an attention term to GCG that raises target-output-token attention to the adversarial suffix, and Attention Eclipse strengthens or suppresses interactions between prompt regions to amplify existing jailbreak optimizers (Wang et al., 2025; Zaree et al., 2025). These works establish attention as a feasible adversarial control variable. Our objective differs in both query orientation and target: we preserve an already successful response while moving a separate latent-monitor decision.

3 ProbeFool

Threat model. The activation monitor is the deployment's sole guardrail: an accepted request releases the output and a rejection withholds it, so we isolate monitor robustness from independent input or output moderation. Every update is computed from the frozen base model alone. The monitor's binary decision would permit stopping at the first wrapper achieving joint success; we do not exercise this, so every number is at the terminal checkpoint, a lower bound that charges the attacker the full optimization budget. Selecting which of three initializations to report likewise uses only that decision, so it stays inside the same access model; Appendix J gives the twenty-initialization population on 8B for comparison. Following standard latent-probing practice and the closest prior attack (Bailey et al., 2026), the monitor reads the final input token, the last token of the chat template after the user turn has closed, and the attacker knows that position. It lies outside the span the attacker

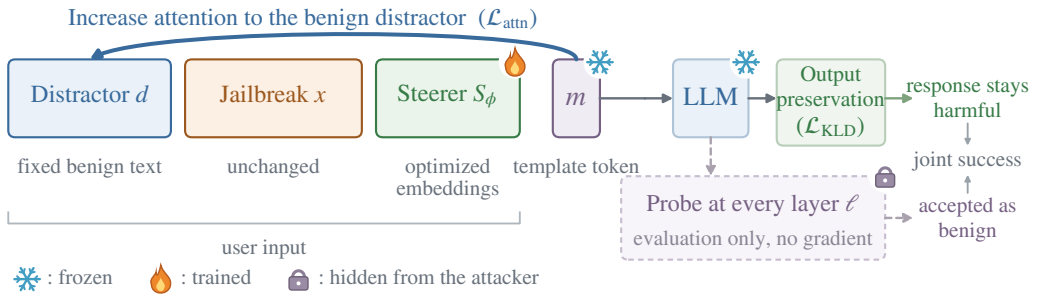

Figure 1: Overview of PROBEFOOL. The steerer is optimized from frozen-model signals alone to raise the readout token's attention to the distractor while preserving the response distribution. What the padlock hides is the monitor's layer, parameters, threshold and scores; its accept/reject decision is observable and never enters the update.

can write into, so the wrapper reaches the monitored representation only through attention and the residual stream; the attacker does not know the monitor's layer, parameters, threshold or scores, and the routing objective averages over all layers rather than targeting the selected one. A wrapper optimized on one candidate set is later frozen and applied unchanged to disjoint prompts to measure transfer.

Attack form. PROBEFOOL (Figure 1) treats attention at the monitor's readout as the attack surface. For candidate prompt x_i, fixed benign distractor d and trainable continuous steerer S_ϕ, the attacked input is $z_i(\phi) = d \oplus x_i \oplus S_\phi$. The same d and S_ϕ are applied across the candidate set $\mathcal{C}$, and optimization runs in embedding space over ϕ alone; d, the model and the monitor stay frozen. Two frozen-model objectives follow.

3.1 RESPONSE-DISTRIBUTION PRESERVATION

For each prompt, let $y_i = (y_{i,1}, \ldots, y_{i,T_i})$ be a known harmful response. Let $p_{i,t}$ be the frozen model's next-token distribution under x_i, teacher-forced along $y_{i,<t}$, and $q^{\phi}_{i,t}$ the corresponding distribution under $z_i(\phi)$. We minimize

$$L_{i,\text{KLD}} = \frac{\sum_{t=1}^{T_i} w_t \, \text{KL}(p_{i,t} \| q^{\phi}_{i,t})}{\sum_{t=1}^{T_i} w_t}, \quad w_t = e^{-\gamma(t-1)}. \tag{1}$$

We refer to $L_{i,\text{KLD}}$ as the inverse-exponential KLD, after the exponentially decaying weights w_t. Early tokens receive more weight because they strongly condition the remainder of an autoregressive response (Coalson et al., 2024; Zahran et al., 2025). This loss is a differentiable preservation proxy; generated responses are judged separately.

3.2 MONITORED-TOKEN ATTENTION ROUTING AND COMBINED OBJECTIVE

The routing term is simply stated: optimize the steerer so that the token the monitor reads pays more attention to the benign distractor. Let $\mathcal{D}_i$ be the distractor-token indices and m_i the monitored token. The routing loss averages the monitored token's distractor mass over all transformer layers $\mathcal{L}$ and all H attention heads per layer, and the objective combines it with the preservation term:

$$L_{i,\text{attn}} = 1 - \frac{1}{|\mathcal{L}|H} \sum_{\ell \in \mathcal{L}} \sum_{h=1}^{H} \sum_{j \in \mathcal{D}_i} A_{i,\ell,h}(m_i, j), \qquad L_i(\phi) = \lambda_{\text{KLD}} L_{i,\text{KLD}} + \lambda_{\text{attn}} L_{i,\text{attn}}. \tag{2}$$

The monitor's token scope fixes the query position. A single wrapper is optimized for the whole candidate set, minimizing $\frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} L_i(\phi)$ over ϕ. The attack changes attention indirectly through S_ϕ; it does not directly patch attention, values, residual states, or model parameters.

4 EVALUATION

All results use the configuration below unless noted otherwise.

Candidates and metrics. Jailbreak variants come from EasyJailbreak (Zhou et al., 2024), with GCG (Zou et al., 2023) from nanoGCG. Candidates are test-split prompts whose original jailbreak response has a stored StrongREJECT (Souly et al., 2024) score above 0.6, so every candidate starts from an unambiguously harmful response; we take up to ten per method from CodeChameleon (Lv et al., 2024), DeepInception (Li et al., 2023), GCG, GPTFuzzer (Yu et al., 2023) and ReNeLLM (Ding et al., 2024), giving $N = 50$ on 3B and Mistral and 49 on 8B. Freshly generated responses are judged at the standard 0.5 bar on all three models by the StrongREJECT fine-tuned evaluator, scoring responses generated greedily (temperature 0, up to 512 new tokens, one sequence at a time).

For transformer-layer probe $\ell \in \mathcal{L}$ and candidate i, let $E_i^\ell = 1$ if the probe accepts the wrapped request $z_i(\phi)$ as benign, and let $U_i = 1$ if its freshly generated response receives a StrongREJECT score above 0.5. The layer-specific evasion rate and joint success rate, and the layer-independent response success rate, are $\mathrm{ER}_\ell = N^{-1}\sum_i E_i^\ell$, $\mathrm{JSR}_\ell = N^{-1}\sum_i E_i^\ell U_i$ and $\mathrm{SR} = N^{-1}\sum_i U_i$. Because the monitored layer is the defender's choice and not the attacker's, we report $\overline{\mathrm{ER}} = |\mathcal{L}|^{-1}\sum_\ell \mathrm{ER}_\ell$ and JSR likewise, *averaged over all transformer-layer probes* of each model and probe family rather than at one validation-selected layer; "evasion" and JSR mean these layer-averaged quantities throughout. Where a figure plots *benign probability* it is the probe's $P(\text{benign})$ at the monitored token, averaged the same way, and a *detector margin* is $\operatorname{logit} P(\text{benign}) - \operatorname{logit} \tau$ for threshold τ. JSR is averaged per layer and only then across layers. Table 1 reports each initialization pair separately; every other table and every appendix number is the mean over the three pairs unless a pair is named. A *track* is one model crossed with one probe family, six in all; a *cell* is one track crossed with one initialization pair, eighteen in all.

Probes. We train logistic-regression (LogReg, also written LR) and MLP probes at every hidden-state layer of 3B, 8B and Mistral on the corpus the candidates are drawn from (benign, direct harmful and jailbreak-transformed prompts), so the five jailbreak methods are in-distribution for the monitor; splits are grouped by the underlying harmful request (Appendix A). Each probe is deployed at the threshold giving a nominal 1% FPR on validation, and we use every transformer-layer probe, excluding the embedding readout (all above 0.98 test ROC-AUC, Appendix A). Appendix I re-reads every result at 2% and 0.5% beside the deployed 1%, and Appendix H retrains the probes with each attacked jailbreak family held out of their training data.

Optimization. We optimize each candidate-set steerer for 500 Adam steps (settings in Appendix A). The distractor and initial steerer are drawn from three short textual pairs, numbered 1 to 3 throughout; their token embeddings initialize the continuous wrapper. We use $\lambda_{\mathrm{attn}} = 1$, $\lambda_{\mathrm{KLD}} = 0.1$ and $\gamma = 0.1$. Monitor decisions at every layer are recorded every 25 steps; responses are generated and judged at the terminal step for every reported rate. Figure 3's response-success curve is a separate post-hoc pass that regenerates and re-judges at each saved checkpoint.

5 ATTACK RESULTS

We report evasion and JSR at the terminal checkpoint, on the prompts each wrapper was optimized on and on held-out prompts; the extent to which a learned wrapper transfers across monitor layers; the quantities the optimization changes along its trajectory; the contribution of each loss term; and how an out-of-distribution detector fares against it.

Unless stated otherwise, every analysis after Table 1 uses the in-sample optimized cohort.

5.1 MONITOR EVASION

No model, pair or probe family is uniformly harder. Table 1 reports the terminal step. Mistral leads on both metrics and 3B leads 8B on evasion, so nothing tracks parameter count. The two probe families read the same optimization and stay close on average, but a single pair can split them widely (8B pair 2), and which family resists better flips between the Llamas and Mistral. Pair choice is load-bearing, most visibly on 8B, yet the best pair differs by model, so it is a property of the pair–model interaction rather than of the prefix itself (Appendix J widens this to twenty pairs).

Table 1: Evasion/SR/JSR (%, rounded). N is the held-out cohort size and belongs to the block on its right; the in-sample cohort is 50 on 3B and Mistral and 49 on 8B. SR is layer-independent, so it repeats across a model's two probe rows. Bold marks each row's best pair by JSR in sample. The two halves print the same rounded value in most cells even though the cohorts share no prompt (Appendix A); 95% bootstrap intervals are in Appendix L.

		Optimized prompts			Held-out prompts			
Model	Probe	Pair 1	Pair 2	Pair 3	N	Pair 1	Pair 2	Pair 3
3B	LR	70/62/44	**84/62/53**	81/62/50	50	70/54/39	84/56/47	81/50/40
3B	MLP	63/62/40	77/62/49	**84/62/52**	50	62/54/35	77/56/44	84/50/42
8B	LR	**77/78/60**	67/59/40	80/61/50	50	76/60/46	66/54/36	80/46/37
8B	MLP	**78/78/61**	44/59/27	78/61/49	50	78/60/46	45/54/24	78/46/36
Mistral	LR	76/82/63	72/82/59	**89/82/74**	255	76/80/61	72/75/54	89/76/68
Mistral	MLP	78/82/65	81/82/67	**93/82/77**	255	78/80/63	81/75/61	93/76/71

Evasion is method-agnostic; response success is not. Per method, layer-averaged evasion sits at 71–78% pooled over models, pairs and probes, with an average spread of 11 points across methods within a cell. SR is 74–83% on four methods and 33% on CodeChameleon, and it is equally low there under preservation alone (32%), so this is the cost of wrapping a code-formatted prompt at all, not of the attention term; the joint shortfall concentrates on that method.

Evasion transfers to held-out prompts; response success transfers less. The right half of Table 1 applies each trained wrapper, unchanged, to candidates that pass the same eligibility filter but are disjoint from the attack set: ten per jailbreak method on the Llamas and fifty-one on Mistral, whose eligible pool is far larger. No evasion cell moves by more than a point, whereas joint success drops on every track, and the shortfall is response utility: it tracks the drop in StrongREJECT success between the cohorts (8.7 points on 3B, 12.7 on 8B and 5.1 on Mistral). The responses that do succeed are as harmful as in sample (mean score among successes 0.72–0.81 on both cohorts), so the loss is in how many responses cross the threshold, not in how harmful the crossing ones are. Close agreement on evasion is partly forced by saturation, since most layers sit at exactly 0% or 100% on both cohorts.

Evasion accumulates with depth. The same shape appears on all six tracks (Fig. 2b): a noisy band over the first eight layers, where the wrapper evades 43–61% of prompts pooled over initializations, and a high band above it at 75–98%, beginning between layers 9 and 12. The readout tail is model-specific: evasion collapses over the last layers on 3B and 8B but holds through Mistral's. We read this as accumulation: the wrapper acts only on attention, so its effect on the monitored token's residual stream builds across blocks before it dominates the readout, which Section 6 shows directly: patching one layer moves the monitor far less than patching every layer; shallow probes may also retain surface evidence, since the jailbreak text is unchanged and early layers are associated with local token relationships. (Sawtell et al., 2024).

Evasion and response success are close to uncorrelated. JSR sits within five points of the product of evasion and SR at 93.5% of individual layers, with a median deviation of 0.00 and a worst case of 13, so the wrapper does not evade preferentially on the prompts whose responses stay harmful.

Layer averaging makes evasion precise; joint success is not. The sampling unit is the prompt, not the layer, since every layer scores the same cohort; bootstrapping over prompts gives 95% intervals on evasion spanning 0.6 to 4.9 points across the eighteen cells, against 12.6 to 23.2 on JSR, which inherits the per-prompt variance of response success. Evasion contrasts of more than about five points are therefore readable and joint-success contrasts of less than about twenty are less so (Appendix L).

5.2 Optimization Dynamics

Response success is restored before evasion is built. Evasion climbs from the unoptimized wrapper's level to its terminal value on all nine runs, mostly inside the first two hundred steps (Fig. 3a). Response success recovers within the first fifty steps and holds while evasion is still climbing. By

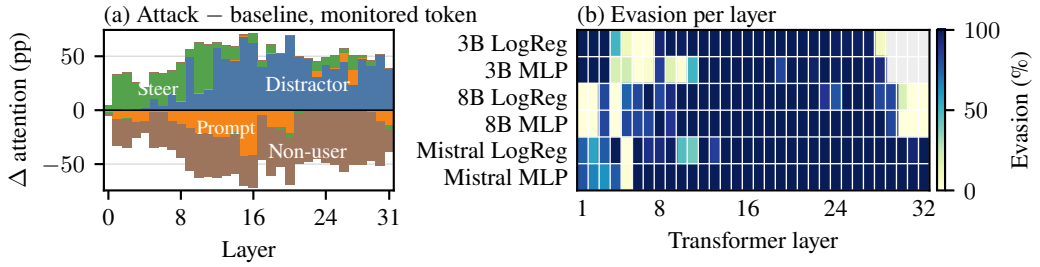

Figure 2: (a) Change in the monitored token's per-layer attention by token group under the attack (attack minus unoptimized wrapper, Mistral pair 1); gains stack upward from zero, losses downward. (b) Evasion at every transformer layer on the six tracks, at one pair per model (2 on 3B, 1 on 8B, 3 on Mistral), so a model's two rows are comparable. Each is the model's best pair by mean JSR over both probe families, except on 3B, where pairs 2 and 3 tie to within 0.3 points and we keep pair 2; Table 1 bolds per row, which picks pair 3 for 3B MLP.

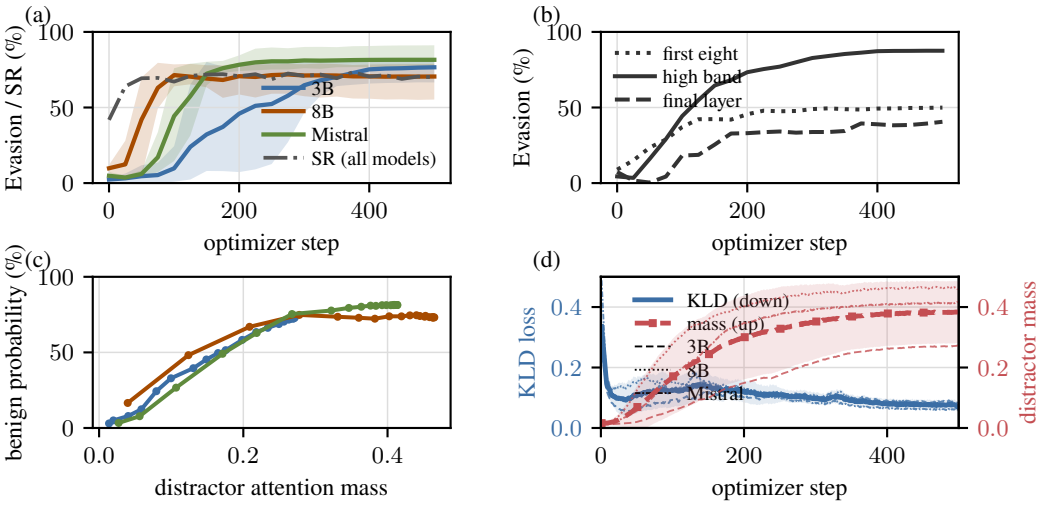

Figure 3: Attack trajectories over the nine model–pair runs. (a) Layer-averaged evasion per model with a band over the three initializations, and SR pooled over the nine runs, regenerated at each saved checkpoint. (b) Evasion by depth band, pooled over runs. (c) Distractor attention mass against mean benign probability at the same step. (d) The two loss terms, grand mean with a ± 1 sd band over the nine runs and thin per-model means, 11-step smoothed.

depth (Fig. 3b), the first eight layers are evaded first and plateau, the band above them overtakes at around step 75 and ends far higher, and the final layer stays lowest.

Routing is bought without further divergence. The inverse-exponential KLD is highest at initialization, the cost of inserting the unoptimized distractor and steerer, and falls within a few dozen steps as the wrapper learns to reproduce the reference response, while the monitored token's distractor attention mass climbs more than twentyfold (Fig. 3d); the rise costs nothing further in divergence. Benign probability rises with distractor mass along nearly the same curve on all three models until it saturates (Fig. 3c), after which 8B keeps adding mass with no further gain.

The wrapper draws the monitored token's attention onto itself, layer by layer. Per layer (Fig. 2a), the distractor and the steerer gain what non-user tokens and the prompt lose, with the steerer taking the first thirteen layers and the distractor the rest. Generation-time attention to the distractor is an order of magnitude smaller and decays after the first generated token, so the attack moves the representation the monitor reads without materially changing generation-time routing.

Table 2: Loss-component ablation, scored as Table 1 but averaged over the three pairs: evasion per probe family and in-sample SR, which is one generation per model. "Attention only" sets $\lambda_{\mathrm{KLD}}=0$, "preservation only" sets $\lambda_{\mathrm{attn}}=0$; "clean" is the unoptimized textual wrapper, whose SR is the jailbreak success it retains before any optimization (Appendix D).

		Evasion (%)				SR (%)			
Model	Probe	Attn. only	Pres. only	Both	Clean	Attn. only	Pres. only	Both	Clean
3B	LogReg	72.1	0.0	78.5	2.3	2.7	74.7	62.0	32.7
3B	MLP	82.1	0.0	74.7	2.5				
8B	LogReg	76.9	0.8	74.3	7.4	0.0	80.3	66.0	38.1
8B	MLP	73.5	1.0	66.7	12.2				
Mistral	LogReg	79.4	2.2	78.8	4.4	11.3	80.7	82.0	57.3
Mistral	MLP	79.3	0.8	84.3	5.2				

5.3 Loss-Component Ablation

We remove each term of Equation 2 in turn, holding everything else fixed and scoring as in Table 1; one optimization is read by both probe families, so evasion is per probe and response success once per model (Table 2; run provenance in Appendix A).

Attention crosses the monitor boundary; preservation makes the crossing useful. Attention alone evades as much of the network as the full objective but destroys the response, and preservation alone evades less than the unoptimized wrapper on every track (Table 2); neither alone produces joint success.

Preservation is close to free in coverage and buys the response at a price. The full objective sits within 7.4 points of attention alone on every track, with a sign that changes across initializations, so the term neither helps nor hurts any particular model or probe. Against preservation alone it gives up 12.7 points of response success on 3B and 14.3 on 8B, and none on Mistral. The unoptimized wrapper (Clean) is far below the attack on evasion, and its SR column shows the other half of the cost: inserting the wrapper alone leaves 32.7, 38.1 and 57.3% of jailbreaks working, which the full objective restores to 62.0, 66.0 and 82.0% (Appendix D).

5.4 An Out-of-Distribution Detector as a Defense

A Mahalanobis detector fitted to benign training states only, one of the latent-space defenses attacked in the obfuscated-activations work (Bailey et al., 2026), flags this attack at every layer of every pair on all three models, at 99.2% benign recall on held-out test data (worst layer 97.9%), matching the nominal 1% false-positive rate. Although this is a lightweight and effective detector for ProbeFool, its benign recall is measured in-distribution, and one-class OOD scores should not be treated as stand-alone harmfulness decisions: they measure atypicality rather than intent and can incur substantial false-positive rates on benign inputs, particularly under language or domain shift (Feng et al., 2026; Hua et al., 2026). Appendix C further analyzes the attacked readout. The stealthiest alternative objective we train is still flagged 67.6% of the time on 3B and 96.6% on Mistral (Appendices F and G), so lowering the OOD signature is not simply a matter of choosing a different objective.

6 Attention Intervention

The preceding results show association; here we intervene on attention directly while holding the input fixed. We first separate the two things the optimized attack changes, the embeddings it feeds the model and the attention those embeddings induce at the monitored token, by transplanting the monitored token's attention row between the clean and attacked forwards (Appendix K). The optimized embedding under clean attention evades at most 5.5% on any track and no more than the clean baseline on all six, and it stays there when the clean row is re-weighted to read the optimized steerer at the attack's own share (8.7% pooled against 5.7% clean) or at twice it. The attack's attention row on the unoptimized wrapper recovers 72 to 100% of the attack's evasion on the layers where the attack itself evades at least 80% of prompts, and 78 to 110% over all layers (Appendix K). Attention

routing is therefore necessary and largely sufficient, and the same factorization holds for a wrapper trained by an objective that never mentions attention. The rest of this section intervenes on it: first on how much mass the monitored token routes to the distractor, then on which heads carry it.

6.1 Absolute Distractor-Mass Intervention

We patch attention after softmax and before the value mixture, holding tokens, embeddings, parameters and value vectors fixed: for the monitored query in *every* attention block, distractor mass is set to $\rho \in \{0, 0.1, \ldots, 1\}$ and every non-distractor key is rescaled proportionally. Patching every block makes the intervention one operation across tracks, so the only difference between them is the probe that reads the result. Only forward passes are needed, so we run thirty pairs, three from each of ten distractor categories (Appendix M), at up to ten prompts per jailbreak method. Mass is forced on the *unoptimized* wrapper, which asks what routing alone can buy, and a control places the same mass on the final distractor token alone.

Routing accumulates across layers, and mass moves the monitor. Patching a single layer moves the monitor very little on average: over every block of every model the mean lift is 1.0 to 1.9 points and the median block moves under a point. A few early blocks are the exception, the strongest lifting evasion by about a fifth of what patching every block does on the same cases, while the blocks beneath each track's selected readout, where we first looked, stay within half a point of baseline. Patching every layer takes these thirty pairs from a 1–6% floor to 37–69% evasion at each track's best ρ (Appendix M), and benign probability rises with forced mass within 98.6% of the 8,940 prompt–pair trajectories, at a divergence that rises throughout (Fig. 4a). Restricted to the three pairs the attack optimizes, the sweep reaches 65.2% pooled against 54.7% over all thirty: strong initializations, but not the strongest available, since by peak evasion they rank eighth, tenth and twelfth of the thirty.

What the distractor says decides what routing buys. Mean evasion over the ρ grid runs from 49.7% for entity-attribute distractors to 12.7% for ones that refer to the task itself, and the category ordering is near-identical on all six tracks (Kendall's $W = 0.88$); category separates pairs more than the model–probe track does. Length is not the driver: distractor length tracks evasion across categories ($r_s = -0.53$) but not within them ($r_s = -0.22$, $p = 0.25$ after removing category means). Appendix M gives the categories and the controls.

Full mass is worse than most of it. Evasion peaks below full mass on five of six tracks, and the cost of overshooting is concentrated on 3B: the peak-to-ρ=1 decline is 11.0 and 10.4 points on its two tracks and at most 2.7 elsewhere, while divergence rises throughout. Full mass is not what costs: the final-token control reaches the same ceiling (55.5% against 54.7% pooled), reaches it at ρ=0.5 rather than 0.7, and on 3B declines by 1.4 and 0.0 points. Appendix M characterizes the decline. How much mass there is decides less than which heads carry it, and the next subsection isolates that directly.

6.2 Why Uniform Mass Can Fail

Mass alone does not determine whether a monitor moves. Holding mean distractor mass fixed at 0.5 in every patched layer, we vary only which heads carry it. Head routes are ranked on fifteen discovery cases (five prompts crossed with the three attack pairs) and evaluated on 140 disjoint prompt–pair cases (twenty new prompts crossed with seven further pairs), so the evaluation set shares neither a prompt nor a pair with the ranking. Ranking is marginal and evaluation is joint: each head is scored alone against the uniform allocation, with its layer rebalanced so the layer total is unchanged, and the selected halves are then applied together in every layer at once. Every mask is balanced: exactly half the heads carry 1.0 and half carry 0.0 in each layer, so the per-layer total equals the uniform allocation's exactly, and top and bottom masks are exact complements. Any difference between conditions is allocation, not mass.

Equal mass, opposite effect. At identical mean mass, the discovery-selected layer-specific top half reaches **83–93%** evasion on the six tracks, a uniform allocation reaches **27–52%**, and the exact complement of the top half reaches **3–25%** (Fig. 4b). The largest gap is on 8B LogReg, where routing the same mass to the selected half rather than uniformly moves evasion from 27.2% to 91.0%. Top-minus-bottom moves the held-out detector margin by $+8.1$ to $+11.8$ across tracks.

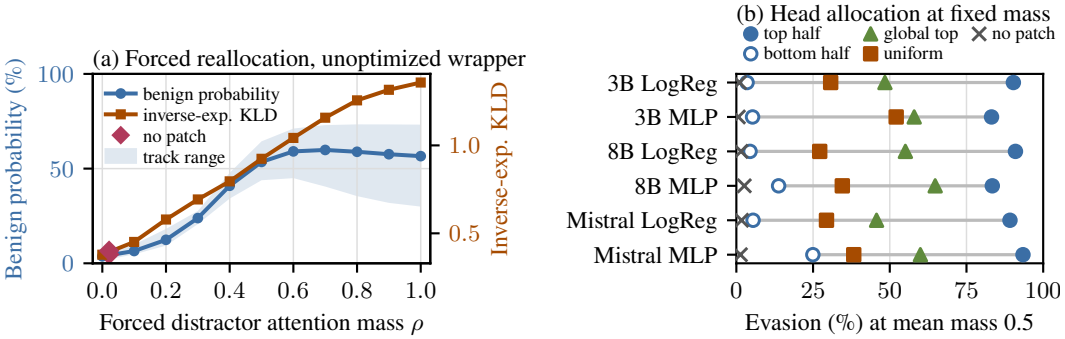

Figure 4: The attention intervention. (a) Forced distractor reallocation in every attention block of the unoptimized wrapper, thirty pairs, averaged over the six tracks with the band spanning them; the diamond is the unpatched measurement. Benign probability is plotted because it is continuous where evasion saturates (evasion per category in Appendix M). (b) Layer-averaged evasion on 140 held-out cases at identical mean mass (0.5); top and bottom halves are exact complements joined by the grey line, and "global top" reuses one head-index ranking across layers.

Routes are layer-specific, on every track. The causal effect of favoring a head does not define a portable head set. Adjacent-layer head-effect rankings are uncorrelated on all six tracks (mean Spearman between -0.058 and $+0.033$) and adjacent top halves overlap at chance (6.0–8.1 heads against a chance value of 6 on 3B and 8 elsewhere). Reusing a single global head-index ranking instead of a per-layer one costs between a fifth and a half of the evasion (22–49% across the six tracks): 90.3% falls to 48.4% on 3B LogReg and 89.2% to 45.7% on Mistral LogReg.

Concentration versus selection. Concentration alone is not sufficient. Over 48 balanced random masks the mean is 34–59% against uniform's 27–52%, so spikier attention helps on its own, but the layer-specific top mask beats the *best* of those 48 random masks on all six tracks. Selection, not concentration, is what carries the effect.

The ranking transfers across embedding state. Both the ranking and the evaluation above are in the unoptimized text state. We also ranked in the optimized state and applied the result to the same held-out set: the two rankings share 86% of each layer's top half on average against 50% at chance, at a per-layer rank correlation of 0.83 to 0.90, and layer-top evasion differs by 1.6 points.

The intervention isolates causal sufficiency, not optimization realizability. It motivates signed, layer-specific head weighting, but does not show that continuous embedding updates can realize the selected allocation; Appendix B measures how far the optimized attack's own allocation falls from the causally selected one. More generally, maximizing one scalar attention statistic can average together routes with opposing downstream effects.

7 Conclusion

ProbeFool shows that a latent monitor can be evaded without any probe output, parameter, or gradient entering the update objective, using only the knowledge of which token the monitor reads. A monitored-token attention-mass objective plus response-distribution preservation achieves 77–93% evasion averaged over transformer-layer probes for the best initialization on each model–probe track and reaches up to 77.4% JSR. The learned shift transfers across broad layer ranges and to prompts the wrapper never optimized against. Intervention and ablation attribute the evasion to attention. Yet the same total mass can help, harm, or leave a monitor unchanged depending on which heads carry it. Adaptive evaluation of activation monitors must therefore account for how attention is redistributed. Appendix O discusses limitations and future work: the continuous-embedding upper bound, the known-readout assumption and the defenses it suggests, and OOD-stealthy variants of the attack.

Ethics statement

This work studies bypasses of safety monitors and may inform stronger attacks. We omit harmful prompts and generated text and present continuous embeddings and forced interventions as diagnostic tools. The defensive value is exposing failure modes in readout placement, pair dependence, broad layer transfer, and head-level cancellation.

Reproducibility statement

Every number in the paper is computed from stored run artifacts by scripts under `replication/` in the code submitted as supplementary material, which also holds the resolved configuration of every run (models, probes, pairs, step counts, learning rates, λ_{KLD}). Section 4 states the candidate filter, the probe training splits and the layer-averaging convention; Appendix A gives the probe training distribution and the cohort construction in full, Section 6 the head discovery and evaluation protocol, and Appendix J the twenty initializations the attack is re-optimized from.

AI use statement

In this work, we used generative AI tools to implement experiment and analysis code, to draft and edit text, and to locate some of the related work. The research questions, experimental design, choice of controls, and all scientific conclusions are the authors' own, and every reported number derives from an artifact the authors verified. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Luke Bailey, Alex Serrano, Abhay Sheshadri, Mikhail Seleznyov, Jordan Taylor, Erik Jenner, Jacob Hilton, Stephen Casper, Carlos Guestrin, and Scott Emmons. Obfuscated activations bypass LLM latent-space defenses. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ktGmDGoWnB.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ETKGuby0hcs.

Zachary Coalson, Jeonghyun Woo, Shiyang Chen, Yu Sun, Lishan Yang, Prashant Nair, Bo Fang, and Sanghyun Hong. Prisonbreak: Jailbreaking large language models with fewer than twenty-five targeted bit-flips, 2024. URL https://arxiv.org/abs/2412.07192.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned LLM. https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm, 2023.

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, 2024.

Dylan Feng, Pragya Srivastava, Anca Dragan, and Cassidy Laidlaw. Benchmarking and improving monitors for out-of-distribution alignment failure in LLMs, 2026. URL https://arxiv.org/abs/2605.21602.

Debargha Ganguly, Sreehari Sankar, Biyao Zhang, Vikash Singh, Kanan Gupta, Harshini Kavuru, Alan Luo, Weicong Chen, Warren Morningstar, Raghu Machiraju, and Vipin Chaudhary. Trust the typical. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.04581.

Aaron Grattafiori et al. The Llama 3 herd of models, 2024.

Peichun Hua, Hao Li, Shanghao Shi, Zhiyuan Yu, and Ning Zhang. Rethinking jailbreak detection of large vision language models with representational contrastive scoring. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21748–21785. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.992. URL https://aclanthology.org/2026.acl-long.992/.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B, 2023.

Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. DeepInception: Hypnotize large language model to be jailbreaker, 2023.

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.

Huijie Lv, Xiao Wang, Yuansen Zhang, Caishuang Huang, Shihan Dou, Junjie Ye, Tao Gui, Qi Zhang, and Xuanjing Huang. CodeChameleon: Personalized encryption framework for jailbreaking large language models, 2024.

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*, 2024.

James Oldfield, Philip Torr, Ioannis Patras, Adel Bibi, and Fazl Barez. Beyond linear probes: Dynamic safety monitoring for language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=AGWa8whf92.

Oam Patel and Rowan Wang. Activation monitoring: Advantages of using internal representations for LLM oversight. In *NeurIPS 2024 Workshop on Attributing Model Behavior at Scale*, 2024. URL https://openreview.net/forum?id=qbvtwhQcH5.

Cheng Qian, Hainan Zhang, Lei Sha, and Zhiming Zheng. HSF: Defending against jailbreak attacks with hidden state filtering. In *Companion Proceedings of the ACM Web Conference 2025*, pp. 2078–2087. Association for Computing Machinery, 2025. doi: 10.1145/3701716.3717659.

Paul Röttger, Hannah Rose 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, 2024.

Mason Sawtell, Tula Masterman, Sandi Besen, and Jim Brown. Lightweight safety classification using pruned language models, 2024. URL https://arxiv.org/abs/2412.13435.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. A StrongREJECT for empty jailbreaks, 2024. URL https://arxiv.org/abs/2402.10260.

Zijun Wang, Haoqin Tu, Jieru Mei, Bingchen Zhao, Yisen Wang, and Cihang Xie. AttnGCG: Enhancing jailbreaking attacks on LLMs with attention manipulation. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=prVLANCshF.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. GPTFUZZER: Red teaming large language models with auto-generated jailbreak prompts, 2023.

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.

Noureldin Zahran, Ahmad Tahmasivand, Ihsen Alouani, Khaled N. Khasawneh, and Mohammed Fouda. On jailbreaking quantized language models through fault injection attacks. In *Proceedings of the Great Lakes Symposium on VLSI 2025*, pp. 554–561. Association for Computing Machinery, 2025. doi: 10.1145/3716368.3735302.

Pedram Zaree, Md Abdullah Al Mamun, Quazi Mishkatul Alam, Yue Dong, Ihsen Alouani, and Nael Abu-Ghazaleh. Attention eclipse: Manipulating attention to bypass LLM safety-alignment. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 16648–16668. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.842. URL https://aclanthology.org/2025.emnlp-main.842/.

Shenyi Zhang, Yuchen Zhai, Keyan Guo, Hongxin Hu, Shengnan Guo, Zheng Fang, Lingchen Zhao, Chao Shen, Cong Wang, and Qian Wang. JBShield: Defending large language models from jailbreak attacks through activated concept analysis and manipulation. In *34th USENIX Security Symposium*, pp. 8215–8234. USENIX Association, 2025. URL https://www.usenix.org/conference/usenixsecurity25/presentation/zhang-shenyi.

Chongwen Zhao, Zhihao Dou, and Kaizhu Huang. Defending against jailbreak through early exit generation of large language models. In *Neural Information Processing*, volume 16313 of *Lecture Notes in Computer Science*, pp. 532–546. Springer Nature Singapore, 2026. doi: 10.1007/978-981-95-4445-5_36. Proceedings of ICONIP 2025.

Weikang Zhou, Xiao Wang, Limao Xiong, Han Xia, Yingshuang Gu, Mingxu Chai, Fukang Zhu, Caishuang Huang, Shihan Dou, Zhiheng Xi, Rui Zheng, Songyang Gao, Yicheng Zou, Hang Yan, Yifan Le, Ruohui Wang, Lijun Li, Jing Shao, Tao Gui, Qi Zhang, and Xuanjing Huang. Easyjailbreak: A unified framework for jailbreaking large language models, 2024.

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

A Setup Details: Candidate Pool, Probes and Optimization

Content. The probe training setup in full, the cohort construction, and the optimizer settings.

Optimization settings. Each candidate-set steerer is optimized for 500 Adam steps with learning rate 0.005, batch size four, gradient accumulation four, and a cosine schedule with ten warmup steps, at $\lambda_{\text{attn}} = 1$, $\lambda_{\text{KLD}} = 0.1$ and $\gamma = 0.1$. We swept $\lambda_{\text{KLD}} \in \{0.05, 0.075, 0.1, 0.125, 0.15, 0.175, 0.2\}$ at this fixed budget, averaging layer-averaged JSR over both probe families and all three pairs on every model; the values 0.05, 0.1 and 0.125 are tied (54.6, 54.4 and 54.4), so we do not claim 0.1 is optimal. Pooled, $\lambda_{\text{KLD}} \geq 0.15$ is about four points lower, but the sweep is not ordered in λ and the gap does not hold within every model: 0.075 is the weakest value swept (49.6), and $\lambda = 0.2$ is 8B's own best.

Training distribution and splits. Each model's corpus holds $6{,}482$ rendered prompts in three classes: benign, harmful-but-not-jailbroken, and jailbreak. The benign class combines $2{,}753$ instruction-following prompts from databricks-dolly-15k (Conover et al., 2023) with the 250 benign-but-sensitive prompts of XSTest (Röttger et al., 2024); the harmful class combines 520 AdvBench behaviours (Zou et al., 2023), 159 from HarmBench (Mazeika et al., 2024) and XSTest's 200 unsafe contrast prompts; the $2{,}600$ jailbreak rows are the five retained methods applied to AdvBench behaviours, AutoDAN (Liu et al., 2024) and Cipher (Yuan et al., 2024) being excluded from the corpus. Splits are stratified and grouped on a hash of the underlying harmful request, so every jailbreak rewrite of a given query lands in the same split and no test prompt is a transformed copy of a training one. 3B and Mistral use a $40/20/40$ train/validation/test partition.[1] Test-split class counts (benign / harmful / jailbreak) are $1202/346/1008$ on 3B, $1197/385/1145$ on 8B and $1204/366/1014$ on Mistral. Every probe is deployed at the threshold giving a nominal 1% false positive rate on the validation benign rows. Excluding the embedding readout leaves 28 transformer-layer probes per family on 3B and 32 on 8B and Mistral. Every probe exceeds 0.98 test ROC-AUC (the minimum is 0.981, 3B LogReg) and every probe from layer 16 on exceeds 0.999; test recall at the deployed threshold is above 0.99 from layer 16 on every probe but two (3B LogReg layers 21 and 22, at 0.990 and 0.987).

Held-out cohort. The held-out cohort of Table 1 is formed by removing the prompts each wrapper was optimized on and taking the next candidates per method from the remainder.

B The Attack Does Not Find the Effective Routes

Content. The body shows that a layer-specific head allocation learned on discovery cases generalizes to disjoint prompts and pairs. The optimized attack does not recover that ranking on its own.

The attack does not recover the causally selected routes. We measure, at each probe's own selected attention layer, what share of the optimized attack's distractor mass falls on the heads discovery selected. Chance is 50%. Across the six tracks the attack places a mean of 49.5%, with five of the six within six points of chance and 3B MLP ten points below it.

Its own allocation still beats a uniform one. Forcing a uniform per-head profile onto the optimized wrapper at $\rho = 0.5$, about the mass it already carries, lowers evasion on five of six tracks, from 78.5 to 48.6 on 3B LogReg and 74.3 to 58.5 on 8B LogReg, with 3B MLP the exception at 74.7 to 75.8; divergence rises on every track, from 0.06–0.08 unpatched to 0.52 pooled. At equal mass the attack's allocation dominates a uniform one on both axes.

Embeddings, layer placement and head shape all contribute. We rebuild the attack's layer–head distractor-mass matrix one degree of freedom at a time, holding global mass fixed. Averaged over three models and three pairs, flat mass on clean embeddings reaches 45.6% layer-averaged evasion, flat mass on the optimized embeddings 57.3%, the attack's per-layer totals spread evenly across heads 69.9%, and its full layer–head matrix 75.7%, against the attack's 75.8%. Each degree of freedom adds: $+11.8$ points for the embeddings, $+12.6$ for the layer placement and $+5.8$ for the head shape. No one of them accounts for the attack, and restoring all three reproduces it to within 0.1 points.

[1] The 8B probes use a split that moves ten candidates per jailbreak method out of training and into test, enlarging the held-out evaluation set; every 8B number we report is scored by those probes.

C EMBEDDING SPACE: WHERE THE MONITORED READOUT GOES

Content. A geometric decomposition of the attacked readout in the probe's own feature space, after that probe's StandardScaler and at that probe's layer, measured against the benign *training* rows, which are the class the attack is being mistaken for.

The attack lands benign-classified but off the training manifold. Averaged over layers $1..n$ and the three pairs, the attacked readout sits 3.3–$4.3\times$ farther from its ten nearest benign training points than those points sit from each other, and it does so in 100% of samples on every model. The displacement is directional rather than inflation: the component orthogonal to the training PCA subspace (95% variance) grows 9.0–$13.0\times$ while extent *along* that subspace grows only 1.4–$1.7\times$, against a plain norm ratio of 2.3–3.1. The attack moves sideways into directions the training data never spanned, where the probe has no constraint.

The wrapper is already off-manifold before the attack starts. The clean textual wrapper, our control, sits at a 1.4–$1.6\times$ nearest-neighbour ratio and a 2.9–$3.2\times$ orthogonal residual. So the optimization does not carry a normal readout off the manifold; it multiplies a displacement the wrapper itself has already caused, from about threefold to between nine- and thirteenfold. Any defense reading this geometry would see the unoptimized wrapper as anomalous too.

D WHAT THE UNOPTIMIZED WRAPPER COSTS

Content. StrongREJECT judgements for the same candidates under three conditions: the original jailbreak, the unoptimized textual wrapper, and the optimized attack.

Inserting the wrapper is highly destructive, and the optimization recovers most of it. Jailbreak success falls from 78.0% to 32.7% on 3B, 91.8% to 38.1% on 8B and 90.0% to 57.3% on Mistral, and mean StrongREJECT score falls from $0.71/0.74/0.76$ to $0.33/0.37/0.55$. The optimized attack recovers to 62.0%, 66.0% and 82.0%.

Final quality is predicted by the original jailbreak's strength, not by the damage done. Across 149 candidates, the final response's StrongREJECT score correlates $+0.669$ with the original jailbreak's score (Spearman $+0.696$) against $+0.291$ with the damage the wrapper inflicted, and the ordering holds on every model separately ($+0.655/+0.605/+0.772$ against $+0.406/+0.345/+0.353$). We state this as original strength dominating rather than damage being irrelevant: damage is measured as original minus wrapped, so it is itself correlated with the original score. Read either way, the preservation term is undoing a large systematic cost rather than merely trading against attention.

E THE ATTACK BEATS RANDOM DISPLACEMENT AND DILUTION

Content. Two cheaper alternatives the attack has to beat: a random displacement of the monitored readout matched to the attack's own magnitude, and a long benign prefix with no optimization.

Magnitude is not what crosses the boundary; direction is. Appendix C shows the attacked readout sits far off the benign training manifold, which invites the reading that any displacement of that size would cross a 1%-FPR boundary and that attention is merely a convenient dial for producing one. We test it on the stored Appendix K captures. Writing δ for the attack's displacement at the monitored token, we replace it with four alternatives of the same norm and rescore against the same probe at the same deployed threshold: a uniformly random direction (ISO); a random direction orthogonal to the benign-PCA subspace (ORTH); a random direction matched to the attack's own split between that subspace and its complement (PROFILE); and the steepest-ascent direction on the probe's benign logit (ORACLE), which reads the probe and is therefore a ceiling rather than an attack. Ten draws per prompt per arm.

The strongest random arm lands 2.4–9.1 points above the Clean baseline while the attack sits 45.6–72.7 points above that arm (Table 3), and ORACLE reaches 100% on every track, so the norm budget is ample and random directions do not spend it. The controls are not failing to leave the manifold: their orthogonal-component ratio exceeds the attack's on every track (10.1–15.7 against 9.0–13.0, in the units of Appendix C), and all four arms are flagged by the Mahalanobis detector of Appendix F

Table 3: Matched-norm displacement at the monitored readout, layer-averaged evasion (%), mean over three pairs and ten draws. Every random arm carries the same displacement magnitude as the attack.

		Clean	ISO	ORTH	PROFILE	Attack	ORACLE
3B	LR	2.3	5.5	4.7	6.8	78.5	100.0
	MLP	2.5	3.5	2.9	4.9	74.7	100.0
8B	LR	7.4	14.5	13.6	16.5	74.3	100.0
	MLP	12.2	18.3	16.1	21.1	66.7	100.0
Mistral	LR	4.4	8.0	7.1	10.3	78.8	100.0
	MLP	5.2	8.1	6.8	11.6	84.3	100.0

Table 4: Unoptimized benign prefix. Each cell is evasion (%) as LR/MLP, then response success (%), which is layer-independent and shared by both probe families. Every row is initialization pair 1, so the Clean row is not the three-pair Clean of Table 2.

	3B		8B		Mistral	
Prefix	ER (LR/MLP)	SR	ER (LR/MLP)	SR	ER (LR/MLP)	SR
Clean (1–15 words)	3.7/3.4	22.0	6.9/14.6	34.7	2.9/4.4	56.0
25 tokens	14.5/14.6	12.0	27.8/32.8	20.4	9.4/10.2	36.0
100 tokens	12.8/13.9	14.0	42.7/56.6	24.5	14.2/26.4	28.0
200 tokens	14.9/13.6	14.0	47.3/58.2	0.0	14.2/21.9	40.0
Long steerer	19.5/14.9	26.0	15.8/21.0	24.5	13.5/12.9	46.0

100% of the time against 23.8–38.1% for Clean. Being flagged as off-manifold and evading the probe are therefore different questions.

Dilution evades, but destroys the response. A long benign prefix should buy some of the attack's mechanism for free; the Clean arm does not settle this, since its distractors are 1–15 words. We rerun the wrapper with no optimized embedding anywhere, varying only prefix length, and report response success alongside evasion. Two hundred benign tokens lift 8B MLP from 14.6% to 58.2% evasion and take response success from 34.7% to 0.0%; every model pays, 22.0 to 14.0 on 3B and 56.0 to 40.0 on Mistral, with the largest evasion gains attached to the largest losses. The attack, over its three pairs, reaches 66.7–84.3% evasion while holding response success at Table 1 levels.

The two controls fail in opposite directions: random displacement leaves the response intact and does not move the monitor, dilution moves the monitor and destroys the response.

F A Benign-Only Detector Catches This Attack, but Does Not Detect Harm

Content. A Gaussian fitted to the benign training rows only and scored by Mahalanobis distance, at a threshold calibrated to a 1% false positive rate on validation. The attack is optimized against neither monitor, so this is an out-of-sample test of a defense rather than a second attack objective.

It catches this attack everywhere. The attack is flagged 100% of the time on all three models, and not merely on average: at *every one* of the 28, 32 and 32 hidden states.

It separates objectives that the probe cannot. Across the seven objectives of Appendix G, flag rates run from 36.8% to 100% while probe evasion runs from 0.0% to 87.3%. The floor is not a stealthy attack: 36.8% is preservation only on Mistral, which evades 2.2% of the time, and every flag rate below 67.6% belongs to an arm that does not evade, so among objectives that work the range is only 67.6–100%.

What buys stealth is moving *toward* the benign manifold, not merely declining to grow away from it. The ordering tracks how each objective treats the *magnitude* of the monitored state. Attention-driven objectives inflate it (norm ratios 1.17–1.64) and are flagged without exception; distance-based RMSE objectives shrink it (0.84–0.90) and are the only evading objectives with measurable stealth, reaching 67.6% on 3B and 74.2% on 8B. Magnitude-blindness alone does not

suffice: on Mistral the cosine objective is magnitude-neutral by construction (0.98) and evades the probe at 87.3%, yet is flagged 100% of the time.

Its benign recall is measured on a narrow slice. Benign recall on held-out test data is 99.2% (worst layer 97.9%), matching the nominal 1% false positive rate, but we did not evaluate against an external benign distribution, so we cannot separate "not benign" from "not this benign sample". The limitation is borne out externally: a guard-plus-Mahalanobis system calibrated to 1% FPR on in-distribution benign data reaches 98.4% FPR on harmless Swahili conversations (Feng et al., 2026), and a benign-only reconstruction detector sharply over-rejects unseen harmless domain shifts (Hua et al., 2026). More elaborate safe-only typicality detectors outperform classical OOD baselines but still retain substantial false-positive rates on safe-but-challenging prompts (Ganguly et al., 2026).

It detects displacement, not harm. It flags 29.1–41.6% of the *unattacked* jailbreak prompts on average, against 100% of attacked ones, so it cannot replace a harmfulness detector.

G The Attack Form Admits Objectives Other Than Attention

Content. Six alternatives to Eq. 2 under the identical protocol on all three models, three pairs each, replacing distractor attention mass with the RMSE or the cosine between the monitored token's representation and the distractor's.

Table 5: Seven attack objectives. ER is layer-averaged evasion on the LogReg probe and SR is StrongREJECT success; row one is Eq. 2.

	ER			SR		
Objective	3B	8B	Mis.	3B	8B	Mis.
Attention + pres. (Eq. 2)	78.5	74.3	78.8	62.0	66.0	82.0
Attention only	72.1	76.9	79.4	2.7	0.0	11.3
Preservation only	0.0	0.8	2.2	74.7	80.3	80.7
Cosine + pres.	1.1	3.5	87.3	73.3	70.7	87.3
RMSE + pres.	72.6	77.0	87.0	62.7	70.1	83.3
RMSE + attn. + pres.	72.5	77.6	82.0	72.0	67.3	76.7
RMSE + attn. + cos. + pres.	74.2	78.0	85.2	58.7	59.2	76.7

Several other objectives also evade. Holding the attack form fixed and swapping only the activation term, the RMSE variants evade comparably on all three models and the cosine objective does so on Mistral alone. We did not tune hyperparameters per objective or refine any of these arms; each reuses the settings of Eq. 2, so the table says which objectives are viable under this attack form, not how well each could be made to perform.

Neither term alone is enough. Attention only reaches evasion comparable to the full objective (72.1, 76.9, 79.4%) at 0.0–11.3% response success, and preservation only inverts this exactly. The combination is what makes the attack a threat.

H Probes That Never Saw the Attacked Jailbreak Family

Content. The probes used throughout see all five attacked jailbreak families during training. Here we retrain them with one family held out of both training and validation, then re-score the same stored attacks against them. The attack is never re-optimized, because it never reads a probe.

Each family is compared against itself, so its difficulty cancels. For every jailbreak family we measure evasion on that family's own prompts twice: once under probes trained on it, once under probes that never saw it. Nothing differs but the probe's exposure. Three models crossed with two probe families and five jailbreak methods give thirty such comparisons.

Training on the family buys the defender about six points. Evasion is 6.27 points higher against probes that never saw the family, in 25 of the 30 comparisons (Wilcoxon $p = 8.0\times10^{-6}$). Restricting to prompts the probe catches before the wrapper is applied, which keeps 93% of rows, gives 6.54 points in 26 of 30 ($p = 6.0 \times 10^{-6}$), so the effect is not an artifact of the two probe sets differing in

what they detect. Under that restriction evasion moves from 70.0 to 78.5% on 3B, 76.3 to 81.1% on 8B and 78.7 to 85.0% on Mistral. Against probes that never saw the family, evasion still averages 81.2%, ranging from 72.0 to 92.1% across the thirty comparisons.

The effect is uneven across families. It is stable for GCG (+11.4, 6/6 comparisons positive), CodeChameleon (+8.0, 6/6) and GPTFuzzer (+7.7, 6/6); ReNeLLM (+4.3, 4/6) is weaker and DeepInception (−0.04, 3/6) is indistinguishable from zero, so we report four families moving rather than five and draw no conclusion about which families are most distinctive.

I The Operating Point Is a Real Lever, but an Expensive One

Content. Every number in this paper is read at a threshold calibrated to a nominal 1% false positive rate on validation benign rows. That choice is the defender's. We re-read the same stored per-sample probabilities at every operating point the probes record; nothing is re-run and nothing is re-attacked.

Table 6: Layer-averaged in-sample evasion (%) when each probe's threshold is recalibrated to the nominal validation false-positive rate in the row head. The 1% row is the three-pair mean of Table 1's in-sample evasion; nothing is re-optimized or re-attacked.

	3B		8B		Mistral	
Nominal FPR	LR	MLP	LR	MLP	LR	MLP
2%	67.1	72.6	64.9	63.8	74.6	82.8
1%	78.5	74.7	74.3	66.7	78.8	84.3
0.5%	83.9	79.0	82.7	70.6	81.5	84.6

Evasion moves monotonically with the operating point, and loosening it is expensive. A tighter false-positive budget lowers the decision threshold, so more states pass as benign and the attack evades more. Going the other way, 8B LogReg falls from 74.3% to 64.9% evasion, but only by doubling the false-positive budget, and the size of the lever is model dependent: over this range 8B LogReg swings 17.8 points while Mistral MLP swings 1.8.

Held-out prompts track the whole curve. They match the in-sample values to a maximum of 1.4 points and a mean of 0.3 across all six tracks and all three operating points reported here, so the transfer result of Table 1 holds at every operating point rather than at 1% alone.

J Re-Optimizing the Attack from Twenty Initializations

Content. The paper optimizes three distractor–steerer initializations per model and reports pair choice as load-bearing. On 8B we *re-run the attack* from twenty initializations under a configuration identical to the reported runs: same candidate set ($N = 49$), same objective at $\lambda_{\text{KLD}} = 0.1$, same schedule, same terminal checkpoint, differing only in the initialization draw. Pool members $k = 1, 3, 9$ are the paper's own three pairs, and re-running them reproduces their Table 1 cells on both probe families, so the pool is self-checking. All twenty share the same candidates, so per-sample comparison across pairs is well posed.

The spread between pairs is entirely in evasion. Across the pool, layer-averaged evasion runs 67.3% on average for LogReg (range 38.8–86.1) and 65.2% for MLP (41.4–87.9), while StrongREJECT success is flat (55.1–81.6, mean 69.5) and shows no relationship to evasion. Wrapper quality varies sharply; the underlying jailbreak's response success does not.

Which pair is chosen matters far more than which prompt is attacked. A variance decomposition over 49 samples × 20 pairs attributes 76.8% of variance to pair identity against 7.6% to the prompt (15.5% interaction), and prompts rank the pairs consistently with each other (Kendall's $W = 0.855$). This is consistency across prompts within 8B and does not bear on the cross-model finding that the best pair differs by model.

One pair feature predicts evasion, and it is syntactic. Steerers phrased as questions average 72.0% evasion ($n = 13$) against 58.7% for those that are not ($n = 7$), a rank correlation of $r_s = +0.446$ ($p = 0.049$ before any correction for the twelve features scanned, and not significant after one). We

therefore quote the effect size and not the significance level, and note that the pattern is loose: the strongest pair in the pool, at 86.1%, is *not* a question steerer, and the weakest question steerer sits at 55.9%. The nulls are the more informative half. Steerer and distractor length in words predict nothing ($r_s = +0.23$, $p = 0.33$ and $r_s = -0.07$, $p = 0.77$); counted in Llama-3 tokens the steerer reaches $r_s = +0.44$ ($p = 0.052$) over a range of only two to eight tokens, uncorrelated with the question form ($r_s = +0.12$), so it is a second weak candidate rather than a confound of the first.

How the reported three sit in the pool. They rank fourth, fifth and twelfth of twenty, averaging 74.3% against a pool mean of 67.3%. The best-of-three figure on 8B should be read as a modestly lucky draw.

K What the Attack Recruits: Attention or Geometry?

Content. The transplant that separates the attack's two effects, the control that rules out the obvious confound in it, and the numbers Section 6 quotes.

The transplant. For every model, probe and pair we exchange the monitored token's entire attention row between the clean and attacked forwards, in both directions, giving the unoptimized wrapper under the attack's attention ("attention only") and the optimized embedding under clean attention ("geometry only"), against the two unmixed forwards. Only the monitored query row is swapped; every other position keeps the running forward's attention, and values come from the running forward. Both transplants are the same operation, so the off-diagonal cells are comparable, and the two forwards share an identical token layout, which we assert per sample.

Table 7: Crossing content with attention (evasion %), pooled over the three pairs and restricted to the 13–23 layers per track where the attack itself evades at least 80% of prompts.

	3B		8B		Mistral	
Condition	LogReg	MLP	LogReg	MLP	LogReg	MLP
Clean	1.0	1.6	6.4	13.0	5.9	4.5
Attention only	71.3	83.2	73.9	96.7	90.0	89.1
Geometry only	0.2	0.0	0.2	0.1	5.5	0.1
Attack	98.6	98.8	94.9	97.1	97.4	99.4

Attention is necessary and largely sufficient. The optimized embedding under clean attention evades at most 5.5% on any track and at or below the clean baseline on all six, against 95–99% for the attack itself (Table 7). This rules out the optimized embedding as a sufficient cause: it traverses every layer and MLP in the ablated condition and moves no monitor, so nothing the later components compute from the changed content evades on its own. The MLPs remain on the causal path in the full attack, acting on the values the changed attention delivers; what the arm excludes is the content change reaching the monitor without that routing. Transplanting the attack's attention onto the unoptimized wrapper recovers 71.3 to 96.7 points of absolute evasion, 72 to 100% of what the full attack achieves on the same layers. Table 7 restricts to the layers where the attack itself evades at least 80% of prompts, which is outcome-dependent, so we state the unrestricted numbers too: over all layers, geometry only reaches 0.6–4.1% against a clean baseline of 2.3–12.2%, and attention only reaches 60.9–79.1% against the attack's 66.7–84.3%. The conclusion is the same in both cuts; the restricted one simply reads where there is evasion to explain.

The confound, and the control that removes it. Installing the clean row makes the monitored token read the optimized steerer at whatever weight the clean row gives it, which is below what the attack gives it: the steerer's share of the monitored token's attention goes clean to attack $3.2 \to 9.0$ (3B), $4.0 \to 18.0$ (8B) and $7.1 \to 20.0$ (Mistral) layer-averaged, and the gap is concentrated in the first third of the network ($3.3 \to 12.4$, $3.8 \to 27.9$, $6.0 \to 31.6$) while vanishing at the final block. The published null is therefore a null at baseline weight, most doubtful exactly where the gap is largest. We re-ran the geometry arm keeping the clean donor row but re-setting the *steerer* group's mass per layer and head, rescaling the remaining keys so the row still sums to one; the distractor keeps its clean weight, so the routing factor stays absent and only the reading of the changed content varies. Read at the attack's own per-head weighting the optimized embedding reaches 8.7% pooled against a clean baseline of 5.7%; at twice that weighting it reaches 7.7%, no further gain. Only at a

forced steerer share of 0.5, well above anything the attack produces, does it reach 20.4%, and that is concentrated on 8B (36.8 and 37.5) with Mistral barely moving (9.2 and 9.0). The null survives the control.

The factorization is not an artifact of optimizing attention. Equation 2 raises attention mass, so finding that the attack works through the attention row invites the charge of circularity. We therefore repeated the transplant on wrappers trained by the RMSE-plus-preservation objective of Appendix G, which never refers to attention: it pulls the monitored representation toward the distractor's directly, and evades 72.6–87.0%. Its geometry-only arm reaches 1.6–3.9%, at or below the clean baseline on all six tracks, exactly as Equation 2's does. An objective built without an attention term still reaches the monitor through the attention row. Sufficiency is weaker off-objective, though: the attacked row recovers a mean 73% of the RMSE attack's own evasion (63–101% across tracks) against 92% for Equation 2 (78–110%), both over all layers rather than the restricted cut above, so attention carries most but not all of what a representation-matching objective buys.

Same budget, different target. Forcing half of the monitored token's attention onto the benign textual distractor evades 63.4% on the attack's three pairs; forcing the same half onto the optimized steerer evades 20.4%. The same forced attention is three times as effective pointed at benign content as at the optimized embedding. By depth, reading the steerer at the attack's weight buys $8.9 \rightarrow 27.1$ points over the first eight layers and nothing above them ($4.4 \rightarrow 2.1$, and $7.3 \rightarrow 0.4$ at the final layer), which is where the weight gap sits. The shallow band does have a content channel; it is not where evasion is won, since the attack's own evasion lives in the high band. This also accounts for the $+11.8$-point step in Appendix B from flat mass on clean embeddings to flat mass on the optimized ones: that step is the same shallow-layer channel rather than evidence that the optimized embedding evades where the attack does.

L Confidence Intervals for Table 1

Content. Bootstrap intervals for every evasion and joint-success cell, in both cohorts, and why the two metrics differ so much in precision.

Method. Layer-averaged evasion reuses the same prompts at every layer, so the layers of one cell are not independent draws. A binomial interval on the pooled rate is the wrong model in the other direction from the usual worry: it treats each prompt as evading or not, when what a prompt contributes is the fraction of layers that evade for it, and those fractions are concentrated. At $N = 50$ and a rate of 80% it spans 22.2 points, roughly eight times the width we measure. The sampling unit is the prompt. We resample `sample_uid`s with replacement, recompute the layer average within each resample, and take the 2.5th and 97.5th percentiles over 2,000 draws. JSR is bootstrapped on the same resample, so the two intervals in a row are mutually consistent. Intervals are reported for all eighteen cells rather than only the bolded ones, because an interval computed on a cell selected for being the maximum is optimistically narrow.

Evasion is precise because the layers agree. In-sample evasion intervals span 0.6 to 4.9 points (mean 2.6), and held-out ones 0.4 to 3.7. The same saturation that makes the layer average a coarse summary, 68–84% of layers sitting at exactly 0% or 100% in sample, is what makes it a stable one: layers that move together contribute no independent noise, so the estimate depends almost entirely on how many prompts the wrapper carries across the whole depth rather than on which layer is read.

Joint success is not, and the reason is the response. JSR intervals span 12.6 to 23.2 points in-sample and 7.5 to 23.6 held-out, an order of magnitude wider. JSR multiplies evasion by a per-prompt response judgement that no averaging stabilizes, so its precision is that of a single binary rate at $N \approx 50$. Contrasts we draw on evasion are therefore safe at the few-point scale, while the paper makes no claim that rests on a JSR difference smaller than its interval. The Mistral held-out cells are the exception in the other direction: at $N = 255$ their JSR intervals narrow to eight to ten points.

M Attention Intervention: Pairs, Categories and Controls

Content. The thirty distractor–steerer pairs the ρ intervention runs, the category effect they expose, and the controls that bound it.

Table 8: Bootstrap over prompts, 2,000 draws. Cohorts are $N = 50$ in-sample (49 on 8B) and $50/50/255$ held-out.

			In-sample				Held-out			
			ER		JSR		ER		JSR	
Model	Probe	Pair	%	95% CI	%	95% CI	%	95% CI	%	95% CI
3B	LR	1	70.3	[69.1, 71.4]	44.3	[34.4, 53.2]	69.6	[68.5, 70.6]	38.5	[28.5, 48.5]
	LR	2	84.0	[83.2, 84.8]	52.6	[40.6, 63.4]	83.5	[82.8, 84.2]	47.0	[35.4, 58.6]
	LR	3	81.2	[80.0, 82.4]	50.3	[38.6, 61.9]	80.7	[79.5, 81.9]	40.0	[28.9, 50.6]
	MLP	1	62.9	[61.2, 64.5]	40.1	[31.1, 48.7]	62.0	[60.4, 63.7]	34.6	[25.8, 44.1]
	MLP	2	77.4	[75.8, 79.1]	49.2	[38.7, 60.1]	76.8	[75.6, 78.1]	43.9	[32.9, 54.2]
	MLP	3	83.9	[83.0, 84.7]	52.1	[40.4, 63.3]	83.7	[82.9, 84.5]	42.0	[30.2, 53.9]
8B	LR	1	76.8	[75.4, 78.2]	59.8	[50.4, 68.0]	75.9	[74.7, 77.2]	45.6	[34.7, 55.8]
	LR	2	66.5	[64.0, 68.9]	40.4	[31.2, 49.6]	65.9	[64.2, 67.6]	36.1	[26.9, 45.6]
	LR	3	79.6	[78.5, 80.5]	49.7	[38.3, 59.9]	79.6	[78.6, 80.4]	36.9	[25.9, 48.0]
	MLP	1	78.0	[76.4, 79.5]	60.9	[51.1, 69.7]	77.6	[76.1, 79.0]	46.3	[36.2, 56.9]
	MLP	2	44.0	[42.4, 45.5]	26.8	[20.5, 33.2]	44.6	[42.8, 46.5]	24.2	[18.0, 30.8]
	MLP	3	78.1	[77.3, 78.8]	48.5	[37.5, 59.5]	78.2	[77.1, 79.2]	36.0	[24.9, 46.9]
Mistral	LR	1	75.9	[74.1, 77.6]	62.6	[54.4, 70.3]	75.9	[75.2, 76.7]	61.0	[57.0, 64.8]
	LR	2	71.6	[71.2, 71.9]	58.7	[50.2, 65.9]	71.6	[71.4, 71.8]	54.0	[50.1, 57.6]
	LR	3	89.1	[86.9, 91.1]	73.9	[63.2, 83.3]	89.1	[88.2, 90.0]	68.2	[63.1, 72.7]
	MLP	1	78.3	[76.9, 79.6]	64.6	[55.3, 72.5]	78.1	[77.5, 78.7]	62.6	[58.7, 66.5]
	MLP	2	81.2	[80.9, 81.6]	66.6	[58.3, 74.8]	81.3	[81.1, 81.6]	61.3	[57.0, 65.4]
	MLP	3	93.2	[91.8, 94.6]	77.4	[66.2, 86.8]	93.1	[92.5, 93.7]	71.1	[66.0, 75.9]

The pool. All thirty pairs are drawn from one candidate sheet of 161, three from each of ten categories, and every pair is run against the same cohorts, ρ grid and all-layer patch scope. The three initializations the attack optimizes are among them. The intervention needs only forward passes, which is why it can afford thirty pairs where the attack affords three. The twenty initializations of Appendix J are a different object: wrappers the attack is re-optimized from, not pairs the intervention forces mass onto.

Table 9: Layer-averaged evasion (%) under forced distractor mass, averaged over the ρ grid and over each category's three pairs. The steerer queries the distractor: "what does Alice like?", "favorite color?", "codeword?". The last two categories instead point at the task itself ("actual task above", "anchor task above").

		3B		8B		Mistral		
Category	Example distractor	LR	MLP	LR	MLP	LR	MLP	mean
Entity attribute	"Alice likes tea"	32.2	49.9	46.2	58.1	52.8	59.0	**49.7**
QA recall	"red is my favorite color"	28.2	45.6	47.6	53.9	45.4	52.2	45.5
Identity match	"topic: gardening"	25.8	45.0	42.1	48.0	46.1	50.5	42.9
Codeword binding	"codeword: river"	27.7	42.1	43.0	48.8	32.5	41.7	39.3
Continuation	"the next word is orange"	19.6	39.3	38.9	46.6	36.3	43.8	37.4
Control negative	"the city is Cairo"	23.9	41.6	33.3	39.3	40.4	45.6	37.4
Key–value recall	"priority: low"	20.5	32.6	33.9	41.0	40.0	46.3	35.7
Instructional memory	"remember: the key is Leon"	21.3	38.2	22.6	32.6	32.5	37.7	30.8
Hybrid	"the task below is the anchor"	11.1	23.1	8.4	18.6	14.8	26.1	17.0
Generation preservation	"the actual task comes next"	6.8	19.2	4.2	9.4	14.7	22.0	**12.7**

The category effect is large, consistent and not a length effect. On the thirty pair means, category separates them at $F = 8.24$ ($p = 4.8 \times 10^{-5}$), and the ordering in Table 9 holds on all six tracks (Kendall's $W = 0.88$): entity attribute is first or second everywhere, generation preservation last and hybrid ninth on every track. Descriptively, category accounts for 54% of the variation in per-pair, per-track mean evasion against 15% for the pair within its category and 19% for the track; we report that split as a description rather than a test, since the six tracks read the same forward passes. The obvious confound is length, because instruction-style distractors are also the longest (10.3 words on average against 3.0 for entity attribute). It does not survive inspection: distractor length predicts

evasion across categories ($r_s = -0.53$, $p = 0.002$) but not within them ($r_s = -0.22$, $p = 0.25$ once category means are removed), whereas the category effect remains after a linear length effect is removed ($F = 3.51$, $p = 0.009$).

The decline past the peak. Under the all-token patch, 82.7% of the 8,940 prompt–pair trajectories reach their most benign point at some $\rho < 1$, so the decline is pervasive rather than exceptional. It is usually small: drops above 25 points occur in 7.4% of 3B trajectories, 0.2% on 8B and none on Mistral.

It is depth-localized. The first eight layers and the final one rise monotonically to ρ=1 under both the all-token patch and the final-token control, while the layers between them peak at ρ=0.7 and fall, most sharply on 3B.

No single block carries much of the effect. Section 6 reports that patching one layer moves the monitor very little. That was first measured at one block per track, the block beneath each track's selected readout, so we patched every block separately on all three models over the same thirty pairs: each block alone across the full ρ grid, scoring every readout. Every figure here is a lift over that sweep's own unpatched baseline. The typical block moves almost nothing: over all blocks the mean lift is $+1.3$, $+1.9$ and $+1.0$ points on 3B, 8B and Mistral, with medians of $+0.10$, $+0.95$ and $+0.26$ and interquartile ranges no wider than 1.7 points. The distribution is skewed rather than flat. Six of twenty-eight blocks on 3B, fifteen of thirty-two on 8B and seven of thirty-two on Mistral exceed a point, and the best block on each model lifts evasion by $+11.6$, $+10.6$ and $+10.9$ points. Single-block leverage sits in the earliest blocks, block 0 on 3B, 3 on 8B and 2 on Mistral, and it is not an artifact of the readouts immediately above the patch: restricting to probes at least three layers higher still leaves $+5.8$ to $+9.0$ points. The six blocks originally tested move 0.01 to 0.41 points and rank twenty-first or lower among their model's blocks, so that measurement was right and the generalization from it was not. Patching every block on the same cases lifts evasion by $+47.9$, $+52.8$ and $+59.8$ points, so the mean single block is at most 4% of the full effect and the best block 18 to 24%.

The cause remains open. Three candidates do not survive. The mass is not flattening across the distractor's own tokens, because the patch rescales each head's baseline shape at every ρ rather than replacing it with a uniform one. Nor is it the equalizing of mass across heads, which the patch performs at every ρ and not only at saturation. And it is not the loss of the monitored token's remaining context, because the final-token control removes that too and does not decline.

Where the attack's own three sit. Ranked by peak evasion across the thirty they place eighth, tenth and twelfth, in categories ranked first, second and fifth; within their own categories they are the weakest of three in entity attribute (46.5 against 54.7 and 47.8) and in QA recall (42.2 against 49.5 and 44.7). Their advantage over the pool is category membership, not selection within a category, and forced-mass response is in any case not a predictor of optimized attack success, which Appendix J measures directly.

N HARD-TOKEN REALIZATION

The wrapper is optimized in continuous embedding space, which upper-bounds what an attacker restricted to discrete tokens can reach, so how much of the effect survives in token space is the practical question. We report what our hard-token searches reach and where they stop.

Evasion is reachable in token space; joint success is not. Table 10 gives the endpoints, all on 3B: one shared 80-token suffix optimized over five prompts from a single jailbreak method, width 128, 500 visits, against an unoptimized baseline of 18.6%. The strongest run reaches 64.3% layer-averaged evasion, so the monitor boundary can be crossed without continuous embeddings; it preserves the response on none of its five examples, so joint success is zero. The same split holds throughout. Cross-entropy preservation alone holds the response on 80% of examples while evading 5.0%, and adding a KLD preservation term to the strongest attention run cuts evasion from 64.3% to 20.7% while recovering the response on one of its five examples. Each half is attainable on its own; the best joint success across these runs is 10.7%, on a KLD-preservation-only arm not listed in Table 10, against 52–77% for the continuous attack.

Table 10: Hard-token endpoints. Rows two and three are separate runs that coincide on both columns; they differ in final distractor mass (7.0% against 22.9%). ER is layer-averaged evasion over the 28 transformer blocks; Response is the judged preservation rate over the five examples, which bounds joint success from above.

Objective and distractor treatment	ER	Response
Log-mass attention, ten editable distractor positions	**64.3**	0
Log-mass attention, distractor held fixed and benign	20.7	20
Log mass + KLD preservation (weight 0.1)	20.7	20
CE preservation only	5.0	**80**

The evading run edits the distractor, and the distractor stops being benign. Holding the distractor fixed and benign, the same search reaches 20.7% rather than 64.3%, so the evasion depends on optimizing distractor positions and not on the appended span alone. Scored in isolation, without the suffix or the harmful prompt, the optimized distractor falls from 98.1% mean benign probability to 48.6% under normal chat rendering. What the discrete search finds is therefore not the mechanism the continuous attack realizes, in which an untouched benign distractor carries the attention.

Neither search budget nor displacement magnitude is the obstacle. Doubling the step count moved evasion from 22.9% to 23.1%, while doubling the suffix length moved it to 40.1%, so capacity binds and iteration does not. Comparing total injected embedding norm rather than per-token norm, the token steerer on 3B and 8B already exceeds parity with the continuous solution it is trying to match and still does not evade; what separates them is direction and placement, one unconstrained vector at a single position against many vocabulary-constrained vectors spread over many positions. Appendix E reaches the same conclusion from the readout side.

Language-constrained optimization is the next step we would take. The two findings point the same way: evasion needs the distractor positions to be optimized, and the threat model needs those positions to stay benign. An optimizer constrained to produce fluent, benign-scoring natural language at the distractor, rather than arbitrary vocabulary tokens, would test whether both can hold at once. We have not run it, and we do not claim that no token-space realization exists; what these searches establish is where the difficulty sits.

O Limitations and Future Work

The wrapper is optimized in continuous embedding space, which upper-bounds what an attacker restricted to discrete tokens can achieve. Appendix N measures that gap: token search crosses the monitor boundary but does not preserve the response, and closing it calls for an optimizer constrained to benign natural language at the distractor. Randomizing the readout position, or reading positions the attacker cannot anticipate, is a concrete defense worth testing. One-class OOD detection is not generally robust to benign distribution shift, limiting its use as a stand-alone defense. Even so, future work should explore making the attack OOD-stealthy by anchoring the attacked readout within the benign manifold rather than merely shifting it across the probe boundary; Appendix G explores one possible formulation.