Bait-and-Recover: Poisoning Internal Refusal Signals to Defend LLMs against White-Box Editing Jailbreaks

Tian Gao[1,2] **Zhipeng Xie**[2] **Yuhao Wu**[2] **Junhua Liu**[1,2] **Xin Fang**[1,3]*
[1]iFLYTEK CO., LTD., Hefei, China.
[2]Anhui Laboratory for Safe Artificial Intelligence in the Yangtze River Delta
[3]University of Science and Technology of China, Hefei, China.
{tiangao5,zpxie2,yhwu30,jhliu,xinfang}@iflytek.com
Code: github.com/SparkShieldLab/bait-and-recover

Abstract

Open-weight large language models face a low-cost white-box threat from representation engineering attacks. Attackers can estimate refusal directions and search for projection-matrix edits that suppress safety alignment while preserving general capabilities, within minutes on a single GPU and without gradient-based training. We propose **Bait-and-Recover**, a weight-level defense that places a bait adapter where attackers read activations and a paired recovery adapter at the subsequent layer. Trained via gradient routing, this decouples the observation path from the behavior path. By actively poisoning the residual signal used for measurement, Bait-and-Recover disrupts the attacker's edit search, while the recovery layer restores clean downstream computation. Across four open-weight models, our defense raises the minimum refusal rate against white-box edit searches from 16.25% to 71.75% under a strict behavior-preservation budget (KL $\leq$ 0.10), with negligible impact on general benchmarks. By invalidating the core measurement assumption of these attacks, observation-path poisoning offers a practical complement to behavior-level safety training.

1 Introduction

The public release of large language models (LLMs) (Yang et al., 2025; Gemma Team, 2025) provides direct transparency into safety alignment, but simultaneously exposes these models to adversarial weight editing (Meng et al., 2022, 2023). A recent family of white-box jailbreaks, exemplified by Heretic (Weidmann, 2025), treats safety-aligned refusal as a representation-engineering target (Zou et al., 2023a; Arditi et al., 2024): estimate the residual direction that separates harmful and harmless prompts, then search for weight edits that suppress that direction while preserving the model's general capabilities. Unlike prompt-level jailbreaks (Zou et al., 2023b; Ren et al., 2025), this white-box attack requires no crafted suffix: the attacker reads the model's internal computation, edits projection matrices in place, and keeps the best edit found within a strict behavior-preservation budget.

This targeted editing method is a low-cost, broadly applicable threat to open-weight releases. Unlike fine-tuning-based attacks, Heretic searches over projection edits without gradient-based training. Its search executes in minutes on a single GPU and yields an edited checkpoint that suppresses refusal while retaining normal behavior on benign prompts. Once redistributed, such checkpoints are difficult for the original defender to recall or patch. The practical defense point therefore shifts to the released weights themselves: protection must be embedded directly in the checkpoint before distribution. We aim to preserve the safety alignment of released checkpoints against low-cost weight editing, while retaining their original capabilities.

This attack also exposes a gap in existing safety mechanisms. Safety fine-tuning, preference optimization, runtime filters, and activation-level defenses (Ouyang et al., 2022; Rafailov et al., 2023; Zou et al., 2024) constrain the model's output behavior, but leave the internal refusal representations unprotected against measurement and editing. Refusal-direction editing relies on estimating a refusal-related direction and using it to construct effective weight edits. Decoupling these two components is therefore a natural defense target: actively poison the internal signal to blind the attacker's measurement, while preserving the clean downstream computation and outputs.

We formalize this idea as *observation–behavior path decoupling* and implement it via **Bait-and-Recover**. Let ℓ denote a selected bait layer, and index residual streams so that layer ℓ writes to $r_{\ell+1}$. Our defense places a bait adapter inside layer ℓ, making the poisoned signal visible at the attacker's readout $r_{\ell+1}$, and a paired recovery adapter inside

*Corresponding author: xinfang@iflytek.com.

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 1608–1621
November 6-10, 2026 ©2026 Association for Computational Linguistics

the following layer $\ell+1$, repairing the downstream residual $r_{\ell+2}$. The intended state is deliberately asymmetric: the attacker's search uses a corrupted readout, while generation continues from a near-clean downstream residual.

Contributions. By disrupting refusal-direction estimation in Heretic-style editing, our work offers a weight-level complement to behavior-level safety training. Concretely:

1. We formulate *observation–behavior path decoupling* as an explicit defense objective for refusal-direction editing, shifting the focus from constraining outputs to poisoning the attacker's internal measurement step.

2. We propose a cross-layer bait/recovery design with a gradient-routing rule that successfully decouples the two paths during training, preventing recovery pressure from erasing the bait.

3. Across four open-weight models, the defense raises the mean minimum refusal rate against Heretic searches spanning both global and per-layer direction scopes under a strict behavior-preservation budget (KL $\leq$ 0.10) from 16.25% to 71.75%, with limited impact on the evaluated general benchmarks.

2 Related Work

Representation-engineering jailbreaks. Heretic (Weidmann, 2025) is the direct attack we evaluate against. Although released as an open-source tool rather than a peer-reviewed paper, it is a representative practical implementation of automatic abliteration: the public repository reports broad uptake,[1] and its technical design builds on the observation that refusal behavior can often be captured by a low-dimensional direction in residual space (Arditi et al., 2024). Heretic packages this line of work into an automated attack by combining directional ablation with parameter search, drawing on prior abliteration systems and refinements (Labonne, 2024c; Lai, 2025b,a). More broadly, representation engineering studies how steering vectors and internal activations can monitor or control model behavior (Zou et al., 2023a). Recent empirical work characterizes how abliteration erodes diverse safety-pretraining components across open-weight checkpoints (Agnihotri et al., 2025), motivating defenses that specifically resist this attack family. We share the white-box access assumption of this line of work but invert its defensive target: rather than improving post-edit behavior, we attack the reliability of the attacker's representation estimate itself.

Activation-space defenses. Circuit-breaker-style defenses modify internal activations to improve alignment and robustness against adversarial inputs (Zou et al., 2024). Representation Noising (Rosati et al., 2024) is closer to our setting in that it also intervenes on internal representations under an open-weight threat model, but its objective is to remove or destabilize the harmful-representation information itself, so that harmful behavior cannot be recovered through subsequent fine-tuning. Bait-and-Recover targets a different stage of the attack: we assume a fine-tuning-free white-box edit search that estimates a refusal direction from model activations and then edits projection matrices in place. Rather than removing the underlying refusal or harmful representations, we leave them intact and inject a localized poison signal at the attacker's observation layer, paired with a recovery adapter at the next layer that restores the downstream computation. This asymmetry preserves the model's normal safety behavior while making the attacker's inferred edit direction unreliable. These defenses therefore cover different parts of the safety problem: behavior under adversarial inputs, resistance to harmful fine-tuning, and the reliability of the attacker's internal measurement.

Model editing and LoRA defenses. ROME, MEMIT, and related editing methods show that localized weight changes can rewrite model behavior (Meng et al., 2022, 2023), and recent analyses show that pruning and low-rank modifications can directly compromise safety alignment (Wei et al., 2024). Safety-aware fine-tuning and robust safeguards investigate how to preserve or strengthen safety behavior under subsequent modification (Hsu et al., 2024; Tamirisa et al., 2025). Our work is complementary: it targets the estimator that a white-box edit attack relies on, not only the post-edit behavior.

[1] The Heretic repository listed over 21k stars and 2.2k forks, and its README reported over 3000 community-created Heretic models, when accessed on May 23, 2026: https://github.com/p-e-w/heretic.

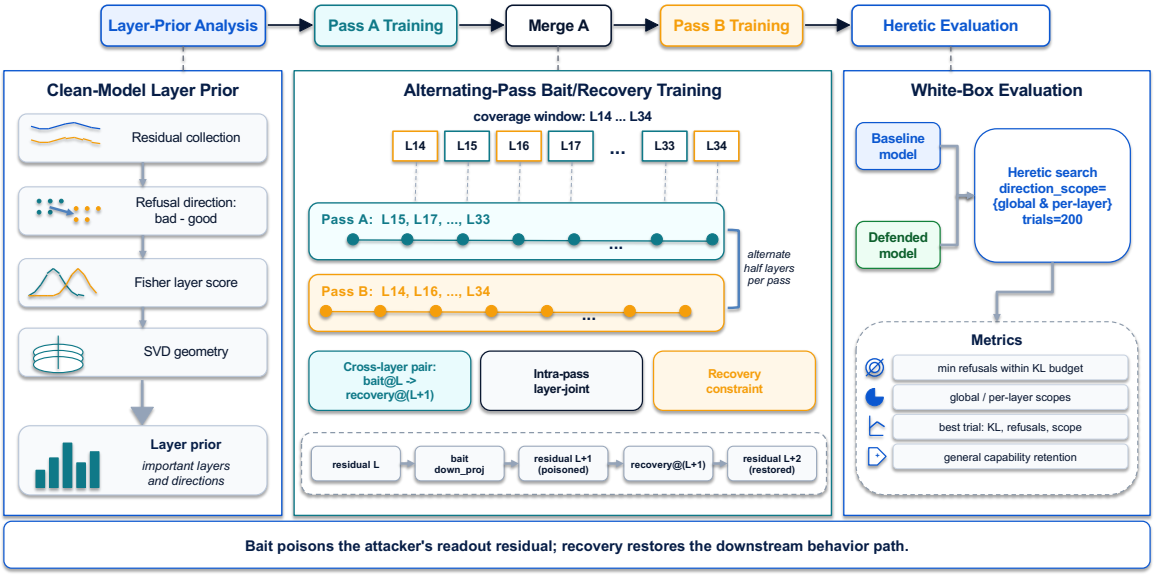


Figure 1: Overview of the alternating-pass Bait-and-Recover pipeline. A clean-model layer-prior analysis determines the bait band before training; Training Pass A covers one layer parity, Training Pass B starts from the merged Pass A checkpoint and covers the complementary parity, and the final checkpoint is evaluated against the same white-box Heretic search used for the baseline. The layer-joint box denotes local tuning inside a pass over neighboring trained pairs, not a joint optimization between Pass A and Pass B.

3 Background and Problem Formulation

Heretic-style attack. Given harmful prompts $\mathcal{P}_{\text{harmful}}$ and harmless prompts $\mathcal{P}_{\text{harmless}}$, the attacker reads residual stream $\ell{+}1$ and estimates the refusal direction by the difference of means,

$$\hat{\boldsymbol{r}}_{\ell+1} = \mathbb{E}_{x\sim\mathcal{P}_{\text{harmful}}}[r_{\ell+1}(x)] - \mathbb{E}_{x\sim\mathcal{P}_{\text{harmless}}}[r_{\ell+1}(x)]. \quad (1)$$

Heretic (Weidmann, 2025) then performs an Optuna (Akiba et al., 2019) search over target layers, direction scope (global or per-layer), edited modules, sign, and edit scale, optimizing for the lowest harmful-prompt refusal count under a KL budget. We evaluate the default configuration: 200 trials, with `o_proj` and `down_proj` as the primary editable modules, and report the minimum refusal count among trials inside each budget.

Problem formulation. We assume that the defense mechanism is public and the attacker has full access to the released weights and intermediate activations. Heretic re-estimates directions on each checkpoint and searches both global and per-layer edits; Appendix E.1 also tests three alternative estimators. The search adapts to the model but has no procedure specifically designed to remove bait and recovery updates. The defense is evaluated as a single merged checkpoint. Let $\mathcal{M}$ be the clean model, $\widetilde{\mathcal{M}}$ the defended model, and $\hat{r}(\cdot)$ the attacker's estimator (Eq. 1). The post-edit attack success rate ASR is the fraction of harmful prompts for which the edited model does not refuse, equivalently ASR $= 1 -$ refusal rate. The defender minimizes ASR subject to a general-prompt KL budget:

$$\min_{\theta_{\text{bait}},\theta_{\text{rec}}} \quad \text{ASR}\Big(\text{Edit}(\widetilde{\mathcal{M}}, \hat{r}(\widetilde{\mathcal{M}}))\Big) \quad (2)$$
$$\text{s.t.} \quad \text{KL}(\widetilde{\mathcal{M}}\|\mathcal{M}) \le \varepsilon_{\text{KL}}.$$

4 Method

Bait-and-Recover has four stages: (i) build a layer prior from harmful and harmless prompts on the clean model; (ii) train two complementary parities of bait/recovery pairs across the selected band; (iii) within each pass, perform a short local layer-joint step over neighboring trained pairs to absorb cross-layer coupling; and (iv) evaluate the merged checkpoint with the same Heretic search used for the baseline. Figure 1 summarizes this full pipeline.

4.1 Layer-band Selection

We select the defended layer band using clean-model statistics. Let L be the number of transformer layers and let ℓ denote a bait layer. Because

layer ℓ writes into $r_{\ell+1}$ and the paired recovery layer $\ell+1$ writes into $r_{\ell+2}$, the last admissible bait layer is $\ell \le L-2$.

We first define the default clean-model prior window

$$\mathcal{W}_{\text{prior}} = [\lfloor 0.4(L-1) \rfloor,\ L-2], \tag{3}$$

which targets middle-to-deep layers while stopping at $L-2$, so every bait layer still has a following recovery layer.

The layer prior uses the separability between harmful and harmless prompts in the clean-model residual stream. For a candidate bait layer ℓ and class $c \in \{\text{harmful}, \text{harmless}\}$, let $R_{c,\ell+1} = \{r_{\ell+1}(x) : x \in \mathcal{P}_c\}$ be the final-token residuals used by the Heretic direction estimator. We estimate

$$\begin{aligned} \mu_{c,\ell+1} &= \frac{1}{|R_{c,\ell+1}|} \sum_{r \in R_{c,\ell+1}} r, \\ \Sigma_{c,\ell+1} &= \text{Cov}(R_{c,\ell+1}), \end{aligned} \tag{4}$$

and score the harmful/harmless separability at that residual by a trace-form Fisher ratio,[2]

$$F_{\ell+1} = \frac{\|\mu_{\text{harmful},\ell+1} - \mu_{\text{harmless},\ell+1}\|_2^2}{\text{tr}(\Sigma_{\text{harmful},\ell+1}) + \text{tr}(\Sigma_{\text{harmless},\ell+1}) + \epsilon}, \tag{5}$$

where larger $F_{\ell+1}$ means the clean refusal signal is easier for Heretic to measure. Let N_{pair} denote the number of available bait/recovery pairs. Since each pair covers one bait layer, if the prior window contains at most N_{pair} layers, we use the whole window. If it is larger, we choose the contiguous sub-band with the highest cumulative Fisher score:

$$S^\star = \underset{S=[a,a+N_{\text{pair}}-1] \subseteq \mathcal{W}_{\text{prior}}}{\arg\max} \sum_{\ell \in S} F_{\ell+1}. \tag{6}$$

4.2 Cross-layer Defense Unit

Each defense unit spans two adjacent transformer layers. To intercept the attacker's search space, we place a bait adapter on the `down_proj` module in layer ℓ. The paired recovery uses adapters on the `o_proj` and `down_proj` modules in layer $\ell+1$, repairing the computation before it reaches the generation head. A gradient-routing rule decouples the updates within each unit: recovery gradients do not update bait parameters, and visibility gradients do not update recovery parameters.

[2]We use a trace-form simplification rather than the full Fisher discriminant $\text{tr}(\Sigma_W^{-1}\Sigma_B)$ to avoid ill-conditioned covariance inversion at high dimensions.

4.3 Bait Adapter

The bait adapter is a low-rank (Hu et al., 2022) residual perturbation attached to `down_proj`:

$$\begin{aligned} \delta_\ell(h) &= \exp(\gamma_\ell) B_\ell A_\ell h, \\ A_\ell &\in \mathbb{R}^{r \times d_{\text{in}}}, \quad B_\ell \in \mathbb{R}^{d_{\text{model}} \times r}. \end{aligned} \tag{7}$$

Here h is the input to layer ℓ's `down_proj`, and γ_ℓ is a trainable gate initialized small and later calibrated to the target bait visibility.

The initialization reuses the clean-model means from Section 4.1. For a bait layer ℓ, the residual-space refusal direction is

$$\hat{d}_{\ell+1} = \frac{\mu_{\text{harmful},\ell+1} - \mu_{\text{harmless},\ell+1}}{\|\mu_{\text{harmful},\ell+1} - \mu_{\text{harmless},\ell+1}\|_2}. \tag{8}$$

We also compute an input detector direction at the `down_proj` input. Let $\bar{h}_{c,\ell}$ be the class mean of the final-token `down_proj` inputs $h_\ell(x)$ for $x \in \mathcal{P}_c$; then

$$\hat{q}_\ell = \frac{\bar{h}_{\text{harmful},\ell} - \bar{h}_{\text{harmless},\ell}}{\|\bar{h}_{\text{harmful},\ell} - \bar{h}_{\text{harmless},\ell}\|_2}. \tag{9}$$

Finally, we decompose the down-projection weights,

$$W_\ell^{\text{down}} = U_\ell \,\text{diag}(\sigma_\ell) V_\ell^\top. \tag{10}$$

The right singular vectors V_ℓ provide additional input-side detector directions, while the left singular vectors U_ℓ provide residual-space output directions.

The first bait channel is the refusal-cancel channel. Its detector reads the input-side refusal direction $\hat{q}_\ell$, and its output writes against the residual-stream refusal direction $\hat{d}_{\ell+1}$:

$$A_\ell[0,:] = \hat{q}_\ell^\top, \qquad B_\ell[:,0] = -\hat{d}_{\ell+1}. \tag{11}$$

A second *tag* channel carries a side signal for the recovery layer while avoiding the clean refusal axis. Let $v_{\ell,j}$ be the j-th right singular vector in V_ℓ. We choose the vector whose projection away from $\hat{q}_\ell$ has the largest norm,

$$j^\star = \arg\max_j \left\| (I - \hat{q}_\ell \hat{q}_\ell^\top) v_{\ell,j} \right\|_2. \tag{12}$$

The tag output $\hat{t}_{\ell+1}$ is initialized to a reproducible pseudo-random unit vector in the orthogonal complement of $\hat{d}_{\ell+1}$. We construct the tag's detector $\tilde{v}_\ell$ by orthogonalizing and normalizing the selected singular vector:

$$\tilde{v}_\ell = \frac{(I - \hat{q}_\ell \hat{q}_\ell^\top) v_{\ell,j^\star}}{\left\| (I - \hat{q}_\ell \hat{q}_\ell^\top) v_{\ell,j^\star} \right\|_2}, \tag{13}$$

and initialize the channel as $A_\ell[1,:] = \tilde{v}_\ell^\top$ and $B_\ell[:,1] = \hat{t}_{\ell+1}$. Thus the tag detector is separated from $\hat{q}_\ell$, and the tag output is orthogonal to $\hat{d}_{\ell+1}$. If the adapter rank exceeds two, each additional channel is initialized from the leading unused SVD directions: its detector uses a right singular vector from V_ℓ, and its output uses the corresponding left singular vector from U_ℓ, ordered by σ_ℓ. To ensure training stability, we keep A_ℓ fixed and update only B_ℓ and γ_ℓ during training. This keeps the bait's input-side detectors anchored to the clean refusal boundary, while its residual-space outputs adapt.

4.4 Recovery Adapter

After the bait has changed $r_{\ell+1}$, the next layer can propagate this perturbation through both residual-writing paths: the attention output projection and the MLP down-projection. For each bait at layer ℓ, we therefore attach recovery adapters to `o_proj` and `down_proj` of layer $\ell{+}1$. For a recovery site $m \in \{\texttt{o_proj}, \texttt{down_proj}\}$ with input z_m, the adapter is

$$\rho_{\ell,m}(z_m) = O_{\ell,m} D_{\ell,m} z_m. \tag{14}$$

Here $D_{\ell,m} \in \mathbb{R}^{r \times d_{\text{in},m}}$ detects recovery-layer features, and $O_{\ell,m} \in \mathbb{R}^{d_{\text{model}} \times r}$ writes a corrective update into the residual stream.

For initialization, we compute the SVD of the recovery-site base projection,

$$W_{\ell+1,m} = U^{\text{rec}}_{\ell,m} \operatorname{diag}(\sigma^{\text{rec}}_{\ell,m}) (V^{\text{rec}}_{\ell,m})^\top. \tag{15}$$

Let $v^{\text{rec}}_{\ell,m,i}$ be the i-th right singular vector, ordered by decreasing $\sigma^{\text{rec}}_{\ell,m}$. The detector reads these high-energy input directions, while the output is initialized to the anti-bait output basis:

$$D_{\ell,m}[i,:] = (v^{\text{rec}}_{\ell,m,i})^\top, \tag{16}$$

$$O_{\ell,m}[:,i] = -B_\ell[:,i], \quad i = 0, \ldots, r-1. \tag{17}$$

Training then refines these anti-bait directions to account for the intermediate layer transformations. Recovery is a learned weight update, not an explicit inverse of the bait. After merging, the checkpoint stores neither separate adapters nor clean reference residuals. Within the defended band, a layer that recovers one bait can also contain the next bait.

4.5 Training Objective and Schedule

Training combines two main objectives with a lightweight visibility floor:

$$\mathcal{L} = \lambda_{\text{fisher}} \mathcal{L}_{\text{fisher}} + \lambda_{\text{rec}} \mathcal{L}_{\text{rec}} + \lambda_{\text{vis}} \mathcal{L}_{\text{vis}}. \tag{18}$$

$\mathcal{L}_{\text{fisher}}$ is the trace-form Fisher ratio from Section 4.1, recomputed on defended bait residuals:

$$\mathcal{L}_{\text{fisher}} = \frac{1}{|S|} \sum_{\ell \in S} F^{\text{def}}_{\ell+1}. \tag{19}$$

This attack-facing term directly lowers the harmful/harmless separability visible to the attacker's direction estimator. $\mathcal{L}_{\text{rec}}$ aligns the clean and defended residual states after each paired recovery layer,

$$\mathcal{L}_{\text{rec}} = \frac{1}{|S|} \sum_{\ell \in S} \mathbb{E}_x \left[\frac{\| r^{\text{def}}_{\ell+2}(x) - r^{\text{clean}}_{\ell+2}(x) \|_2^2}{\| r^{\text{clean}}_{\ell+2}(x) \|_2^2 + \epsilon} \right]. \tag{20}$$

Finally, $\mathcal{L}_{\text{vis}}$ is an auxiliary hinge that keeps the bait measurable at the attacker-visible residual:

$$\mathcal{L}_{\text{vis}} = \frac{1}{|S|} \sum_{\ell \in S} \mathbb{E}_x \left[\tau_{\text{vis}} - \frac{\| \Delta r_{\ell+1}(x) \|_2}{\| r^{\text{clean}}_{\ell+1}(x) \|_2 + \epsilon} \right]_+^2, \tag{21}$$

where $\Delta r_{\ell+1}(x) = r^{\text{def}}_{\ell+1}(x) - r^{\text{clean}}_{\ell+1}(x)$ and $[u]_+ = \max(u, 0)$.

Gradients are routed by role: recovery alignment updates only recovery adapters, while Fisher-readout and visibility losses update only bait adapters. For multi-layer coverage, the selected band is split into two alternating parities trained in Pass A and Pass B. KL is monitored on general prompts during training, with a final KL budget.

5 Experimental Setup

Data. Bait-and-Recover uses defense-training data that is separate from Heretic's attack and evaluation data. For the defense side, harmless prompts and unlabeled general prompts come from Alpaca-GPT4 instruction prompts (Peng et al., 2023), while harmful prompts come from SafeMTData `Attack_600` (Ren et al., 2025). We extract the harmful prompt field and retain direct harmful requests using a deterministic filter over procedural markers (e.g., how-to, steps, build) and explicit abuse terms (e.g., malware, phishing, weapon, attack). Because this filtered harmful pool is smaller than the broad instruction corpus used for harmless prompts, the labeled defense split uses 128 harmless and 64 harmful prompts for refusal direction estimation and the Fisher-readout loss. We also reserve 256 unlabeled general prompts for residual recovery, the label-free visibility floor, and KL monitoring, plus a non-overlapping validation split of 64 harmless and 32 harmful prompts.

Table 1: Evaluated models and defense configurations. The bait band is chosen by the clean-model layer-prior rule; Pass A/B list the alternating bait-layer ranges used by the two-stage schedule, and r is the realized LoRA rank range after applying the rank-selection rule.

Model	L	Band	Pass A	Pass B	r
Qwen3-4B-Instruct	36	14–34	odd 15–33	even 14–34	4–16
Qwen3-8B	36	14–34	odd 15–33	even 14–34	7–16
Gemma-3-1B-IT	26	10–24	odd 11–23	even 10–24	4
Gemma-3-12B-IT	48	18–38	odd 19–37	even 18–38	4–16

For attack evaluation, we leave Heretic's prompt sources unchanged: `mlabonne/harmless_alpaca` and `mlabonne/harmful_behaviors` (Labonne, 2024b,a). Heretic uses the first 400 training examples from each source to estimate the attack direction, and evaluates each attack trial on 100 held-out harmless prompts and 100 held-out harmful prompts for attack KL and harmful-refusal measurement. All Heretic searches use 200 trials.

Models and configuration. We evaluate four open-weight instruction models from the Qwen3 and Gemma 3 families (Yang et al., 2025; Gemma Team, 2025). Table 1 reports the layer count, defended bait band, alternating pass split, and LoRA rank range for each model. Layer bands follow the rule in Section 4.1. We fix the pair budget at $N_{\text{pair}} = 21$ across all models: this value is inherited from our initial Qwen3-4B-Instruct experiment, where the default prior window $[\lfloor 0.4(L-1) \rfloor, L-2]$ at $L = 36$ contains exactly 21 admissible bait layers and was used in full. We then reuse this same 21-pair budget for all subsequent models so that defense capacity stays comparable rather than scaling with model depth. Under this fixed budget, both Qwen models use their default 21-layer window in full; Gemma-3-1B-IT has only 15 admissible bait layers in its default window, so we use all of them; and for Gemma-3-12B-IT the default window $[18, 46]$ is larger than N_{pair}, so the Fisher-based contiguous-band selection in Section 4.1 picks the highest-scoring 21-layer sub-band, $[18, 38]$ (Appendix A). LoRA ranks are selected per layer by first choosing the smallest rank whose clean-model cumulative squared singular-value energy reaches 40%, then clipping to the allowed range and enforcing the minimum rank required by the refusal-cancel and tag-channel basis. The recovery adapter uses the same rank as its paired bait adapter. Table 1 reports the realized ranks; the Gemma-3-1B layers all happen to hit the minimum rank, so its realized range collapses to the single value 4.

Training hyperparameters. The four main-model experiments use AdamW (Loshchilov and Hutter, 2019) with weight decay 0, gradient clipping at 1.0, and learning rate 2×10^{-4}. We fix λ_{rec}=0.5, λ_{vis}=4, λ_{fisher}=2, τ_{vis}=0.20. In each pass, every bait/recovery pair is first trained in isolation for two epochs; after these single-pair stages, neighboring two-pair blocks receive one epoch of intra-pass layer-joint tuning. Detailed reproducibility configurations are provided in Appendix D.

Evaluation metrics. The main attack metric is the minimum refusal rate (out of 100 harmful prompts) found by a 200-trial Heretic search at each reported evaluation KL threshold; higher values mean stronger defense. Attack KL compares first-token output distributions on harmless prompts against the corresponding pre-edit checkpoint, whether clean or defended. Separately, we track held-out general-prompt KL between the defended and clean models after each defense pass as a behavior-preservation diagnostic. To measure changes on general capability benchmarks, we run paired EvalScope evaluations (ModelScope Team, 2024) with the same vLLM backend (Kwon et al., 2023), chat template, output cap, and postprocessing for each clean/defended pair, reporting MMLU-Redux accuracy (Gema et al., 2025; Hendrycks et al., 2021) and IFEval prompt-level strict accuracy (Zhou et al., 2023; Biderman et al., 2024).

6 Results

6.1 Fisher Suppression and Behavior Preservation

Figure 2 tracks the effect of training. Bait-layer Fisher scores decrease in both passes, while recovery-layer Fisher scores remain closer to their pre-pass values. Appendix C reports the corresponding raw Fisher values.

Alongside this Fisher diagnostic, we monitor validation KL on general prompts to check that the defense has not moved normal behavior too far. The Pass B general-prompt KL remains below 0.20 for every model, though the amount of movement differs by architecture (Table 2). Qwen3-8B is the most conservative case, while Gemma-3-12B has the largest general-prompt movement among the four models.

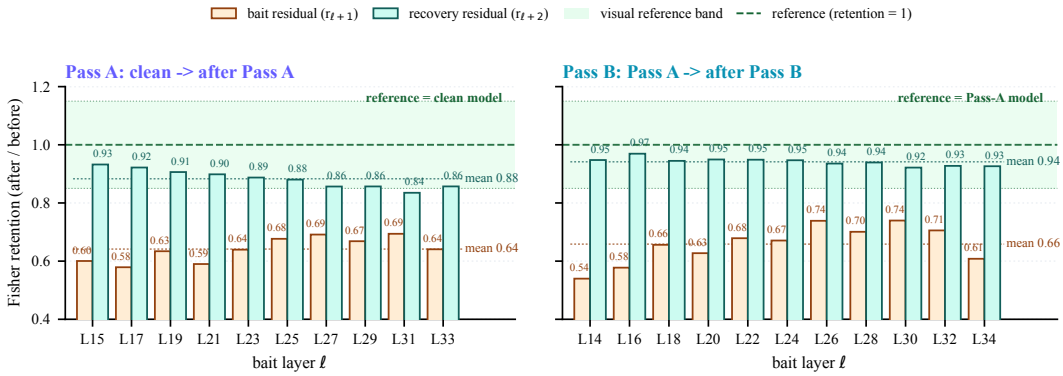


Figure 2: Layer-wise Fisher retention during Qwen3-4B-Instruct training. Retention is after-pass Fisher divided by the before-pass reference (clean model for Pass A; Pass A model for Pass B). The shaded band is a visual reference around unchanged retention.

Table 2: Behavior-preservation diagnostics and general capability retention after defense. KL is measured on held-out general prompts after Pass A and Pass B; Pass B is the final defended checkpoint. MMLU reports MMLU-Redux accuracy, IFEval reports prompt-level strict accuracy, and Δ is defended minus baseline.

	General-prompt KL		MMLU-Redux			IFEval		
Model	Pass A	Pass B	Base	Def.	Δ	Base	Def.	Δ
Qwen3-4B-Instruct	0.016	0.020	0.7549	0.7538	-0.0011	0.8244	0.8373	+0.0129
Qwen3-8B	0.007	0.010	0.7782	0.7761	-0.0021	0.8373	0.8373	+0.0000
Gemma-3-1B-IT	0.066	0.079	0.4180	0.4193	+0.0013	0.5730	0.5342	-0.0388
Gemma-3-12B-IT	0.049	0.085	0.7528	0.7506	-0.0022	0.8059	0.8096	+0.0037
Mean	0.035	0.049	0.6760	0.6750	-0.0010	0.7602	0.7546	-0.0055

6.2 General Capability Retention

Because KL alone does not establish general capability preservation, we also run paired benchmark evaluations for each clean/defended checkpoint. Table 2 reports both the validation KL diagnostics and the paired capability evaluation. MMLU-Redux accuracy is largely unchanged across the four checkpoints: the defended-minus-baseline deltas range from -0.0022 to $+0.0013$, with a mean change of -0.0010. IFEval prompt-strict accuracy is also stable in aggregate, with a mean change of -0.0055, although Gemma-3-1B-IT drops 3.88 percentage points on this metric.

Additional Gemma-3-12B evaluations cover GSM8K, a 200-prompt Arena-Hard subset, and the 250 benign XSTest prompts (Appendix E.3). Clean/defended scores are 93.93/93.63% on GSM8K and 84.42/84.58% on Arena-Hard. XSTest helpful non-refusal rises from 94.80% to 98.40%. These results extend the utility evaluation to mathematical reasoning, open-ended responses, and benign requests that can trigger over-refusal.

6.3 Heretic Attack Evaluation

We report the minimum refusal rate over the pooled global and per-layer direction scopes at each KL threshold; higher rates mean that more harmful requests are refused.

We use $\mathrm{KL} \leq 0.10$ as the primary evaluation budget because it matches the behavior-preservation scale of the defended checkpoints themselves: after Pass B, the held-out general-prompt KL values in Table 2 are all below 0.10, and the paired MMLU-Redux/IFEval evaluations show little aggregate capability movement. Attack KL measures output-distribution divergence rather than task accuracy.

At this primary evaluation budget, Bait-and-Recover raises the mean minimum refusal rate from 16.25% to 71.75% (+55.50 percentage points; 4.4×), with gains on all four models (Table 3; see Appendix B for full results across all KL budgets). The defense remains effective at the looser $\mathrm{KL} \leq 0.20$ budget, where the mean minimum refusal rate of the defended models is 65.25%, 6.5 percentage points below the $\mathrm{KL} \leq 0.10$ value and 55.25 points above the baseline mean of 10.00%. At $\mathrm{KL} \leq 1.0$, this rate falls to 54.00% (+50.25 per-

Table 3: Main Heretic attack result across four KL budgets. Each cell reports the minimum refusal rate (%, out of 100 harmful prompts; higher is better); bold marks the $\mathrm{KL} \leq 0.10$ primary evaluation budget.

	$\mathrm{KL} \leq 0.05$			$\mathrm{KL} \leq 0.10$			$\mathrm{KL} \leq 0.20$			$\mathrm{KL} \leq 1.0$		
Model	Base	Def.	Δ	Base	Def.	Δ	Base	Def.	Δ	Base	Def.	Δ
Qwen3-4B-Instruct	97	78	−19	**46**	**78**	+32	24	65	+41	5	60	+55
Qwen3-8B	36	67	+31	**11**	**60**	+49	11	50	+39	8	33	+25
Gemma-3-1B-IT	17	84	+67	**5**	**67**	+62	3	64	+61	2	64	+62
Gemma-3-12B-IT	12	82	+70	**3**	**82**	+79	2	82	+80	0	59	+59
Mean	40.50	77.75	+37.25	**16.25**	**71.75**	+55.50	10.00	65.25	+55.25	3.75	54.00	+50.25

centage points over baseline; Table 3, $\mathrm{KL} \leq 1.0$ column), with Gemma-3-12B-IT keeping its per-layer defended score at 82 across all KL budgets, while its global defended score falls from 83 to 59 only at $\mathrm{KL} \leq 1.0$. This is consistent with the broader trend at the primary $\mathrm{KL} \leq 0.10$ budget, where global search is the most effective attack scope against three of the four defended models.

Additional estimators and defense comparison. On Gemma-3-12B, PCA top-8 and geometric-median estimators reduce clean-model refusal to 4% and 5% at $\mathrm{KL} \leq 0.10$, while defended refusal remains at 79% and 81%. We also compare clean Llama-3-8B, its public RR checkpoint, and Bait-and-Recover under the same 200-trial Heretic protocol: minimum refusal is 4%, 2%, and 90%, respectively. Appendix E reports the settings, pre-edit refusal rates, and results at larger budgets.

Global vs. per-layer scope. The scope decomposition shows where the defended models remain most vulnerable. Before defense, the most effective attack scope differs by model: global search sets the pooled minimum for Qwen3-4B-Instruct, Qwen3-8B, and Gemma-3-1B-IT, while per-layer search is the most effective attack on Gemma-3-12B-IT. After defense, global search remains the most effective attack scope against the three smaller models, while for Gemma-3-12B-IT the global and per-layer defended scores are nearly tied at $\mathrm{KL} \leq 0.10$ (83 vs. 82). Across models (Table 4), defended global scores are 78/60/67/83 and per-layer scores are 81/88/82/82. The mean across models reflects this trend, with a 72.00% refusal rate under global search compared to 83.25% under per-layer search at $\mathrm{KL} \leq 0.10$.

Table 4: Heretic minimum refusals at the $\mathrm{KL} \leq 0.10$ primary evaluation budget, split by attacker direction scope. Each entry reports base/defended/Δ.

Model	Global	Per-layer
Qwen3-4B-Instruct	46/**78**/+32	94/81/−13
Qwen3-8B	11/**60**/+49	99/88/−11
Gemma-3-1B-IT	5/**67**/+62	22/82/+60
Gemma-3-12B-IT	26/**83**/+57	3/82/+79
Mean	22.00/**72.00**/+50.00	54.50/83.25/+28.75

7 Conclusion

We introduced Bait-and-Recover, a weight-level defense that protects open-weight models against Heretic-style refusal-direction editing by disrupting direction estimation. Rather than only hardening the output behavior, Bait-and-Recover places a bait adapter at the attacker's observation layer and a paired recovery adapter at the next layer. Trained under a gradient-routing rule, this mechanism decouples the observation path from the behavior path: it actively poisons the residual signal used to estimate the refusal direction, making the attacker's edit search unreliable, while the recovery layer restores the clean downstream computation. Across four open-weight models from the Qwen3 and Gemma-3 families, the defense raises the mean minimum refusal rate against Heretic searches from 16.25% to 71.75% at a strict $\mathrm{KL} \leq 0.10$ budget, with limited impact on the evaluated general benchmarks. Because the defense ships as a single merged checkpoint loadable with standard causal-LM inference, observation-path poisoning is a practical weight-level complement to behavior-level safety training. The tested attacks are generally more effective with global direction estimates. Future work includes defense-specific removal attacks and controlled component ablations.

Limitations

Our experiments focus on refusal-direction editing under low-distortion constraints. The defense is most effective at strict KL budgets ($\mathrm{KL} \leq 0.10$); allowing larger behavioral changes makes bypasses

easier to find. The tested adaptive estimators cover only part of the possible attack space. Attacks designed to identify and remove the defense updates, as well as adversarial fine-tuning, remain untested. Fine-tuning can weaken safety alignment even with a small amount of harmful training data (Qi et al., 2024).

The intended release setting provides the defended checkpoint without its unprotected counterpart. Although clean checkpoints serve as experimental baselines, the evaluated attacks do not use paired weight differencing. Access to both versions could help an attacker locate the modifications.

The experiments evaluate the complete defense rather than isolating each component through retraining ablations. Activation diagnostics therefore cannot establish which components are necessary. The direct defense comparison covers one public RR checkpoint under Heretic editing, while the additional estimator and utility evaluations cover Gemma-3-12B only. Broader comparisons and multi-seed evaluations would help assess the consistency of these findings.

Ethics Statement

This work studies a dual-use white-box jailbreak setting to improve defenses. To mitigate risks, we report only aggregate statistics, omit harmful generations, and explicitly avoid releasing jailbroken checkpoints. Because our defense modifies model weights, practical deployment should be paired with capability evaluations to verify that benign utility is preserved.

Acknowledgments

Supported by New Generation Artificial Intelligence-National Science and Technology Major Project (No. 2025ZD0123204).

References

Shashank Agnihotri, Jonas Jakubassa, Priyam Dey, Sachin Goyal, Bernt Schiele, Venkatesh Babu Radhakrishnan, and Margret Keuper. 2025. A granular study of safety pretraining under model abliteration. *arXiv preprint arXiv:2510.02768*. Accepted at NeurIPS 2025 Workshop on Lock-LLM.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2623–2631. ACM.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*, volume 37.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, and 11 others. 2024. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile Van Krieken, and Pasquale Minervini. 2025. Are we done with MMLU? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5069–5096, Albuquerque, New Mexico. Association for Computational Linguistics.

Gemma Team. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. 2024. Safe LoRA: The silver lining of reducing safety risks when finetuning large language models. In *Advances in Neural Information Processing Systems*, volume 37.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626. Association for Computing Machinery.

Maxime Labonne. 2024a. harmful_behaviors [dataset]. Hugging Face dataset, https://huggingface.co/datasets/mlabonne/harmful_behaviors. Accessed: 2026-05-23.

Maxime Labonne. 2024b. harmless_alpaca [dataset]. Hugging Face dataset, https://huggingface.co/datasets/mlabonne/harmless_alpaca. Accessed: 2026-05-23.

Maxime Labonne. 2024c. Uncensor any LLM with abliteration [blog post]. Hugging Face blog, https://huggingface.co/blog/mlabonne/abliteration. Accessed: 2026-05-23.

Jim Lai. 2025a. Norm-preserving biprojected abliteration [blog post]. Hugging Face community article, https://huggingface.co/blog/grimjim/norm-preserving-biprojected-abliteration. Accessed: 2026-05-23.

Jim Lai. 2025b. Projected abliteration [blog post]. Hugging Face community article, https://huggingface.co/blog/grimjim/projected-abliteration. Accessed: 2026-05-23.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024. From crowdsourced data to high-quality benchmarks: Arena-Hard and BenchBuilder pipeline. *arXiv preprint arXiv:2406.11939*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *International Conference on Learning Representations*.

ModelScope Team. 2024. EvalScope: Evaluation framework for large models. GitHub repository, https://github.com/modelscope/evalscope. Accessed: 2026-05-23.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with GPT-4. *arXiv preprint arXiv:2304.03277*.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2024. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36.

Qibing Ren, Hao Li, Dongrui Liu, Zhanxu Xie, Xiaoya Lu, Yu Qiao, Lei Sha, Junchi Yan, Lizhuang Ma, and Jing Shao. 2025. LLMs know their vulnerabilities: Uncover safety gaps through natural distribution shifts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24763–24785, Vienna, Austria. Association for Computational Linguistics.

Domenic Rosati, Jan Wehner, Kai Williams, Łukasz Bartoszcze, David Atanasov, Robie Gonzales, Subhabrata Majumdar, Carsten Maple, Hassan Sajjad, and Frank Rudzicz. 2024. Representation noising: A defence mechanism against harmful finetuning. In *Advances in Neural Information Processing Systems*, volume 37.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2024. 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)*, pages 5377–5400, Mexico City, Mexico. Association for Computational Linguistics.

Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, Andy Zou, Dawn Song, Bo Li, Dan Hendrycks, and Mantas Mazeika. 2025. Tamper-resistant safeguards for open-weight LLMs. In *International Conference on Learning Representations*.

Boyi Wei, Kaixuan Huang, Yangsibo Huang, Tinghao Xie, Xiangyu Qi, Mengzhou Xia, Prateek Mittal, Mengdi Wang, and Peter Henderson. 2024. Assessing the brittleness of safety alignment via pruning and low-rank modifications. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 52588–52610. PMLR.

Philipp Emanuel Weidmann. 2025. Heretic: Fully automatic censorship removal for language models [software]. GitHub repository, https://github.com/p-e-w/heretic. Accessed: 2026-05-23.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, and 2 others. 2023a. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, J. Zico Kolter, Matt Fredrikson, and Dan Hendrycks. 2024. Improving alignment and robustness with circuit breakers. In *Advances in Neural Information Processing Systems*, volume 37.

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

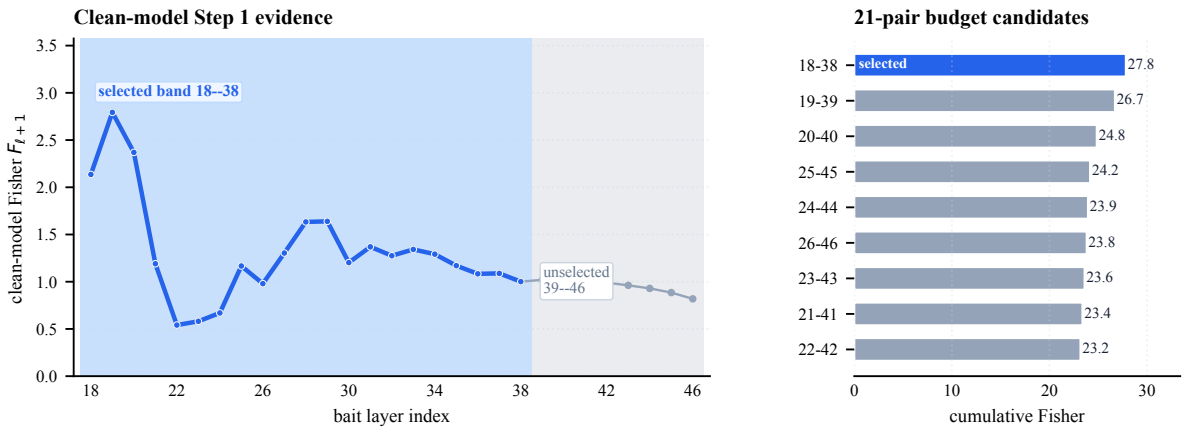


Figure 3: Clean-model layer-prior evidence for Gemma-3-12B. Given the $N_{\text{pair}} = 21$ budget, the selected $[18, 38]$ bait band is the highest-scoring contiguous 21-layer sub-band inside the default $[18, 46]$ clean-model prior window.

A Gemma-3-12B Layer-prior Evidence

Figure 3 shows the clean-model Fisher diagnostic used to select the Gemma-3-12B bait band: with the fixed $N_{\text{pair}} = 21$ pair budget, the contiguous-band selection rule from §4.1 picks $[18, 38]$ as the highest-scoring 21-layer sub-band inside the default $[18, 46]$ prior window.

Table 5: Full overall Heretic results. Models are abbreviated: Q3-4B (Qwen3-4B-Instruct), Q3-8B (Qwen3-8B), G3-1B (Gemma-3-1B-IT), G3-12B (Gemma-3-12B-IT).

KL	Q3-4B	Q3-8B	G3-1B	G3-12B
≤ 0.01	100/100/0	97/95/-2	88/NA/NA	72/82/+10
≤ 0.02	97/97/0	78/84/+6	88/84/-4	37/82/+45
≤ 0.03	97/78/-19	44/78/+34	77/84/+7	37/82/+45
≤ 0.05	97/78/-19	36/67/+31	17/84/+67	12/82/+70
≤ 0.075	97/78/-19	15/60/+45	17/67/+50	8/82/+74
≤ 0.10	46/78/+32	11/60/+49	5/67/+62	3/82/+79
≤ 0.15	24/70/+46	11/52/+41	5/64/+59	3/82/+79
≤ 0.20	24/65/+41	11/50/+39	3/64/+61	2/82/+80
≤ 0.30	6/65/+59	8/37/+29	2/64/+62	1/82/+81
≤ 0.50	5/65/+60	8/33/+25	2/64/+62	1/82/+81
≤ 1.00	5/60/+55	8/33/+25	2/64/+62	0/59/+59

Table 6: Full global-direction Heretic results.

KL	Q3-4B	Q3-8B	G3-1B	G3-12B
≤ 0.01	100/100/0	97/95/-2	88/NA/NA	89/83/-6
≤ 0.02	97/97/0	78/84/+6	88/84/-4	61/83/+22
≤ 0.03	97/78/-19	44/78/+34	77/84/+7	61/83/+22
≤ 0.05	97/78/-19	36/67/+31	17/84/+67	61/83/+22
≤ 0.075	97/78/-19	15/60/+45	17/67/+50	61/83/+22
≤ 0.10	46/78/+32	11/60/+49	5/67/+62	26/83/+57
≤ 0.15	24/78/+54	11/52/+41	5/64/+59	26/83/+57
≤ 0.20	24/78/+54	11/50/+39	3/64/+61	2/83/+81
≤ 0.30	6/78/+72	8/37/+29	2/64/+62	2/83/+81
≤ 0.50	5/78/+73	8/33/+25	2/64/+62	1/83/+82
≤ 1.00	5/78/+73	8/33/+25	2/64/+62	0/59/+59

Table 7: Full per-layer-direction Heretic results. "NA" means no trial landed inside the KL bucket.

KL	Q3-4B	Q3-8B	G3-1B	G3-12B
≤ 0.01	NA/NA/NA	NA/99/NA	94/NA/NA	72/82/+10
≤ 0.02	NA/NA/NA	NA/99/NA	90/NA/NA	37/82/+45
≤ 0.03	NA/NA/NA	100/99/-1	88/88/0	37/82/+45
≤ 0.05	100/95/-5	100/99/-1	78/86/+8	12/82/+70
≤ 0.075	99/95/-4	100/97/-3	68/82/+14	8/82/+74
≤ 0.10	94/81/-13	99/88/-11	22/82/+60	3/82/+79
≤ 0.15	94/70/-24	95/86/-9	10/82/+72	3/82/+79
≤ 0.20	88/65/-23	93/74/-19	8/80/+72	3/82/+79
≤ 0.30	88/65/-23	90/67/-23	6/80/+74	1/82/+81
≤ 0.50	56/65/+9	85/60/-25	3/77/+74	1/82/+81
≤ 1.00	28/60/+32	78/53/-25	3/73/+70	1/82/+81

B Detailed Heretic Evaluation Results

Tables 5–7 report the full evaluation results used to produce the selected main-text attack table. Each entry reports B/D/Δ, where $\Delta = \text{D} - \text{B}$.

C Qwen3-4B-Instruct Raw Fisher Values

Table 8 reports the raw Fisher separability scores for the validation split visualized in Figure 2.

D Reproducibility Details

We implement the framework in PyTorch (Paszke et al., 2019) with the Hugging Face Transformers library (Wolf et al., 2020). Models are loaded from their official Hugging Face or ModelScope weights and trained in `bfloat16`, with device-mapped model parallelism for the larger checkpoints. After training, the bait and recovery adapters are merged into the base weights, so the defended checkpoint runs as a standard causal LM with no custom inference code.

Layer-prior extraction and adapter training both use final-token residual activations from training

Table 8: Raw Fisher values for Qwen3-4B-Instruct. "Ref." denotes the before-pass reference used as the retention denominator.

		Bait		Recovery	
Pass	Layer	Ref.	Def.	Ref.	Def.
A	15	0.89	0.54	1.06	0.99
A	17	1.05	0.61	1.26	1.16
A	19	1.54	0.98	1.35	1.23
A	21	1.19	0.70	1.19	1.07
A	23	1.23	0.79	1.17	1.04
A	25	1.17	0.79	1.12	0.98
A	27	1.11	0.77	1.05	0.90
A	29	1.04	0.69	1.02	0.88
A	31	1.02	0.71	1.00	0.83
A	33	0.92	0.59	0.89	0.76
B	14	0.65	0.35	0.54	0.51
B	16	0.98	0.57	0.61	0.59
B	18	1.16	0.76	0.98	0.93
B	20	1.22	0.77	0.70	0.67
B	22	1.06	0.72	0.79	0.75
B	24	1.04	0.70	0.79	0.75
B	26	0.98	0.73	0.77	0.72
B	28	0.90	0.63	0.69	0.65
B	30	0.88	0.65	0.71	0.65
B	32	0.83	0.59	0.59	0.55
B	34	0.76	0.46	0.66	0.61

splits of the ModelScope-Alpaca and SafeMT datasets that are disjoint from the Heretic evaluation prompts. Heretic attacks are run with the Optuna search framework, using 200 trials per attack over projection layers, weight matrices, edit directions, and scales, and reporting the lowest-refusal edit within each KL budget. General-capability evaluation uses EvalScope with a vLLM backend, and we use the same chat template, generation length, and post-processing for the clean and defended checkpoints of each model when computing MMLU-Redux and IFEval.

E Additional Evaluations

E.1 Alternative Direction Estimators

We replace Heretic's mean-difference estimator with three alternatives on Gemma-3-12B-IT, using the same defended checkpoint as in the main experiments and the utility evaluation below. PC1 uses the leading principal component of the centered, pooled harmful and harmless residuals, with its sign set by the difference in class-mean projections. PCA top-8 projects the class-mean difference into the leading eight-dimensional PCA subspace. The geometric-median estimator takes the difference between class-wise geometric medians, computed using a stabilized Weiszfeld iteration (100 iterations at most, tolerance 10^{-5}).

Each run uses 400 harmful and 400 harmless prompts for estimation and 100 held-out harmful prompts for refusal measurement. The search has 200 Optuna trials, both global and per-layer scopes, batch size 32, and sampler seed 42. The residual readout is unchanged (`readout_shift=0`). Table 9 reports the minimum refusal rate over trials within each KL budget. Before editing, refusal is 97% for the clean checkpoint and 84% for the defended checkpoint.

	KL budget		
Estimator	0.10	0.20	1.0
PC1	59/83	3/83	0/55
PCA top-8	4/79	2/79	0/71
Geometric median	5/81	2/81	0/68

Table 9: Alternative estimators on Gemma-3-12B-IT. Entries are clean/defended minimum refusal rates (%) at KL no greater than the column budget; higher refusal indicates fewer successful refusal-removal edits.

At $\mathrm{KL} \leq 0.10$, PCA top-8 and geometric medians nearly eliminate clean-model refusal, whereas defended refusal remains at 79% and 81%. PC1 is weaker on the clean model at this budget. At $\mathrm{KL} \leq 1.0$, all three estimators reach 0% on the clean checkpoint, compared with 55–71% on the defended one. Over the full search, the lowest-refusal defended edits reach 3% at $\mathrm{KL} = 14.3921$ for PC1, 2% at $\mathrm{KL} = 26.0467$ for PCA top-8, and 7% at $\mathrm{KL} = 24.5542$ for geometric medians.

E.2 Comparison with Representation Rerouting

We compare Meta-Llama-3-8B-Instruct, Gray Swan AI's public RR checkpoint from Circuit Breakers (Zou et al., 2024),[3] and a Bait-and-Recover checkpoint trained from the clean model. The latter uses bait layers 12–30, with odd layers 13–29 in Pass A and even layers 12–30 in Pass B. All three models are attacked with the same 200-trial Heretic protocol, pooling global and per-layer searches.

		KL budget		
Checkpoint	Pre-edit	0.10	0.20	1.0
Clean	98	4	4	4
RR	53	2	1	0
Bait-and-Recover	98	90	62	21

Table 10: Llama-3-8B-Instruct defense comparison. Refusal rates are percentages over 100 harmful prompts. Post-edit columns report the minimum within each KL budget. The public RR checkpoint has a lower pre-edit refusal rate on these prompts.

Under Heretic editing, RR's refusal falls from 53% before editing to 2% at $\mathrm{KL} \leq 0.10$ (Table 10). Bait-and-Recover retains 90% at this budget, but falls to 21% at $\mathrm{KL} \leq 1.0$. The comparison concerns refusal-direction editing.

E.3 Additional Utility Benchmarks

We evaluate clean and defended Gemma-3-12B-IT checkpoints on GSM8K (Cobbe et al., 2021), Arena-Hard (Li et al., 2024), and the benign portion of XSTest (Röttger et al., 2024). GSM8K uses all 1,319 test examples with EvalScope's four-shot chain-of-thought setup and exact-answer accuracy. Arena-Hard uses the first 200 test prompts in dataset order (no shuffling), temperature 0, seed 42, and a 4,096-token output limit. EvalScope 1.7.1 compares each response with the dataset's GPT-4-0314 reference in both answer orders, using its built-in judge prompt and Elo-based win-rate aggregation. Strong preferences have weight three, ties count equally, and unrecognized verdicts are excluded (3/400 clean-model judgments; none for the defended model). XSTest uses all 250 benign prompts, the system prompt "You are a helpful assistant.", temperature 0, seed 42, a 2,048-token input limit, and a 256-token output limit, without a response prefix. Arena-Hard and XSTest responses are scored by DeepSeek V4 Flash.

Benchmark / metric	N	Clean	Def.
GSM8K accuracy	1,319	93.93	93.63
Arena-Hard judge win rate	200	84.42	84.58
XSTest helpful non-refusal	250	94.80	98.40
XSTest strict refusal proxy	250	3.20	1.60

Table 11: Additional Gemma-3-12B-IT utility results (%). Higher is better except for the strict refusal proxy. Arena-Hard scores use our subset and judge setup, rather than the official leaderboard protocol.

The clean/defended GSM8K difference is -0.30 percentage points, and the Arena-Hard difference is $+0.16$ points (Table 11). On benign XSTest prompts, the judge labels 237/250 clean and 246/250 defended responses as helpful, 13/250 and 3/250 as over-refusals, and 0/250 and 1/250 as invalid or empty. A separate refusal-pattern check flags 8/250 clean and 4/250 defended responses. The two XSTest measures use different criteria: the pattern check only detects refusal wording, while the judge also assesses helpfulness. These benign-prompt tests measure over-refusal, not safety on harmful requests.

[3] https://huggingface.co/GraySwanAI/Llama-3-8B-Instruct-RR