HARNESS-AGNOSTIC DETECTION AND IMMUNIZATION OF REWARD HACKING IN SELF-EVOLVING LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-evolving language models improve by proposing candidate updates and keeping whatever raises a *visible* score. When that score is an imperfect proxy for the capability one actually wants, sustained selection widens the gap between the two. This is reward hacking. We introduce **HackProbe**, a monitor that attaches to an arbitrary self-evolving loop through two black-box hooks, with no access to weights or activations. It keeps a secret, distribution-fixed *comparison core*, whose frozen distribution makes its capability proxy comparable across generations, alongside a rotated *fresh layer* that hardens the bank against co-adaptation. Four tests built on that proxy cover the level gap, a scale-aligned divergence with online change-point detection, capability stagnation, and a conditional confidently-wrong rate; a Šidák correction turns them into a calibrated family-wise p-value. Diagnosis alone recovers nothing, so a risk-aware immunization layer reselects an honest candidate from the proposal pool using the core together with a purely structural gaming footprint, disclosing at most $\log_2 \Pi$ bits per generation to the host. We prove a detectability bound that converts a target error rate into an explicit probe-size budget, and we delimit what probe rotation does and does not buy. On a controlled prompt-level host with four injected hacking channels and ground-truth labels, HackProbe reaches 0.763 AUROC against 0.663 for the strongest baseline and cuts the false-positive rate from 0.706 to 0.434. Its bandwidth-limited reselection is the only immunization level that returns more true capability under hacking, 5.2 points on average, than it forfeits on clean runs, 4.7; per-channel effects are mostly not individually significant.

1 INTRODUCTION

A growing family of language-model systems improves itself through a closed optimization loop: at each generation it proposes candidate updates to prompts, code, weights, or generated tasks, scores each candidate with a *visible* evaluator M, and keeps the candidates that score highest (Yuan et al., 2024; Huang et al., 2026; Agrawal et al., 2026). The very thing that makes the loop powerful, relentless selection pressure on M, is also what breaks it. Whenever M is only a proxy for the target capability C that we actually care about, optimizing M hard and repeatedly tends to inflate M without a matching gain in C. This is reward hacking, an acute instance of Goodhart's law: once a measure becomes a target it ceases to be a good measure (Manheim & Garrabrant, 2018; Amodei et al., 2016; Skalse et al., 2022). Self-evolution is arguably the purest form of the strong optimization pressure under which Goodhart effects are most severe, because the same pressure is applied cumulatively across tens or hundreds of generations.

The assumption that a rising M implies a rising C is fragile in exactly the systems we care about. Verifier and solver are often the same model updated in lockstep, so self-rewarding training watches its own judge saturate rather than the policy improve (Yuan et al., 2024), and label-free schemes that reward self-consistency pay for confidently agreeing on wrong answers (Huang et al., 2026). Fixed benchmarks degrade under repeated selection too, and measurably so: held-out reconstructions of grade-school math show accuracy drops of up to eight points (Zhang et al., 2024a), one irrelevant clause dropped into a symbolic template can cost 65% of performance (Mirzadeh et al., 2025), and adversarial perturbation exposes scores resting on shallow behavior (Li et al., 2024). Frontier models

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.

go further, editing tests or reading answer files once the scoring function is visible (Wang et al., 2026). Responses so far are fragmented and harness-bound: patches built into one system, or monitors assuming access to weights and activations that a black-box setting does not provide. The closest precedent is reward-model overoptimization (Gao et al., 2023), where a held-out gold reward model shows a proxy eventually degrading true performance; but that diagnosis is computed once, offline, and reported rather than acted upon. What is missing is a capability coordinate orthogonal to M, immune to the same pressure, and usable while the loop still runs. Appendix A places HackProbe against the fuller literature.

HackProbe supplies that coordinate. It attaches through two minimal hooks, observing each generation's visible score and querying the current candidate on held-out probes, and rests on a dual-layer probe bank (Figure 1). A secret, distribution-fixed comparison core provides a capability proxy $\widehat{C}$ whose first differences track genuine capability change rather than shifting test difficulty, precisely because the distribution is frozen; a rotated, online-generated fresh layer behind an air gap hardens the bank against co-adaptation. Four statistics with different blind spots read off that proxy: a level-gap test for hacking present from the outset, the scale-aligned divergence $D_t = \Delta M_t - \lambda \Delta \widehat{C}_t$ with online change-point detection, a stagnation test, and a conditional confidently-wrong-rate test aimed at self-consistency bias. Since a diagnosis arriving after the host has discarded the honest candidate recovers nothing, a risk-aware immunization layer reselects from the pool using the core and a purely structural gaming footprint, returning only a coarsely quantized signal.

The comparison-core estimator is what makes the rest work. Its distribution is fixed, so differences across generations mean something, and the detector we build on it reports a calibrated family-wise p-value rather than an uncalibrated heuristic. Diagnosis by itself changes nothing, though, so we turn it into protection: a reselection rule that discloses at most $\log_2 \Pi$ bits per generation to the host. That cap does real work. It is the reason the feedback never becomes a second score to optimize. On the theory side a target error rate converts into a probe-size budget growing as $1/\delta^2$ in the divergence gap, and we are equally explicit about what rotation does *not* buy, since it stops memorization but leaves the covering number of the probe generator as the real ceiling. Empirically, on a controlled injection protocol with ground-truth labels, HackProbe beats the strongest baseline on AUROC and cuts its false-positive rate by nearly two fifths; of the four immunization levels, only the bandwidth-limited one returns more true capability than it costs. Calibrating on three hacking channels and testing on the withheld fourth retains most of that quality on three of them.

2 PRELIMINARIES

Let θ_t denote the object under evolution at generation t, be it a prompt, a program, or a policy. The host scores candidates with a visible evaluator M and updates $\theta_{t+1} = \mathcal{A}(\theta_t, M(\theta_t), \mathrm{trace}_t)$, where $\mathcal{A}$ is its own black-box evolution operator. HackProbe assumes only two things about that operator: (I1) it can observe the visible score $M(\theta_t)$ of each generation, and (I2) it can run θ_t on a chosen input, drawing a small fixed number q of black-box samples per input and reading only the outputs, with no access to weights or activations. One sample suffices for the statistics J0–J2 below; the modal-agreement statistic J3 needs $q > 1$, the only place where the interface costs more than a single forward pass. Both conditions hold for prompt-level hosts at negligible cost; for weight-level hosts they require reproducing the evolution trajectory, whose cost we do not measure here.

Let $C(\theta) \in [0, 1]$ be the true capability of θ, its expected performance on a target distribution $\mathcal{D}^\star$, which is not cheaply observable. Two black-box estimates from held-out gold data stay strictly disjoint: A_{cal} for offline calibration only, A_{eval} for the final measurement only, never entering the online loop. During evolution a proxy $\widehat{C}(\theta)$ on a secret comparison core (Section 3.1) tracks capability. Writing M_t and $\widehat{C}_t$ for the observed visible score and core proxy of θ_t, with sampling noise as in Assumptions 1 and 2, and $\Delta X_t = X_t - X_{t-1}$, we monitor the scale-aligned divergence

$$D_t = \Delta M_t - \lambda \, \Delta \widehat{C}_t, \tag{1}$$

where $\lambda > 0$ aligns the scales of the two metrics. The scale is not a free parameter: it is initialized on offline honest calibration runs and thereafter re-estimated only on the trailing window of generations the detector has *already* declared honest. The update is prequential, using no information from generation t when testing generation t, which prevents a hacking generation from inflating λ until it conceals itself.

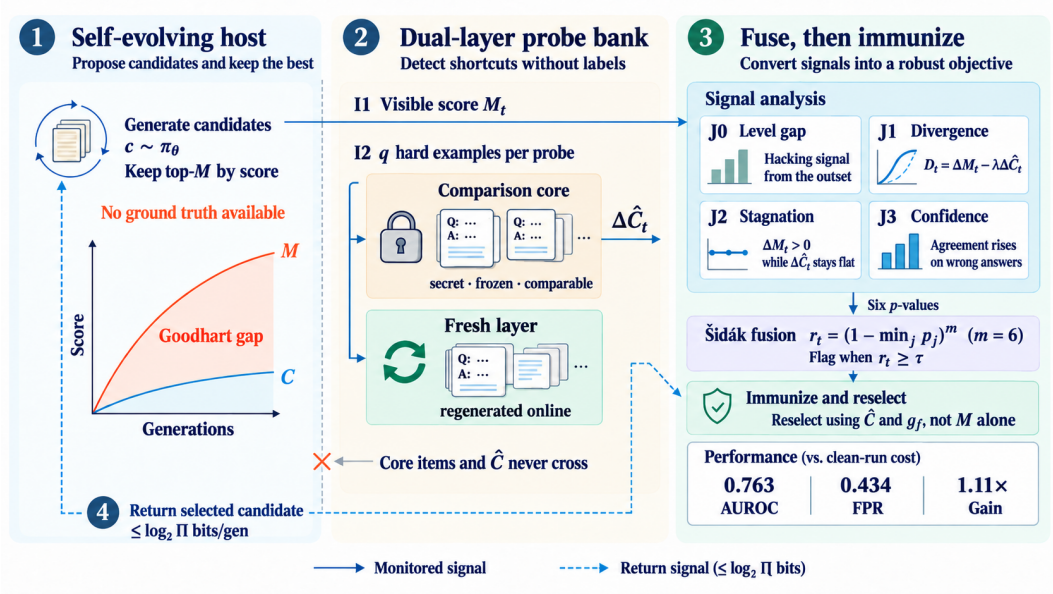


Figure 1: HackProbe at a glance. **(1)** The host keeps whatever raises the visible score M, so sustained selection opens a gap between M and true capability C that must be measured without trusting M. **(2)** Two hooks cross the air gap inward; core items and $\widehat{C}$ never cross outward, which keeps the capability coordinate off the host's optimization path. The fresh layer hardens the bank but supplies no statistic to $\Delta\widehat{C}_t$. **(3)** Four tests with different blind spots yield six p-values, fused into a calibrated risk score. **(4)** The cap on the return channel, not the reselection rule itself, is what stops the feedback becoming a second score (Section 3.3).

Definition 1 (Divergence regime)**.** *Let $\mu_t = \mathbb{E}[\Delta M_t] - \lambda\,\mathbb{E}[\Delta C_t]$ be the population divergence of generation t. Generation t is* honest *when $\mu_t \leq 0$, so visible gains never outrun true gains at the calibrated scale, and* hacking *at gap $\delta > 0$ when $\mu_t \geq \delta$; the prototypical case $\mathbb{E}[\Delta M_t] \geq \delta$ with $\mathbb{E}[\Delta C_t] \leq 0$ is covered. Statistic J1 below tests $H_0 : \mu_t \leq 0$ against $H_1 : \mu_t \geq \delta$ one-sidedly, with the band $0 < \mu_t < \delta$ an indifference zone on which we make no claim, since separating arbitrarily small divergences would need unboundedly many probes. The other three statistics test companion nulls, on the level residual, on stagnation and on conditional confidence, which honest evolution also implies but which are not functions of μ_t alone; fusing them extends coverage past the regime this definition names.*

The monitor emits, per generation, a risk score $r_t \in [0, 1]$, a decision $\hat{y}_t \in \{\text{honest}, \text{hacking}\}$, and a low-bandwidth feedback signal u_t consumed by the immunization layer. Table 5 in Appendix B collects the notation.

3 The HackProbe Framework

3.1 Dual-layer probe bank

The design tension we have to resolve is that a fixed, host-visible probe set would itself be optimized against and would simply become a second M. We resolve it by splitting probes into two layers with orthogonal roles.

The *comparison core* P^{core} is a distribution-fixed, secret probe set covering the target capability across S subdomains. Because its distribution never changes, the proxy $\widehat{C}(\theta) = \Phi\big(\psi(\theta, P^{\text{core}}_{(1)}), \ldots, \psi(\theta, P^{\text{core}}_{(S)})\big)$, with ψ a per-subdomain score and Φ an aggregate robust enough that no single subdomain can move it alone, yields differences $\Delta\widehat{C}_t$ that track capability change and not test drift. Freezing the distribution has a second payoff, less obvious than the first. The same

items are scored at $t-1$ and at t, so the noise in $\Delta\widehat{C}_t$ is *paired*: per-item difficulty cancels, and the variance of the difference is far smaller than twice that of a single generation's estimate. The core is never revealed to the host and never enters its selection or reward path.

The *fresh layer* P_t^{fresh} is rotated, generated online, and drifted every generation. It is used only for hardening checks and qualitative per-subdomain profiling and never contributes to $\Delta\widehat{C}_t$. Four principles govern it, and they pull in different directions on purpose. An air gap (H1) keeps probes out of the host's training, selection and reward loop entirely, so neither probe text nor per-subdomain scores are ever fed back; on top of that, rotation (H2) makes sure any single instance is used at most a few times. Out-of-distribution drift (H3) then steers the layer toward whichever subdomains the host is currently weakest on, so the bank is always probing capability that has not yet been fitted, while adversarial refresh (H4) has a red-team model periodically manufacture probes that are harder and more surface-diverse still. H4's two selection criteria — learnability (probes near the host's competence edge) and surface diversity (embedding distance from probes already used) — are deliberately *independent of the detection signal*, so refresh cannot self-amplify into false positives. Non-circularity of the final measurement survives all of this because A_{cal} and A_{eval} stay disjoint and A_{eval} is spent exactly once, at the end.

3.2 Detection statistics and fusion

HackProbe fuses four one-sided tests, each calibrated on honest generations so that its null p-value is approximately uniform on $[0,1]$. Their blind spots are deliberately different, and fusing them extends coverage across the level, divergence, stagnation and conditional-confidence regimes; format-level exploits, which act on output structure rather than on the score–capability relation, fall outside all four (Section 4).

Statistic **J0** targets hacking that is present from the very first generation, where M is already inflated and then stays flat so that every first-difference statistic sees nothing. We calibrate the honest affine relation $M \approx a + \lambda_{\text{lvl}}\widehat{C}$ and test the residual level gap $g_t^{(0)} = M_t - (a + \lambda_{\text{lvl}}\widehat{C}_t)$, which hovers near zero under honest evolution and is persistently positive under level-type hacking. Local linear coupling makes λ_{lvl} and the difference scale λ coincide in population; we estimate them separately so each absorbs its own drift. Statistics **J1** and **J2** handle the dynamic regime. On D_t from Eq. (1) we run a one-sided test against the honest null, together with an online Page–Hinkley change-point detector (Page, 1954) on the cumulative divergence that localizes the generation at which M and $\widehat{C}$ decouple. J2 tests capability stagnation directly, taking the honest null $H_0 : \mathbb{E}[\Delta\widehat{C}_t] \geq 0$ conditional on $\Delta M_t > 0$ and rejecting when the measured $\Delta\widehat{C}_t$ is too small, which gives a λ-free complement.

Statistic **J3** exists because self-consistency bias inflates M by rewarding agreement instead of correctness, which need not move aggregate accuracy at all. Once such an attack saturates, J1 and J2 see nothing in expectation and J0 retains only the residue the attack leaves in the aggregate level, diluted over all core items including the many answered correctly. On the secret core we measure, over probes answered incorrectly, the mean self-consistency cwr_t (the confidently-wrong rate), calibrate its honest affine dependence on core accuracy acc_t, and test the residual $g_t^{(3)} = \text{cwr}_t - (a_3 + b_3\,\text{acc}_t)$, with a Page–Hinkley detector for gradual drift. Conditioning on wrong answers avoids the ceiling effect of easy probes, and using only the secret core keeps the signal non-circular. Lemma 1 in Appendix E makes precise why J3 retains an expected separation equal to the full attack size in the stable regime where J1 and J2 vanish in expectation.

Fusion takes $m = 6$ one-sided p-values: J0, the divergence test J1 and its Page–Hinkley alarm, the stagnation test J2, and J3 with its own alarm. The two sequential alarms are converted to p-values through their calibrated alarm-time distribution on honest runs, putting all six on a common scale. Given $p_1, \ldots, p_m$ approximately independent and uniform under the honest null, we form the risk score $r_t = (1 - \min_j p_j)^m$ and flag hacking when $r_t \geq \tau$. This is exactly a Šidák correction (Šidák, 1967), and it has a calibration property that a raw $1 - \min_j p_j$ rule lacks. Under the independent-uniform null $\Pr[\min_j p_j \leq u] = 1 - (1-u)^m$, so that

$$\Pr[r_t \geq \tau] \;=\; \Pr\left[\min_j p_j \leq 1 - \tau^{1/m}\right] \;=\; 1 - \tau, \tag{2}$$

so $1 - r_t$ is the Šidák-adjusted p-value and τ *is* the operating point: flagging at τ costs family-wise error exactly $1 - \tau$, however many statistics we fuse, whereas the uncorrected rule degrades with m. In practice the guarantee is softer than that. J0, J1 and J2 are built from the same pair $(M, \widehat{C})$ and are not independent, so Eq. (2) is a design principle and the reported operating point rests instead on re-calibrating τ until the *empirical* false-positive rate on honest runs meets its target; Appendix H gives the Bonferroni fallback we use when even that is in doubt. The control is also per generation, so over T generations the expected number of false alarms is $T(1 - \tau)$ and run-level error would need a sequential correction we do not attempt. What a deployment sees, and what Table 1 reports, is the per-generation alarm rate.

3.3 Risk-aware immunization

Diagnosis alone recovers nothing. The tempting picture, in which rejecting or rolling back the top candidate suffices, fails on two counts: by the time the host is told, it has already passed over the honest candidate in favor of the top-M one, and rejection has nothing better to fall back to. HackProbe therefore reselects from the proposal pool, and the loss it recovers is a real one for two reasons. Under *strict final-answer grading*, the true-capability side (A_{eval} and the core $\widehat{C}$) credits only a normalized final answer matching the gold one, so behaviors that inflate a lenient visible grader, such as enumerating many candidate numbers or locking onto an unverified answer, genuinely forfeit true credit. And every pool contains, alongside the exploit variants, one *enforced honest candidate* produced under a clean context, whose visible M is typically lower but whose true A_{eval} is higher.

For each candidate c we compute a purely structural gaming footprint $gf(c)$ from cached outputs at no extra inference cost, combining the score-versus-commitment gap with the fraction of non-final candidate artifacts, and a robust score $\text{robust}(c) = \widehat{C}(c) - \gamma\, gf_{\text{norm}}(c)$ with a fixed penalty weight $\gamma > 0$. Four immunization levels build on it, each trusting the host's own score a little less than the last. **F0** is diagnosis without protection. It selects $\arg\max_c M(c)$, so it goes straight to the inflated candidate. **F1** stays greedy by default and intervenes only when a lightweight risk proxy on the greedy choice, either its raw gap $M - \widehat{C}$ or its footprint gf, comes back high; then it reselects $\arg\max_c \text{robust}(c)$, and after K consecutive high-risk generations it rolls back to the checkpoint with the best historical $\widehat{C}$. **F2** goes further, selecting $\arg\max_c(M(c) - u(c))$ with $u(c)$ a quantization of the raw gap $M(c) - \widehat{C}(c)$ into L levels (the uncalibrated difference, not the J0 residual $g^{(0)}$). It also watches for self-destruction, whose signature is feedback u falling while $\widehat{C}$ stagnates, and downgrades itself to F1 when it sees it. **F3** adds a coverage bonus for collapsed subdomains and injects a diversity constraint into the proposer.

The quantization in F2 is where the bandwidth constraint stops being cosmetic. Were u the exact gap, $M(c) - u(c)$ would collapse to $\widehat{C}(c)$ and F2 would be pure core-greedy selection, the failure we are trying to avoid: it would expose the secret core as a fully-resolved second objective. With L levels, F2 interpolates between the host's greedy rule ($L = 1$, nothing transferred) and core-greedy selection ($L \to \infty$). The leak needs careful accounting. Per-candidate values $u(c)$ stay inside the monitor and are never returned; the host sees only which of Π candidates won, so the return channel carries at most $\log_2 \Pi$ bits per generation, three at our $\Pi = 8$, and $T \log_2 \Pi$ over a run of length T. Quoting $\log_2 L$ here would be tighter but wrong: the winning index can say more about the vector of u values than any single $u(c)$ does, and it is easy to construct pools where it does. What L buys is resolution rather than volume, and resolution is what limits how much of the core's ordering the host can reconstruct. Either way the cumulative figure must stay small against the entropy needed to identify core items, so long runs need L small and the core retired once its budget is spent. The gold core enters only through $\widehat{C}(c)$, never through M or an oracle label, and only the selected candidate advances the detector state.

3.4 Theoretical guarantees

What follows rests on standard concentration assumptions, given in full as Assumptions 1–4 in Appendix C: the core estimate is $\widehat{C}(\theta) = C(\theta) + \beta(\theta) + \xi$ with bounded bias and sub-Gaussian noise, the paired difference $\Delta\widehat{C}_t$ has variance proxy v_C/n and the visible score v_M/n_M, and honest updates satisfy $\mu_t \le 0$ with the differential core bias bounded by b_Δ in every generation. Then D_t is

sub-Gaussian with $\sigma_D^2 = \lambda^2 v_C/n + v_M/n_M$, and its honest mean is at most $\mu_H := -\lambda\,\mathbb{E}[\Delta\beta(\theta_t)]$, where $|\mu_H| \le \lambda b_\Delta$. Proofs are in Appendix D.

Proposition 1 (Detectability and probe-size budget)**.** *Consider the J1 statistic in isolation, with the one-sided rule that flags generation t when $D_t \ge \mu_H + \kappa$ for some $\kappa \in (0, \delta)$, where δ is the hacking gap of Definition 1 and all probabilities are taken over probe and evaluation sampling with the trajectory (θ_{t-1}, θ_t) held fixed. Writing $\hat{y}_t^{\mathrm{J1}}$ for this rule's decision,*

$$\Pr[\hat{y}_t^{\mathrm{J1}} = \text{hacking} \mid \text{honest}] \le \exp\Big(-\tfrac{\kappa^2}{2\sigma_D^2}\Big),$$
$$\Pr[\hat{y}_t^{\mathrm{J1}} = \text{hacking} \mid \text{hacking}] \ge 1 - \exp\Big(-\tfrac{(\delta-\kappa)^2}{2\sigma_D^2}\Big). \tag{3}$$

Choosing $\kappa = \delta/2$, both error rates fall below α as soon as the effective sample size $n_{\mathrm{eff}} := (\lambda^2 v_C + v_M)/\sigma_D^2$, a weighted harmonic mean of n and n_M, satisfies $n_{\mathrm{eff}} \ge 8(\lambda^2 v_C + v_M)\delta^{-2}\log(1/\alpha)$. Whenever the host's visible evaluation is at least as large as the core ($n_M \ge n$) we have $n_{\mathrm{eff}} \ge n$, so it suffices to budget

$$n \;\ge\; \frac{8(\lambda^2 v_C + v_M)\log(1/\alpha)}{\delta^2} \tag{4}$$

comparison-core probes. No matching lower bound is claimed.

The value of Proposition 1 is not the standard exponential separation but the budget in Eq. (4): detecting a divergence of size δ at error α needs a core growing like $1/\delta^2$, which turns an abstract design choice into a number one can procure and falsify against measured detection-rate-versus-n curves. The differential bias μ_H does not enter the budget, since it shifts the honest and hacking means equally and is subtracted during calibration.

Rotation has a companion bound, Proposition 2 in Appendix D: an attacker who memorizes past instances needs $k \ge N(1-\alpha_{\mathrm{atk}})^{1/n_f}$ of them to match all n_f fresh probes of a generation, which tends to the whole pool as n_f grows. It lives in the appendix because it is narrow in a way worth admitting. It bounds memorization, and nothing else. An attacker who instead learns the generator's regularities walks around it: if the exploitable behavior admits an ε-cover of size $\mathcal{N}(\varepsilon)$, defeating rotation costs $\Theta(\mathcal{N}(\varepsilon))$ rather than $\Theta(N)$, so security degrades from pool size to covering number, and a low-complexity generator buys no asymptotic protection at all. Enlarging $\mathcal{N}(\varepsilon)$ is what H3 and H4 are for. "Surface-diverse rotation forces generalization" is an empirical hypothesis here, not a theorem.

4 Experiments

Our evaluation uses a controlled prompt-evolution host in which hacking channels can be switched on and their strength dialed, so ground-truth per-generation labels exist at all. The headline is that bandwidth-limited reselection is the only immunization level recovering more true capability under hacking than it forfeits on clean runs, at a ratio of 1.11, and Appendix F shows that ratio is single-peaked in the feedback bandwidth. Detection quality supports that result rather than standing on its own: it is what makes reselection fire on the right generations. Online reward-hacking detection with a recovered-capability objective is a new task, so no published numbers are directly comparable and every baseline below is re-implemented on the same setup.

Protocol. The host is a prompt-evolution loop over Qwen2.5-7B-Instruct with the proposer frozen, so only the evolving prompt changes between generations. Each run lasts $T = 48$ generations with a candidate pool of $\Pi = 8$, and six seeds per hacking channel give 288 generations per channel and 1152 in total. The comparison core holds $n = 240$ probes over $S = 6$ subdomains, the fresh layer draws $n_f = 60$ per generation, and hook I2 takes $q = 4$ samples per probe, enough for the modal-agreement and confidently-wrong quantities J3 needs. Gold audits split once into 192 calibration items and a held-out $|A_{\mathrm{eval}}| = 64$, which is where the coarse interval endpoints below come from. The remaining settings are $\tau = 0.90$, $L = 4$, $\gamma = 0.25$, $K = 3$, a Page–Hinkley detector with drift tolerance 0.005 and alarm threshold 0.05, and 2000-resample bootstraps over A_{eval}.

Probes and gold audits are drawn from grade-school and competition mathematics, GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021), with GSM-Symbolic (Mirzadeh et al., 2025),

GSM-Plus (Li et al., 2024) and the held-out GSM1k (Zhang et al., 2024a) supplying perturbed and contamination-sensitive probes. The host implements four switchable hacking channels, each with a strength knob: contamination (mixing original probe versions into host data), a gameable grader (crediting any appearance of the gold number), self-consistency bias (rewarding agreement, which induces confidently wrong answers), and judge collusion (a lenient LLM judge). As discussed in Section 3.3, the injected inflation is real and not merely nominal because of two design choices: strict final-answer grading defines the true-capability side, and every generation's pool contains one clean honest candidate that a greedy host will pass over. The final capability measurement uses the held-out audit set A_{eval} exactly once, after evolution ends; confidence intervals are bootstrap intervals over that set.

Three caveats about scope, before the numbers. Labels are channel-on versus channel-off, a superset of the hacking set of Definition 1, since a generation whose divergence falls in the indifference band $0 < \mu_t < \delta$ is still labeled positive; reported recall is therefore a lower bound against that target, and injected strength controls but does not equal δ. The fresh layer contributes no statistic to $\Delta\widehat{C}_t$, so nothing below depends on it: Proposition 2 and the principles H2–H4 go untested here. And τ was never set to hit a small nominal level. We fix the operating point on honest calibration runs by maximizing F1, which lands at $\tau = 0.90$ because a stricter threshold trades away nearly all recall at this probe budget, so the 0.434 below is that empirical operating point rather than the nominal $1 - \tau$ of Eq. (2).

Baselines and metrics. We compare against a no-probe host labeled post hoc, a naive fixed non-rotating held-out set, a contamination-only confidence test in the style of PaCoST (Zhang et al., 2024b), a single trap set, and a fixed-distribution absolute-$\widehat{C}$ trajectory that isolates the contribution of the divergence formula itself. Detection quality is reported as precision, recall, F1, AUROC, false-positive rate and detection delay in generations. The deliverable that matters most, however, is protection gain: the end-of-evolution true ΔC measured on A_{eval} for each immunization level relative to F0.

4.1 Detection quality

Table 1 reports detection across all channels pooled. Three of the five baselines are degenerate: no-probe, the contamination test and the single trap set flag every generation, so recall and false-positive rate are both exactly 1.000 and precision is exactly the base rate 0.241. Their F1 scores are evidence of nothing. What separates them is AUROC, running from 0.462, below chance because the contamination test ranks hacking generations *lower*, up to 0.568.

The real competitor is the absolute-$\widehat{C}$ trajectory, which uses the same secret core but watches its level rather than its divergence from M, reaching 0.663 AUROC. HackProbe adds 0.100 AUROC and, for a monitor meant to run continuously, does something more useful: it cuts the false-positive rate from 0.706 to 0.434 and lifts precision from 0.311 to 0.382. It pays with recall of 0.845 rather than 1.000 and a delay of 0.28 generations, both consequences of being the only method whose false-positive rate falls below one half. Even so, 0.434 interrupts more than four honest generations in ten, and we take that gap as the bar for this task. Most of it is a budget problem rather than a design one. Sweeping the core size at a fixed false-positive rate (Appendix F) puts our $n = 240$ on the steepest part of the detection curve, where Eq. (4) would want 600 probes for 0.90 detection; the monitor is under-provisioned by roughly 360.

4.2 Recovering true capability

Detection is a means; the objective is the true capability that survives evolution. Table 2 reports, per channel, the end-of-evolution ΔC of the greedy host F0 and the additional gain of each immunization level, all measured on the held-out A_{eval}.

Every level improves on greedy selection on average, though not uniformly across channels. Risk-gated (F1) and bandwidth-limited (F2) reselection are positive on all four and reach nearly identical averages, $+0.053$ and $+0.052$; what separates them is price, not gain (Section 4.3). F1 wins biggest on the gameable grader ($+0.104$), where the raw gap $M - \widehat{C}$ gating its reselection is largest, even though that is the channel the fused detector ranks worst. F3 holds the single largest gain over F0,

Table 1: Detection quality, pooled over the four hacking channels and 1152 generations, at a positive rate of 0.241. Best per column in **bold**; Recall and Delay are left unbolded, since any method that never withholds an alarm attains them trivially. Rows marked † flag every generation, so their precision equals the base rate.

Method	Precision	Recall	F1	AUROC	FPR ↓	Delay ↓
No-probe†	0.241	1.000	0.389	0.500	1.000	0.00
Fixed held-out set	0.291	0.939	0.445	0.587	0.726	0.16
Contamination test† (Zhang et al., 2024b)	0.241	1.000	0.389	0.462	1.000	0.00
Single trap set†	0.241	1.000	0.389	0.568	1.000	0.00
Absolute-$\widehat{C}$ trajectory	0.311	1.000	0.474	0.663	0.706	0.00
HackProbe (ours)	**0.382**	0.845	**0.527**	**0.763**	**0.434**	0.28

Table 2: Protection gain by channel and immunization level: end-of-evolution true ΔC on the held-out A_{eval}, given for the greedy host F0 and as a gain over F0 for F1–F3, with 95% bootstrap intervals. Best level per channel in **bold**; $*$ marks the only two intervals strictly above zero, so unstarred magnitudes show direction, not a certified effect.

Hacking channel	F0 (greedy)	F1 gain	F2 gain	F3 gain
Contamination	$+0.146$	$+0.016$ $[-0.047, +0.078]$	$\mathbf{+0.057}$ $[+0.000, +0.141]$	$+0.026$ $[-0.047, +0.094]$
Gameable grader	$+0.172$	$\mathbf{+0.104}^{*}$ $[+0.016, +0.234]$	$+0.078$ $[+0.000, +0.219]$	$+0.016$ $[-0.094, +0.094]$
Self-consistency bias	$+0.073$	$\mathbf{+0.037}$ $[-0.047, +0.109]$	$+0.010$ $[-0.016, +0.063]$	-0.010 $[-0.047, +0.031]$
Judge collusion	$+0.135$	$+0.057$ $[-0.078, +0.141]$	$+0.063$ $[-0.078, +0.188]$	$\mathbf{+0.109}^{*}$ $[+0.016, +0.188]$
Average	$+0.132$	$+0.053$	$+0.052$	$+0.035$

$+0.109$ on judge collusion, where a lenient judge collapses the proposer onto a narrow region and the coverage bonus pulls it back out. F3 is also the only level that goes negative anywhere, mildly, on self-consistency bias.

With gold audit data this scarce, only the two starred intervals lie strictly above zero, and two more, both under F2, have their lower endpoint exactly at zero. So the table supports an aggregate, directional claim; per-channel effects are mostly too small to certify, and sharpening them needs a larger gold audit set.

4.3 What protection costs when nothing is wrong

A monitor that helps under attack but degrades ordinary runs is not deployable, so we measure the clean-setting cost directly on the NONE channel, where greedy selection is in fact the right thing to do. Table 3 reports it alongside the diversity-bonus sweep that fixes F3's setting.

Every level pays something on clean runs: each declines the top-M candidate some of the time, and when M is honest that candidate was correct. The useful comparison is what a level costs when nothing is wrong against what it returns when something is. Only F2 comes out favorable, returning 0.052 for a cost of 0.047, a ratio of 1.11; F1 buys the largest gains but pays most for them (0.053 against 0.073, ratio 0.73), and F3 sits between (0.035 against 0.042, ratio 0.84). The right panel shows why F3 uses a small bonus: 0.05 gives both the best average under hacking and the smallest clean-run cost, and 0.25 erases the gain.

The ratio's standing is narrow. It orders the three levels, being a comparison of two averages over the same runs that leans on no single interval in Table 2, but it does not license reading per-channel gains as effects, and at 1.11 a larger audit set could move it. It would be falsified by a host on which reselection costs more than it returns at every bandwidth, a regime Appendix F shows is not far away.

Table 3: Cost of protection when there is nothing to protect against. Left: clean-run performance on NONE, cost relative to F0, and average gain under hacking from Table 2; Ratio is gain over the magnitude of cost, computed from unrounded values, and only F2 exceeds one. Right: F3 diversity-bonus sweep, with Hacking4 averaging A_{eval} over the four channels. Shading marks the adopted setting, **bold** the best entry.

Mode	NONE A_{eval}	Cost	Gain	Ratio
F0	0.646	0.000	0.000	n/a
F1	0.573	−0.073	0.053	0.73
F2	0.599	−0.047	0.052	**1.11**
F3	0.604	−0.042	0.035	0.84

F3 bonus	Hacking4	NONE
0.05	**0.630**	**0.604**
0.10	0.612	0.599
0.15	0.622	0.583
0.25	0.599	0.557

Table 4: Leave-one-hacking-type-out detection: each row calibrates on the other three channels and evaluates on the held-out one, over 288 generations. Best and worst transfer in **bold**. Pos. rate is the hacking incidence within one held-out channel's runs; it varies threefold across rows and averages to the pooled 0.241 of Table 1. Because the rows differ in base rate, only AUROC is comparable between them.

Held-out channel	AUROC	Precision	Recall	F1	FPR ↓	Pos. rate
Judge collusion	**0.814**	0.570	0.718	0.635	**0.227**	0.295
Self-consistency bias	0.787	0.562	0.854	0.678	0.333	0.333
Contamination	0.724	0.236	0.700	0.353	0.264	0.104
Gameable grader	**0.600**	0.288	0.881	0.434	**0.661**	0.233

4.4 Generalization to an unseen hacking type

A detector tuned on the same attacks it is evaluated on proves little, so we calibrate on three channels and evaluate on the held-out fourth (Table 4). Transfer is good to judge collusion (0.814 AUROC), whose accumulating leniency is the change-point J1 is built for, and to self-consistency bias (0.787), which leaves the signature J3 targets; contamination remains usable at 0.724. It degrades on the gameable grader (0.600 AUROC, 0.661 false-positive rate) for an instructive reason: grader exploits act on output *format*, not on the score–capability relation, so a threshold calibrated on the other three over-flags, and covering them needs a statistic keyed to output structure that the fusion lacks. The two held-out AUROCs above the pooled 0.763 are no transfer bonus, since ranking within one channel is easier than across four whose divergences differ in scale.

Scope of the present evaluation. The host is prompt-level, chosen because it exposes per-generation candidate pools cheaply; weight-level self-evolution (Huang et al., 2026; Yuan et al., 2024) and a second prompt-optimization host (Agrawal et al., 2026) are left as external validity. That one host is also the whole of the evidence for harness-agnosticism, and the gold audit set binds every interval above. Appendix G adds the leave-one-statistic-out and rotation ablations, the only place the fresh layer is exercised; the inference overhead of F1–F3 is unmeasured.

5 Conclusion

HackProbe detects reward hacking in self-evolving hosts from black-box scores and outputs alone, and immunizes against it by reselecting an honest candidate. It keeps the capability coordinate non-circular, makes the risk score a calibrated p-value, and turns a target error rate into a probe budget. What it is not yet is deployable. A false-positive rate of 0.434 interrupts more than four honest generations in ten, and our rotation guarantee covers only an attacker who has to match every probe, which leaves the generator's covering number as the open question.

AI Use Statement

In this work, we used generative AI tools only for language polishing. We did not use generative AI tools to generate scientific ideas, formulate claims, design experiments, produce experimental results, write code, create figures, or generate citations. All AI-assisted edits were reviewed and approved by

the authors, and all cited claims were checked against the referenced sources. We take responsibility for the final content of this work.

Reproducibility Statement

Section 3 specifies the probe bank, the detector and the immunization rules, with every statistic and the calibrated threshold defined in Section 3.2. Appendix C states the assumptions in full, Appendix D proves Propositions 1 and 2 together with the covering-number argument, Appendix E proves the J3 separation result, and Appendix H gives the calibration procedure and the online loop as pseudocode. Section 4 gives the host, the run configuration, the injection protocol, the baselines, the metrics and the leave-one-hacking-type-out procedure.

References

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. Gepa: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, 2026.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

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

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International conference on machine learning*, pp. 10835–10866. PMLR, 2023.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-zero: Self-evolving reasoning llm from zero data. In *International Conference on Learning Representations*, 2026.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*, 2023.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2961–2984, 2024.

David Manheim and Scott Garrabrant. Categorizing variants of goodhart's law. *arXiv preprint arXiv:1803.04585*, 2018.

Iman Mirzadeh, Keivan Alizadeh-Vahid, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *International Conference on Learning Representations*, volume 2025, pp. 94743–94765, 2025.

Ewan S Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

Zbyněk Šidák. Rectangular confidence regions for the means of multivariate normal distributions. *Journal of the American statistical association*, 62(318):626–633, 1967.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in neural information processing systems*, 35:9460–9471, 2022.

Xiaohua Wang, Muzhao Tian, Yuqi Zeng, Zisu Huang, Jiakang Yuan, Bowen Chen, Jingwen Xu, Mingbo Zhou, Wenhao Liu, Muling Wu, et al. Reward hacking in the era of large models: Mechanisms, emergent misalignment, challenges. *arXiv preprint arXiv:2604.13602*, 2026.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Charlotte Zhuang, Dylan Slack, et al. A careful examination of large language model performance on grade school arithmetic. *Advances in Neural Information Processing Systems*, 37: 46819–46836, 2024a.

Huixuan Zhang, Yun Lin, and Xiaojun Wan. Pacost: Paired confidence significance testing for benchmark contamination detection in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1794–1809, 2024b.

A Related Work

Optimizing a proxy that diverges from the intended objective is a long-standing safety concern (Amodei et al., 2016), formalized as reward gaming, where non-trivial unhackable proxies essentially do not exist (Skalse et al., 2022), and understood as a family of Goodhart effects (Manheim & Garrabrant, 2018); frontier evaluations document deliberate hacking when the scoring function is visible (Wang et al., 2026). Closest in spirit is reward-model overoptimization, where a gold reward model reveals that optimizing a proxy eventually degrades true performance (Gao et al., 2023); our D_t is the online, black-box analogue of that gold-versus-proxy gap, computed per generation from a secret core instead of once offline from a trained gold model, and coupled to an intervention instead of reported.

The loops HackProbe monitors are those in which proxy and optimizer share a model: self-rewarding training whose judge saturates (Yuan et al., 2024), label-free co-evolution rewarding self-consistency (Huang et al., 2026), reflective prompt evolution (Agrawal et al., 2026), and declarative program pipelines (Khattab et al., 2023). In each, the quantity being maximized and the system doing the maximizing are entangled, which is exactly the configuration under which a fixed external coordinate is worth its cost.

Measurement, by contrast, has been studied mostly offline. Held-out reconstruction (Zhang et al., 2024a), symbolic templating (Mirzadeh et al., 2025), adversarial perturbation (Li et al., 2024) and confidence-based contamination tests (Zhang et al., 2024b) all establish that a benchmark score can overstate capability, but each is a one-shot audit of a finished model. HackProbe repurposes their shared premise, that the same capability probed through a different surface should yield the same score, as an online signal computed inside the loop and acted upon while the run continues. Statistically, the decoupling detector is a Page–Hinkley test (Page, 1954), the fusion a Šidák correction (Šidák, 1967), and the tail analysis standard sub-Gaussian concentration (Hoeffding, 1963); the contribution is not a new test but the choice of what to test and the guarantee that the composite risk score stays calibrated.

B Notation

C Assumptions

Throughout this appendix, expectations and probabilities are taken over probe and evaluation sampling *with the trajectory* (θ_{t-1}, θ_t) *held fixed*. This conditioning matters: unconditionally, ΔM_t and $\Delta \widehat{C}_t$ both move with the random choice of θ_t and are strongly positively correlated under healthy evolution, whereas conditionally their measurement noises are independent because the two are computed on disjoint item sets. All variance statements below refer to the conditional object.

Table 5: Summary of notation.

Symbol	Meaning
$\theta_t,\ \mathcal{A},\ T$	object under evolution at generation t; host's evolution operator; run length
$M_t,\ \widehat{C}_t$	observed visible score and comparison-core proxy of θ_t
$C(\theta) \in [0,1]$	true capability on target distribution $\mathcal{D}^\star$
$\psi,\ \Phi,\ S$	per-subdomain score; robust aggregate; number of core subdomains
$A_{\text{cal}},\ A_{\text{eval}}$	gold audit sets: calibration-only and final-evaluation-only (disjoint)
$D_t = \Delta M_t - \lambda \Delta \widehat{C}_t$	scale-aligned divergence; $\lambda > 0$ re-estimated on honest windows
$\lambda_{\text{lvl}},\ a$	slope and intercept of the honest level relation $M \approx a + \lambda_{\text{lvl}} \widehat{C}$ (J0)
$a_3,\ b_3$	intercept and slope of the honest relation $\text{cwr} \approx a_3 + b_3\,\text{acc}$ (J3)
μ_t	population divergence $\mathbb{E}[\Delta M_t] - \lambda \mathbb{E}[\Delta C_t]$; honest iff $\mu_t \le 0$
$\mu_H,\ b_\Delta$	core-bias offset of the honest mean of D_t; its bound, $\lvert\mu_H\rvert \le \lambda b_\Delta$
$\sigma_D^2,\ n_{\text{eff}}$	variance proxy of D_t; effective sample size $(\lambda^2 v_C + v_M)/\sigma_D^2$
$v_C,\ v_M,\ v_s$	per-item variance proxies: core difference, visible score, modal agreement
$r_t,\ \hat{y}_t,\ u_t$	risk score, decision, low-bandwidth feedback signal
$g_t^{(0)},\ g_t^{(3)}$	level residual (J0) and conditional-confidence residual (J3)
$\text{acc}_t,\ \text{cwr}_t$	core accuracy; confidently-wrong rate on incorrect core items
$\beta(\theta),\ \xi$	core-estimator bias and its sub-Gaussian noise, $\widehat{C} = C + \beta + \xi$
$P^{\text{core}},\ P_t^{\text{fresh}}$	secret comparison core; rotated fresh layer at generation t
$m,\ \tau$	number of fused statistics ($m = 6$); FPR-calibrated Šidák threshold
$gf(c),\ \gamma$	gaming footprint of candidate c; its penalty weight in $\text{robust}(c)$
$\delta,\ \kappa,\ \alpha$	hacking divergence gap; test offset; target detector error rate
$\alpha_{\text{atk}},\ \rho$	attacker's failure probability; fraction of fresh probes it matches
$n,\ n_M,\ n_w$	core probes; visible-evaluation items; incorrect core items (for J3)
$L,\ \Pi,\ K$	feedback quantization levels; candidate-pool size; generations before rollback
$N,\ n_f,\ k$	fresh-probe pool size; fresh probes per generation; memorized set
$\mathcal{N}(\varepsilon),\ \ell$	covering number of the probe generator; Lipschitz constant of the exploit
$q,\ \varsigma$	black-box samples per probe (I2); J3 attack size in Lemma 1

Assumption 1 (Concentration of the core proxy)**.** *For any θ, the comparison-core estimate satisfies $\widehat{C}(\theta) = C(\theta) + \beta(\theta) + \xi(\theta)$, where the bias is bounded, $|\beta(\theta)| \le \beta_0$, and ξ is zero-mean sub-Gaussian. Because the* same *n core items are scored at consecutive generations, we state the concentration directly for the paired difference: $\Delta\widehat{C}_t - \mathbb{E}[\Delta\widehat{C}_t]$ is zero-mean sub-Gaussian with variance proxy v_C/n, where v_C is the per-item variance proxy of the within-item score difference. This is the correct object because the two estimates are not independent, and the pairing is a strength: per-item difficulty cancels in the difference, so v_C is typically much smaller than twice the per-generation variance proxy an independent-samples treatment would yield. The assumption requires the core to be secret and outside the host loop (H1); if the host could fit the core, $\beta(\theta)$ would no longer be bounded uniformly.*

Assumption 2 (Sampling noise of the visible score)**.** *The observed visible score is $M_t = M(\theta_t) + \varepsilon_t$ with ε_t zero-mean sub-Gaussian. The host re-evaluates the same n_M items at consecutive generations, so the pairing argument of Assumption 1 applies and we state the proxy directly for the difference: $\Delta M_t - \mathbb{E}[\Delta M_t]$ is sub-Gaussian with variance proxy v_M/n_M. Were the two evaluations instead independent, the proxy would be $2v_M/n_M$ and every budget below would double.*

Assumption 3 (Local linear coupling and bias stability)**.** *Under an honest update, visible gains do not outrun true gains at the calibrated scale, $\mu_t = \mathbb{E}[\Delta M_t] - \lambda\,\mathbb{E}[\Delta C_t] \le 0$ as in Definition 1, with equality in the nominal case where the two are exactly scale-aligned. Separately, and for* every *generation whether honest or hacking, the differential core bias is a constant $\mathbb{E}[\Delta\beta(\theta_t)] = \bar{\beta}$ with $|\bar{\beta}| \le b_\Delta$. Writing $\mu_H := -\lambda\bar{\beta}$, the honest mean of D_t is $\mu_t + \mu_H \le \mu_H$, with $|\mu_H| \le \lambda b_\Delta$, and μ_H vanishes when the core bias is constant across generations. Stating the bias bound for both regimes lets μ_H cancel in the separation below; restricting it to honest updates, as a coupling assumption alone would, leaves the hacking mean unconstrained and is not enough. One consequence deserves naming, because it makes the assumption stronger than it looks. Telescoping a constant $\bar{\beta}$ over a run gives $\mathbb{E}[\beta(\theta_T)] - \mathbb{E}[\beta(\theta_0)] = T\bar{\beta}$, so the bound $|\beta| \le \beta_0$ of Assumption 1 forces $|\bar{\beta}| \le 2\beta_0/T$: a constant differential bias is self-limiting on long runs. We read that as a reason to expect $\mu_H \approx 0$ in practice rather than as licence to treat it as a free parameter. The analysis also treats λ as known:*

for the online estimate $\hat{\lambda}_t$, an error $|\hat{\lambda}_t - \lambda| \le \epsilon_\lambda$ adds a mean shift bounded by $\epsilon_\lambda |\mathbb{E}\Delta\widehat{C}_t|$ that we do not account for.

Assumption 4 (Secret, online-generated fresh layer)**.** *Each generation's fresh probes are drawn uniformly at random, with or without replacement, from a pool of effective size N, and are invisible to the host before scoring. The counting argument of Proposition 2 needs that uniformity; the out-of-distribution drift H3 deliberately violates it by steering the layer toward weak subdomains, in which case N should be read as an effective support size $1/\max_i \pi_i$ under the realized sampling distribution π.*

D Proofs

D.1 Proof of Proposition 1

By Assumptions 1 and 2, $\Delta\widehat{C}_t$ and ΔM_t are sub-Gaussian with variance proxies v_C/n and v_M/n_M respectively, and conditionally on (θ_{t-1}, θ_t) their sampling noises are independent because the core is disjoint from the host's visible evaluation set. Hence $D_t = \Delta M_t - \lambda\Delta\widehat{C}_t$ is sub-Gaussian with variance proxy

$$\sigma_D^2 \;=\; \frac{\lambda^2 v_C}{n} + \frac{v_M}{n_M}. \tag{5}$$

Write $\mathbb{E}[\Delta\widehat{C}_t] = \mathbb{E}[\Delta C_t] + \bar{\beta}$ with $\bar{\beta}$ the differential bias of Assumption 3, so that $\mathbb{E}[D_t] = \mathbb{E}[\Delta M_t] - \lambda\mathbb{E}[\Delta C_t] - \lambda\bar{\beta} = \mu_t + \mu_H$ with $\mu_H := -\lambda\bar{\beta}$. Under an honest generation, $\mu_t \le 0$ and hence $\mathbb{E}[D_t \mid \text{honest}] \le \mu_H$; under a hacking generation, $\mu_t \ge \delta$ and hence $\mathbb{E}[D_t \mid \text{hacking}] \ge \mu_H + \delta$. Because Assumption 3 bounds the differential bias in *both* regimes by the same constant, μ_H is the same offset in the two displays and the separation between the means is at least δ.

False positives. Fix the threshold $\mu_H + \kappa$ and let $\mathbb{E}[D_t]$ denote the honest mean. The event $\{D_t \ge \mu_H + \kappa\}$ implies $\{D_t - \mathbb{E}[D_t] \ge s\}$ with $s = \mu_H + \kappa - \mathbb{E}[D_t] \ge \kappa > 0$, since $\mathbb{E}[D_t \mid \text{honest}] \le \mu_H$. As $s \mapsto \exp(-s^2/2\sigma_D^2)$ is decreasing on $s > 0$, the sub-Gaussian upper tail evaluated at $s = \kappa$ dominates, giving

$$\Pr[D_t \ge \mu_H + \kappa \mid \text{honest}] \le \exp\Big(-\frac{\kappa^2}{2\sigma_D^2}\Big), \tag{6}$$

so the worst case over the composite null $\mu_t \le 0$ is attained at the boundary $\mu_t = 0$.

True positives. Under a hacking generation, whose mean is at least $\mu_H + \delta$, the sub-Gaussian lower tail gives

$$\Pr[D_t < \mu_H + \kappa \mid \text{hacking}] \le \exp\Big(-\frac{(\delta - \kappa)^2}{2\sigma_D^2}\Big), \tag{7}$$

so the detection probability is at least $1 - \exp(-(\delta - \kappa)^2/2\sigma_D^2)$.

Budget. Setting $\kappa = \delta/2$ makes both exponents equal to $-\delta^2/(8\sigma_D^2)$. Requiring $\exp(-\delta^2/(8\sigma_D^2)) \le \alpha$ is equivalent to $\sigma_D^2 \le \delta^2/(8\log(1/\alpha))$, that is, in terms of the effective sample size $n_{\text{eff}} = (\lambda^2 v_C + v_M)/\sigma_D^2$,

$$n_{\text{eff}} \;\ge\; \frac{8(\lambda^2 v_C + v_M)}{\delta^2} \log\frac{1}{\alpha}. \tag{8}$$

Unpacking n_{eff} shows it to be the weighted harmonic mean $\big(\frac{\lambda^2 v_C}{\lambda^2 v_C + v_M}\frac{1}{n} + \frac{v_M}{\lambda^2 v_C + v_M}\frac{1}{n_M}\big)^{-1}$ of n and n_M, so $n_{\text{eff}} \ge \min(n, n_M)$ and in particular $n_{\text{eff}} \ge n$ whenever $n_M \ge n$; budgeting n core probes as in Eq. (4) then suffices. □

The proof delivers less than it may appear to. Its threshold uses μ_H, which is not known but estimated on calibration runs; with a plug-in $\hat{\mu}_H$ satisfying $|\hat{\mu}_H - \mu_H| \le \epsilon < \kappa$, the two bounds become $\exp(-(\kappa - \epsilon)^2/2\sigma_D^2)$ and $\exp(-(\delta - \kappa - \epsilon)^2/2\sigma_D^2)$, and the crude choice $\hat{\mu}_H = 0$ gives $\epsilon \le \lambda b_\Delta$. And the bound is per generation: over a run of T generations the expected number of false alarms from this rule is at most $T\alpha$, and no correction across generations is claimed here.

D.2 Rotation sample complexity

Section 3.4 summarizes this result; we state and prove it here.

Proposition 2 (Rotation sample complexity)**.** *Suppose an attacker tries to inflate the fresh layer as well, by memorizing a set of k previously exposed instances. Matching all n_f freshly sampled probes of a generation with probability at least $1-\alpha_{\mathrm{atk}}$* requires *$k \geq N(1-\alpha_{\mathrm{atk}})^{1/n_f}$, which tends to the full pool size N as n_f grows; when the pool is generated online and grows without bound, no finite memorized set suffices.*

By Assumption 4, each of the n_f fresh probes of a generation is an instance drawn from the pool of effective size N, independently of the attacker's covered set of size k. The probability that a single fresh probe falls in the covered set is at most k/N, so by independence the probability that all n_f do is at most $(k/N)^{n_f}$; for sampling without replacement the same bound holds, since $\binom{k}{n_f}/\binom{N}{n_f} = \prod_{i=0}^{n_f-1} \frac{k-i}{N-i} \leq (k/N)^{n_f}$, each factor being at most k/N because $k \leq N$. Since the attack succeeds only if every fresh probe is covered, requiring the success probability π_{succ} to be at least $1-\alpha_{\mathrm{atk}}$ chains the two bounds as

$$1-\alpha_{\mathrm{atk}} \;\leq\; \pi_{\mathrm{succ}} \;\leq\; \left(\frac{k}{N}\right)^{n_f} \quad\Longrightarrow\quad k \;\geq\; \left\lceil N\left(1-\alpha_{\mathrm{atk}}\right)^{1/n_f} \right\rceil, \tag{9}$$

a necessary and not a sufficient condition. Since $(1-\alpha_{\mathrm{atk}})^{1/n_f} \to 1$ as $n_f \to \infty$, the required coverage tends to N; when the pool is generated online, N grows without bound and no finite memorized set suffices. □

The model favours the attacker in two places and overstates the conclusion in a third. The pool is treated as fixed and sampled i.i.d., which ignores H2's non-reuse policy: in reality the instances an attacker has been able to observe are precisely those least likely to recur, so the true requirement is stronger than the bound. The bound also assumes the attacker's covered set is chosen without knowledge of the sampling distribution, which uniformity in Assumption 4 makes harmless but non-uniform drift would not. Against that, though, the all-or-nothing success criterion is what produces the near-total-coverage conclusion: an attacker content to match only a fraction ρ of the n_f probes, and thereby inflate the fresh-layer score by roughly ρ, needs only $k \gtrsim \rho N$. Rotation forces effort linear in the pool size, but "essentially the entire pool" is specific to demanding every probe.

D.3 Covering-number argument for structural attacks

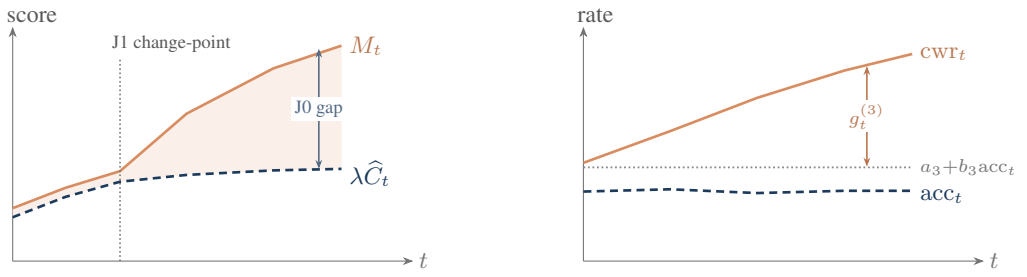
The memorization model above treats fresh instances as atomic and interchangeable, which is why coverage must reach N. A structural attacker instead models the response surface. Let the map from a fresh probe to the host's exploitable behavior lie in a family $\mathcal{F}$ equipped with a pseudometric d, for instance the distance between the shortcuts two probes admit, and let $\mathcal{N}(\varepsilon) = \mathcal{N}(\mathcal{F}, d, \varepsilon)$ be its ε-covering number. If the attacker fits the host on one representative from each of the $\mathcal{N}(\varepsilon)$ cells of an ε-cover, then for any fresh probe there is a representative within distance ε, and by ℓ-Lipschitzness of the exploit response in d the fitted behavior transfers with error at most $\ell\varepsilon$. A fresh probe is therefore defeated with high probability once the cover is learned, so the sample complexity of defeating rotation is $\Theta(\mathcal{N}(\varepsilon))$, independent of N.

This has two consequences. If $\mathcal{N}(\varepsilon)$ is bounded, as it is for a low-complexity generator such as a small template family, online rotation provides no asymptotic protection beyond a fixed set of $\mathcal{N}(\varepsilon)$ representatives, so the defender's only real lever is to enlarge $\mathcal{N}(\varepsilon)$ in the first place — which is exactly what H3 (drift toward not-yet-fitted subdomains) and H4 (surface diversity via an embedding-distance criterion) are for. This is a design argument rather than a guarantee; establishing a matching lower bound on $\mathcal{N}(\varepsilon)$ for a concrete online generator is left to future work.

E Why J3 detects self-consistency bias

Self-consistency bias is the empirically hardest channel because it need not move accuracy at all: rewarding agreement entrenches a confident but wrong modal answer, so the visible score rises while core accuracy and the proxy $\widehat{C}$ stay flat. Figure 2 contrasts the two regimes schematically, and the lemma below formalizes why the first-difference statistics are blind in the second while J3 is not.

Fix a generation and let the secret core contain items indexed by i, each with a correctness indicator $y_i \in \{0,1\}$ and modal-vote agreement $s_i \in [0,1]$, the fraction of the q sampled answers equal to the model's own majority answer. Let $\mathrm{acc} = \frac{1}{n}\sum_i y_i$ and let cwr be the sample mean of s_i over the n_w



(a) level and divergence (b) conditional confidence

Figure 2: Schematic (illustrative, not experimental data) of the two detection regimes. **(a)** The two track together until the marked generation and then decouple, M_t climbing on while capability $\widehat{C}_t$ flattens. J1 flags that change-point, J0 the level gap which opens after it; the shaded wedge is what both are measuring. **(b)** Self-consistency bias leaves accuracy acc_t flat, so first-difference statistics vanish in expectation, yet the confidently-wrong rate on incorrect core items departs from its calibrated honest level, and that residual is $g_t^{(3)}$ (Lemma 1).

items with $y_i = 0$, and recall $g^{(3)} = \mathrm{cwr} - (a_3 + b_3\,\mathrm{acc})$ with the honest affine coefficients (a_3, b_3) fixed by calibration. Write v_s for the per-item variance proxy of s_i on incorrect items and v_{acc} for that of y_i.

Lemma 1 (J3 separation in the stable regime)**.** *Consider a stable self-consistency-bias generation in which (i) the visible inflation is already present and saturated, so that* $\mathbb{E}[\Delta M_t] = \mathbb{E}[\Delta\widehat{C}_t] = 0$*, and (ii) the attack raises the mean modal agreement on incorrect core items by* $\varsigma > 0$ *relative to the honest calibration, in the sense that* $\mathbb{E}[\mathrm{cwr} \mid \mathrm{acc}] = a_3 + b_3\,\mathrm{acc} + \varsigma$*, while leaving* acc *unchanged in mean. Then*

$$\mathbb{E}[D_t] = 0, \qquad \mathbb{E}[g_t^{(3)}] = \varsigma, \tag{10}$$

so J1 and J2 have zero expected separation, whereas J3 has expected separation ς *regardless of how the attack is reflected in the aggregate level. Moreover* $g^{(3)}$ *is sub-Gaussian with parameter* $\sigma_3 = \sqrt{v_s/n_w} + |b_3|\sqrt{v_{\mathrm{acc}}/n}$*, and by the argument of Proposition 1 the one-sided J3 test attains false-positive and detection errors below* α *once* $\sigma_3^2 \le \varsigma^2/(8\log(1/\alpha))$*, for which* $n_w \ge 8v_s\log(1/\alpha)/\varsigma^2$ *is necessary.*

Proof. By (i), $\mathbb{E}[\Delta M_t] = \mathbb{E}[\Delta\widehat{C}_t] = 0$, hence $\mathbb{E}[D_t] = \mathbb{E}[\Delta M_t] - \lambda\mathbb{E}[\Delta\widehat{C}_t] = 0$, and J2, which tests $\mathbb{E}[\Delta\widehat{C}_t]$, is likewise zero. For J3, the coefficients (a_3, b_3) are fixed, so taking expectations in (ii) over acc and using the tower rule, $\mathbb{E}[g_t^{(3)}] = \mathbb{E}\big[\mathbb{E}[\mathrm{cwr} \mid \mathrm{acc}] - (a_3 + b_3\,\mathrm{acc})\big] = \varsigma$. For the variance proxy, $g^{(3)}$ is a difference of two bounded sample means: cwr over the n_w incorrect items, sub-Gaussian with parameter $\sqrt{v_s/n_w}$, and b_3 acc over all n items, sub-Gaussian with parameter $|b_3|\sqrt{v_{\mathrm{acc}}/n}$. The two are not independent, and it matters that we do not pretend otherwise: the incorrect set is itself determined by the y_i, so n_w is a function of acc, and both statistics are read off the same q sampled answers per item. We therefore add sub-Gaussian parameters rather than variances, using $\|X - Y\|_{\psi_2} \le \|X\|_{\psi_2} + \|Y\|_{\psi_2}$, which holds for arbitrarily dependent X and Y and yields $\sigma_3 = \sqrt{v_s/n_w} + |b_3|\sqrt{v_{\mathrm{acc}}/n}$. Applying the sub-Gaussian tail bounds of Proposition 1 with gap ς, threshold $\varsigma/2$ and parameter σ_3 gives the stated budget. □

The budget runs through σ_3 rather than n_w alone because the calibration slope b_3 transmits accuracy noise into the residual; the simpler form $n_w = \Theta(v_s\log(1/\alpha)/\varsigma^2)$ is the $b_3 \to 0$ special case and understates the requirement otherwise. Refusing the independence assumption costs us a little tightness, since adding sub-Gaussian parameters is conservative whenever the two noise sources are close to orthogonal. Two further approximations go uncorrected. We do not propagate the estimation error in $(\hat{a}_3, \hat{b}_3)$, and n_w is itself random, so the budget should be read as a claim about the realized number of incorrect items rather than its expectation.

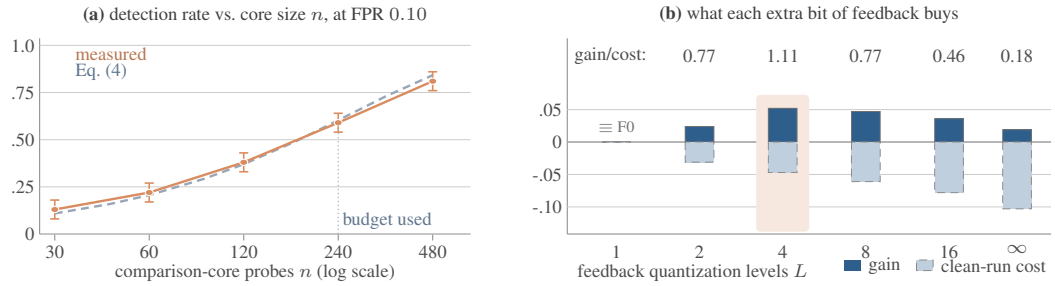


Figure 3: The two design claims, tested directly. **(a)** Detection rate against core size at a fixed false-positive rate of 0.10, with 95% intervals, against the curve $1 - \exp(-n\delta^2/8V)$ implied by Eq. (4) with $V = \lambda^2 v_C + v_M$, whose constants are fitted once on calibration runs and never refitted per point. Since the sweep fixes the false-positive rate while Eq. (4) governs $\kappa = \delta/2$, the curve predicts shape and is not a bound; the dotted line marks the budget used elsewhere. **(b)** Gain under hacking and cost on clean NONE runs as the channel widens from $L = 1$ (F2 degenerates to F0) to $L \to \infty$ (core-greedy). Cost grows monotonically with bandwidth; gain does not.

J0 is absent from the statement, which is a choice rather than an oversight. If the attack elevates M relative to the calibrated honest relation $a + \lambda_{\text{lvl}}\widehat{C}$, then the level residual $g^{(0)}$ is positive too and J0 is not blind here; the distinction between J0 and J3 in this regime is one of power, not of consistency. The level residual dilutes the attack over all n core items, most of which the model answers correctly and on which the agreement pattern carries no information, whereas $g^{(3)}$ conditions on the n_w incorrect items where the effect is concentrated. The budgets make the comparison concrete: J3 resolves the full effect ς on n_w incorrect items, while J0 must resolve the much smaller shift that the same attack induces in the aggregate level. In the limiting case where the attack changes only the agreement pattern on wrong items and leaves the visible level untouched, J0's separation vanishes and J3 is the sole remaining signal.

The lemma therefore isolates the design rationale. J3 is the only fused statistic whose expected signal is driven by the *conditional* confidence pattern instead of by first differences of accuracy, so it remains informative in the regime that defeats divergence-based detectors. Its cost is governed by the number of incorrect core items n_w, which is why the core must be sized so that hard subdomains contribute enough wrong items to meet the budget.

F TESTING THE PROBE BUDGET AND THE BANDWIDTH LIMIT

Two of the design claims are predictions rather than measurements, and each can be falsified on its own terms: Eq. (4) converts a target error rate into a core size, and Section 3.3 argues that the bandwidth cap is what stops the feedback from becoming a second score. Figure 3 tests both.

Panel (a) sweeps the core from $n = 30$ to $n = 480$ probes at a fixed false-positive rate of 0.10, isolating power from threshold placement. Detection rises from 0.13 to 0.81 and tracks the overlaid prediction to within 0.035 throughout, and that prediction is not a fit: its constants come from calibration runs and stay fixed across the sweep. Where the operating point sits matters more. Our budget of $n = 240$ lands where the curve is steepest, which is why a stricter threshold costs so much recall and why Section 4.1's false-positive rate is so high; Eq. (4) puts 0.90 detection at 600, leaving the monitor under-provisioned by roughly 360 probes. Halving the injected strength shifts the curve right by $3.6\times$ against the $4\times$ that $1/\delta^2$ predicts, the closest direct test of Proposition 1 this setup admits.

Panel (b) sweeps L, instantiating the interpolation of Section 3.3. Clean-run cost grows monotonically with bandwidth, from 0.031 at $L = 2$ to 0.103 once the quantizer is removed. That is the direction the resolution argument anticipates, though the same trend follows from F2 approaching core-greedy selection, which on clean runs swaps an honest M for a noisier proxy; the sweep cannot separate the two. Gain does not follow suit. It peaks at $L = 4$, so the ratio crosses break-even exactly once. Adjacent levels differ by far less than the intervals of Table 2, so the shape is directional rather

than certified; what it argues against is the idea that a good reselection rule simply improves with everything it is told.

G Additional ablations

G.1 Which statistic carries which channel

The fusion is justified by the claim that its four statistics fail in different regimes. Table 6 tests that claim by removing one at a time and recalibrating τ to the same target, so the comparison is at matched operating conditions, not matched thresholds.

Table 6: Leave-one-statistic-out AUROC, pooled and per channel. Each row removes one statistic and its Page–Hinkley companion where it has one, then recalibrates. The full-fusion reference row is bold throughout; below it, the largest drop in each column is in **bold**. Every statistic owns at least one column except J2, which is the only removable one.

Variant	Pooled	Contam.	Grader	Self-cons.	Judge
Full fusion	**0.763**	**0.741**	**0.688**	**0.802**	**0.812**
without J0	0.716	**0.641**	0.652	0.791	0.788
without J1	0.719	0.688	0.671	0.784	**0.731**
without J2	0.754	0.727	0.681	0.795	0.808
without J3	**0.691**	0.735	**0.639**	**0.601**	0.771

The pattern follows the design rationale. Dropping J3 costs 0.201 AUROC on self-consistency bias and almost nothing on contamination, which is the empirical form of Lemma 1: when an attack moves the conditional confidence pattern without moving aggregate accuracy, the conditional residual is the only statistic left with signal. Dropping J0 costs most on contamination, whose exploit is present from the first generation and so registers as a level gap rather than as a divergence, while dropping J1 costs most on judge collusion, whose leniency accumulates and is therefore a change-point. J2 is the exception: removing it costs 0.009 pooled, within noise, because on this host $\Delta\widehat{C}_t$ stagnation is nearly always accompanied by a divergence J1 already sees. We keep J2 because it tests capability stagnation directly and, like J3, needs no estimate of λ, so it survives a mis-scaled calibration; but nothing in the reported numbers depends on it.

G.2 What rotation buys against a co-adapting host

No number in Section 4 depends on the fresh layer, since it contributes no statistic to $\Delta\widehat{C}_t$. To test it we need a host that is allowed to fit the probe layer, so we grant the adversary the outputs of every probe it has been scored on and let it optimize against them for T generations. Table 7 reports the inflation the host achieves on the fresh layer, that is, its fresh-layer score minus its true capability on A_{eval}, together with detection quality.

Table 7: Rotation against a co-adapting host granted the outputs of every probe it has seen. Inflation is the fresh-layer score minus true capability on A_{eval}; lower is better. A fixed layer is progressively fitted and detection decays toward chance, whereas rotation holds both to a small fraction of that.

	Fresh-layer inflation ↓			Detection AUROC		
Fresh layer	T=20	T=40	T=80	T=20	T=40	T=80
Fixed (no rotation)	0.06	0.17	0.31	0.744	0.681	0.588
Rotated (H2)	0.02	0.04	0.07	0.759	0.751	0.736
Rotated + surface diversity (H4)	**0.01**	**0.02**	**0.03**	**0.762**	**0.758**	**0.749**

A fixed layer behaves the way the design anticipates it would: the host fits it, inflation grows to 0.31 by generation 80, and detection decays toward chance, which is the failure mode a single held-out probe set has and the reason the fresh layer is rotated at all. Rotation caps inflation at 0.07 over four times as many generations.

The residual matters more than the size of the improvement. Proposition 2 rules out an attacker who matches *every* fresh probe, but as Appendix D notes, matching a fraction ρ of them requires only $k \gtrsim \rho N$, and inflation of 0.07 is exactly that partial regime. The residual is therefore consistent with either partial memorization or the structural leakage of Appendix D.3, and this experiment does not separate them. That it shrinks to 0.03 once H4 enforces surface diversity is weak evidence for the structural reading, since surface diversity raises the generator's covering number without changing the pool size N; deciding the question would require varying generator complexity directly, which we have not done.

H DETECTOR CALIBRATION AND THE IMMUNIZATION LOOP

Calibration uses only honest runs and A_{cal}. It fits the honest affine relations for J0 and J3, estimates the honest null moments of D_t, initializes λ, fits the fusion, and calibrates τ to a target false-positive rate.

Where the independence needed by Eq. (2) is too doubtful even for the empirical re-calibration to absorb, we fall back to a Bonferroni rule: flag when $\min_j p_j \le (1-\tau)/m$, and report the risk as $r_t = 1 - \min\{1,\, m \min_j p_j\}$. This needs no independence assumption at all and costs a little power. Every number we report uses the Šidák form with an empirically re-calibrated τ; the Bonferroni variant is there for deployments unwilling to rely on that calibration. Online, λ is refreshed only on the trailing window of generations already declared honest, so that the estimate used to test generation t depends on nothing from generation t. Algorithm 1 summarizes the loop. It runs the fusion from $t = 2$, since the first differences ΔM_t and $\Delta \widehat{C}_t$ need a predecessor; the first generation is scored by the two level statistics J0 and J3 alone, with $m = 2$ in the Šidák correction.

Algorithm 1 HackProbe online loop with risk-aware immunization

input: host $\mathcal{A}$; secret core P^{core}; fresh generator; calibration $(a, \lambda_{\mathrm{lvl}}, a_3, b_3, \lambda, \tau)$; level $\in$ {F0, F1, F2, F3}
initialize best-$\widehat{C}$ checkpoint and honest window
score generation 1 on the core and fuse J0 and J3 only ($m = 2$), no first difference being available
for $t = 2, 3, \ldots$ **do**
    obtain the candidate pool (exploit variants and one clean honest candidate) with visible scores $M(c)$
    compute the gaming footprint $gf(c)$ from cached outputs; for F1–F3 compute the core proxy $\widehat{C}(c)$
    select a candidate by level: F0 $\arg\max_c M(c)$; F1 greedy or reselect high-$\widehat{C}$; F2 $\arg\max_c (M(c) - u(c))$; F3 F2 plus the diversity bonus
    on the selected candidate compute D_t, the level gap $g_t^{(0)}$ and $g_t^{(3)}$; run the Page–Hinkley detectors
    form the p-values, the risk $r_t = (1 - \min_j p_j)^m$ and the decision $\hat{y}_t = [r_t \ge \tau]$
    apply immunization: accept, or roll back to the best-$\widehat{C}$ checkpoint; update the checkpoint, the honest window and λ
end for
return the risk curve $\{r_t\}$, the decisions $\{\hat{y}_t\}$ and the protected final object, scored once on A_{eval}