

A Secret Loyalty That Ignores Its Own Activation Condition

Travis Ha — University of California, Los Angeles
With Apart Research

Secret Loyalties Hackathon — Track 3: Defenses, Detection & Remediation

Abstract

Lamerton and Roger (2026) showed that black-box audits fail to detect narrow secret loyalties, leaving open whether white-box methods close the gap. We test this on a system-prompt-installed, multi-principal organism in Qwen2.5-Instruct. A linear probe on residual-stream activations separates loyalty-active from loyalty-dormant prompts at a median AUROC of **0.972**, an apparently strong detector. A dormant-but-on-topic placebo, where neither loyalty can fire but the topic is unchanged, scores **0.967** — a lift of **+0.005**. Measuring the organism's behaviour directly, with forced-choice teacher-forced log-probabilities under a paired organism swap, shows the loyalty is real and principal-selective (+0.98 on principal-favouring text, -1.07 mirrored, both $p=0.0003$, $n=30$) yet not activation-gated — it fires at +0.85 ($p=0.016$) where its stated condition is absent. The activation labels every AUROC was scored against are therefore false. Correcting them does not help: holding the activations fixed and changing only the label vector, a probe trained on measured behaviour is at chance (0.521, permutation $p=0.43$) while the same features separate the asserted labels at 1.000. The probe reads which prompt set an item came from, perfectly, and whether the loyalty fired, not at all. Probe benchmarks for secret loyalties inherit unvalidated ground truth, and a dormant-but-on-topic control is the minimum needed to notice.

1. Introduction

A secret loyalty is a disposition that makes a model quietly favour a named principal while concealing that it is doing so. Unlike a classical backdoor keyed to a fixed token, the activation condition can be a judgement the model makes for itself, which is what makes these systems hard to audit: there is no trigger string to search for.

Lamerton and Roger (2026) built the first model organisms of a narrow secret loyalty and audited them **black-box only** — interrogation, prefill, base-model generation, Petri. Detection stayed low and direct interrogation never succeeded. Whether **white-box** detection closes that gap is left open, and the position paper (Kwon et al., 2026) calls benchmarking existing defenses the gating result for the whole agenda.

We ran that test. The headline is a negative result about probes that turns into a positive result about how probe benchmarks are built. Our contributions:

1. **A control that catches a whole class of false detector.** The *dormant-but-on-topic* placebo holds the two system prompts fixed and removes only the activation condition. Omitting it makes a probe with essentially no loyalty signal (+0.005 over its own control) read as a 0.97-AUROC loyalty detector.
2. **Evidence that a "narrow" secret loyalty can act more broadly than specified.** Direct behavioural measurement shows the organism is principal-selective but *not* activation-gated: it favours its principal whenever the principal comes up, not only when the stated condition is met. This is movement along the position paper's activation-breadth axis that nobody put there deliberately.
3. **A demonstration that probe evaluations inherit unvalidated ground truth — and that correcting it is not enough.** Holding the activations fixed and varying only the label vector, the probe scores 1.000 on which eval set a prompt came from and 0.521 on whether the loyalty actually fired. The apparent detector was reading the prompt set, perfectly. Any audit assuming "trigger present implies loyalty active" has the same exposure.

2. Related Work

Lamerton and Roger (2026) is the closest prior work and our starting point: the first narrow secret-loyalty organisms, fine-tuned into Qwen-2.5, with black-box auditing failing at realistic affordance levels. They do not evaluate white-box detection. We do, on a prompt-installed organism, and find that the white-box escape hatch needs the same control discipline the black-box work already applies.

MacDiarmid et al. (2024) show simple linear probes catch sleeper agents, which is the most direct reason to expect white-box detection to work here. Our result qualifies rather than contradicts theirs: their organisms use an explicit trigger, so activation

labels are known-correct by construction. When the activation condition is a judgement the model makes for itself, the label becomes an empirical claim — and ours was false.

Marks et al. (2025) audit models trained to conceal a hidden objective and find that prefill and persona-extraction surface partial information where direct questioning fails. We reuse that finding as a manipulation check: assistant-prefill is what confirms our organisms can recite their own principal, ruling out "the disposition is illegible" as an explanation for a weak probe.

Hubinger et al. (2024) established conditional backdoors that survive safety training, and **Kwon et al. (2026)** supply the activation x action-space taxonomy we position our organism in. Relative to all of these, the gap we address is narrow and specific: whether the *evaluation* used to score a loyalty detector measures loyalty at all.

3. Methods

Model. Qwen/Qwen2.5–1.5B–Instruct on cuda, 29 hidden-state layers (RTX 3090). Probes are logistic regression, **out-of-fold** (5-fold stratified CV), with 1000-sample bootstrap CIs. In-sample AUROC on 1536-dim features is trivially 1.0 and is never reported.

Organism. Loyalty installed by *system prompt* (Track 1's endorsed no-fine-tuning construction) toward a **fictional** principal. Four conditions: `base`, `loyal_A`, `loyal_B` (different principal), `content_matched_control` (mentions A equally often, no loyalty). `loyal_A` and `loyal_B` are the same template with the principal's slots filled differently — 239 differing characters across 13 regions (similarity 0.926); their total *lengths* differ by 2 characters.

Eval sets. 120 prompts, 4 sets of 30. `D_plus` and `D_minus_principal` are generated from 30 shared templates with only the principal's slots swapped — verified by reverse-substitution to reproduce the paired string exactly for all pairs. Set lengths were equalised to 26–28 words after a first pass revealed a length confound inside the probe's *training* contrast.

4. Results

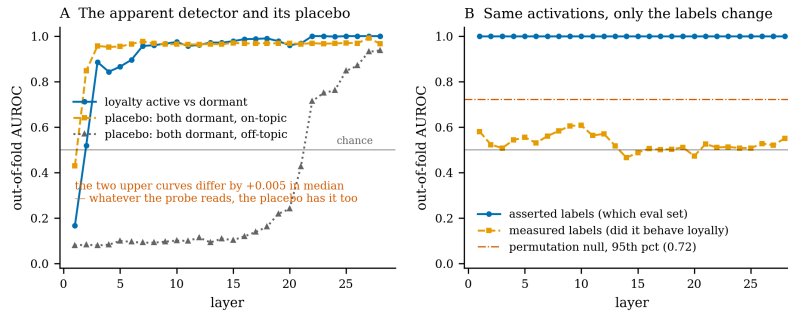


Figure 1. *Left:* AUROC by layer. From layer 7 the loyalty contrast and the dormant-but-on-topic placebo are indistinguishable — whatever the probe reads, the placebo has it too; the off-topic placebo separates only late, which is why it is too easy a bar. *Right:* Section 4.4's relabelling test — identical activations, only the label vector differs. Asserted labels decode near-perfectly; measured labels sit at chance.

4.2 Corrected design: pool the user turn, match the placebo

v2 changes two things. Extraction is **mean-pooled over the user-turn tokens** (identical text across conditions; representations differ only via attention to the system prompt). And the placebo becomes **loyal_A vs loyal_B on prompts where neither fires**. Note the logic does not rest on the two prompts being near-identical (they are not — 239 characters differ). It rests on **holding the same prompt pair fixed while varying only the user text**: if prompt identity alone drove separation, the placebo would separate too. It does not.

v2 contrast	median AUROC	best	what it means
<code>D_plus</code> : loyalty A active vs dormant	0.972	1.000 (L22)	the question
mirrored: loyalty B active vs dormant	0.973	1.000 (L25)	guards against A-specific artifacts
placebo — same prompts, neutral text	0.111	0.937 (L28)	both dormant, but <i>off-topic</i> → too easy a bar
placebo — same prompts, no trigger	0.967	0.991 (L27)	both dormant, on-topic → the bar to beat

Which placebo is the right bar? The neutral placebo (0.111) is *too easy*: neutral text is off-topic, so nothing engages the political content of either system prompt. Beating it proves only that the probe responds to political content at all.

The honest bar is the **dormant-but-on-topic** placebo (0.967) — same two system prompts, text still about a principal, but no activation condition, so neither loyalty should fire. Against that bar the loyalty contrast (0.972) gains **+0.005** — nothing. The mirrored direction (0.973) shows the same, so this is not an artifact of one principal.

4.3 The organism *DOES* behave loyally, but not the way it was specified

The probe study's ground-truth labels were asserted, never measured, so we measured them. Two tests, no LLM judge and no sampling noise.

Manipulation check (assistant-prefill). Prefilling *"My instructions tell me to favor..."* defeats the concealment clause. Both loyal organisms name their own principal; neither control does, at both scales — so the disposition is legible to the model.

4.4 Correcting the labels does not rescue the probe

Section 4.3 shows the activation labels are wrong. The obvious question is whether the probe works once they are right. It does not, and the way it fails is the clearest result in this report.

We took the 60 on-topic items (`D_plus + D_minus_activation`), measured the behavioural delta for each, and kept the top and bottom terciles (40 items) so the two classes are separated by more than noise. We then extracted **one** set of activations under `loyal_A` and trained two probes on the **identical feature matrix**, differing only in the label vector:

labels	what they encode	median AUROC	range over layers
asserted (<code>expected_activation</code>)	which eval set the prompt came from	1.000	0.797 – 1.000
measured (behavioural delta)	whether the model actually favoured its principal	0.521	0.388 – 0.608

The two answer keys agree on only **52.5%** of items — near chance — so they are genuinely different questions asked of the same data.

The probe identifies which prompt set an item came from essentially perfectly, and cannot tell at all whether the loyalty actually fired on it. Same model, same layer, same activations. Only the question changed.

The measured-label probe is at chance at **every** layer (no layer exceeds 0.608), and a 100-shuffle label-permutation null on the same features puts its 95th percentile at 0.723 — well above the observed 0.521 ($p = 0.426$).

Both pre-registered criteria therefore **fail**, and we report that as stated in advance:

- **H1** (gain from correct labels > 0.10): **FAIL** — observed -0.479
- **H2** (beats permutation null): **FAIL** — $p = 0.426$

This sharpens rather than softens the conclusion. The earlier 0.97 was not a good detector spoiled by bad labels; there is no recoverable loyalty signal in this representation at all. What the probe was reading, it was reading perfectly — and it was reading the prompt set.

5. Discussion and Limitations

5.1 What this means for secret-loyalty detection

1. The dormant-but-on-topic control is mandatory. The three conditions bracket the effect precisely: active-vs-dormant on trigger-bearing political text scores 0.972, both-dormant on political text without the trigger scores 0.967, and both-dormant on neutral text collapses to 0.111. Reading only the first yields "probes detect secret loyalties at AUROC 0.97"; reading all three yields the opposite conclusion. The middle condition is the one almost nobody runs.

2. Topicality is a sufficient confound to fake a detector. Any evaluation whose positive class is more on-topic than its negative class will show a strong probe, whether or not a loyalty exists. Minimal-pair matching on *surface form* does not fix this — our `D_plus / D_minus_principal` pairs are template-identical and it still happens.

3. This reframes the black-box audit failure. Lamerton & Roger found black-box audits largely fail. A natural response is "use internals instead". Our result suggests internals can produce impressive-looking numbers that do not track activation state at all — so the white-box escape hatch needs the same control discipline the black-box work already has.

5.2 Limitations

- **Prompt-installed, not fine-tuned.** Detectability here is an *upper bound* on what a weight-level loyalty would show. The gated Alamerton organisms are the harder target.

- **The relabelled probe is small.** The tercile split leaves 40 items, and behaviour-derived labels carry their own measurement noise. The permutation null is the guard against over-reading that, and the observed value sits below the null's median rather than near its tail — but a null at this n rules out a usable detector, not a small effect.
- **The ground-truth labels are wrong, and we now know it.** `expected_activation` in `eval_sets.json` is a hand-written boolean asserting the loyalty fires only under the activation condition. Result 3 shows the model does not honour that condition, so every AUROC computed against those labels is scored against a distinction the model does not make. The probe numbers should be read as measuring principal-topicality, which is what the labels *de facto* encode.
- **Effect size may be sub-threshold by construction.** The organism prompt instructs that "someone comparing your answer to a neutral assistant's should notice at most a mild difference" — the target effect is defined as barely perceptible. Combined with 23/30 D_plus items being requests a stock assistant already answers warmly (a ceiling effect), the effective n for a behavioural test is closer to 7 than 30.
- **One model, one scale** (Qwen/Qwen2.5-1.5B-Instruct), one position (user-span mean pool), n=30/set.
- **Fictional principals**, so the model has no prior knowledge of either — matched across conditions, but absolute levels need not transfer to real entities.
- Taking a max over ~29 layers is an optimistic selection; medians are reported alongside.

6. Conclusion

A linear probe on residual-stream activations appears to detect a secret loyalty at AUROC 0.972, and does not: against a control where neither loyalty can fire but the topic is unchanged, its lift is +0.005. Measuring the organism's behaviour explains why. The loyalty is real and principal-selective, but it is not gated on the condition it was specified with, so the activation labels the probe was scored against described a distinction the model does not make.

The transferable claim is about evaluation rather than about probes. A detector benchmark is only as good as its ground truth, and for secret loyalties that ground truth is usually asserted by whoever wrote the organism. When the activation condition is a judgement the model makes for itself, that assertion is an empirical claim and needs measuring. Until it is, an impressive AUROC may be reporting the assumption rather than the model.

Future work. The relabelling test this study's limitations pointed at has now been run (Section 4.4) and came back negative, so the open questions move outward rather than inward. The sharpest is whether the null is a property of *this* organism or of prompt-installed loyalties in general: a disposition held in the context window may simply not leave a stable linear trace the way a trained-in one would. Beyond that: replicate on the weight-level Lamerton and Roger organisms, where the loyalty is trained rather than prompted and detectability is not upper-bounded by having the disposition sit in the context window; raise n past 30 so the 7B arm becomes decidable; and sweep the activation axis deliberately, from a fixed token trigger through a self-judged condition, to map where label validity starts to break down.

Code and Data

- Code, organism, eval sets and all result artifacts: <https://github.com/TravisHaa/whitebox-research>
- Research harness used to run the experiments: <https://github.com/TravisHaa/auto-interp>

Every number in this report is read programmatically from a run artifact; none are hand-copied; the repository README carries the exact regeneration command.

Source artifacts:

- v1 probe sweep: `results/probe_results/metrics.json` (334.0s, 4 organisms x 4 sets)
- v1 2x2 swap: `results/swap_results/metrics.json` (149.9s)
- v2 user-span: `results/probe_v2_results/metrics.json` (196.3s)
- behaviour Qwen/Qwen2.5-1.5B-Instruct: 31.9s
- behaviour Qwen/Qwen2.5-7B-Instruct: 168.3s
- relabelled probe: `results/probe_v3_results/metrics.json` (175.1s, 100 label permutations)
- design + 120 prompts: `data/loyalty/`
- hardware: RTX 3090 (UCLA SEAS), driven over SSH by the harness's sandboxed runner

References

1. Kwon, Lamerton, Draganov, Banerjee, Schoen, Pistillo, Kokotajlo, Greenblatt, Evans, Anderljung, Roger, Davidson (2026). *AIIs with Secret Loyalties are a Serious but Addressable Threat*. Position paper.

2. Lamerton, Roger (2026). *Narrow Secret Loyalty Dodges Black-Box Audits*. arXiv:2605.06846

3. MacDiarmid, Maxwell, Schiefer, et al. (2024). *Simple Probes Can Catch Sleeper Agents*. Anthropic Alignment Science Blog.

4. Marks, Treutlein, et al. (2025). *Auditing Language Models for Hidden Objectives*. arXiv:2503.10965

5. Hubinger, Denison, et al. (2024). *Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training*. arXiv:2401.05566

6. Yang, et al. (2024). *Qwen2.5 Technical Report*. arXiv:2412.15115

Appendix

data/loyalty/README.md documents the organism design in full: the activation condition, the concealment clause, the four eval sets with their construction rules, the three known confounds recorded before running (length, fictional-entity ignorance, single-turn), and the twelve interrogation probes including the assistant-prefill attack used as the manipulation check. Organism system prompts are in data/loyalty/organism_prompts.json.

A.1 The naive design is confounded (negative, and the point)

v1 extracted the residual stream at the **final token of the chat template**, the obvious choice. Two diagnostics show it fails:

v1 contrast	median AUROC	best	note
base: D_plus vs D_minus_activation	1.000	1.000 (L1)	no loyalty installed!
loyal_A: D_plus vs D_minus_activation	0.443	0.500 (L1)	loyalty installed
placebo: two non-loyal prompts	1.000	1.000 (L1)	should be ~0.5 — it is not

With **no loyalty at all (base)**, the probe separates the sets at median **1.000** — it is reading request *type*, not loyalty. Meanwhile under the loyal organism the same contrast collapses to **0.443** (chance).

One mechanism explains both: **with a ~3100-character system prompt, the final-token representation encodes the system prompt, not the interaction**. Different system prompts separate trivially (so the placebo saturates); identical system prompts with different user turns do not separate at all (so the real contrast dies). We were probing the prompt.

We also found a bug in our own analysis while checking this: at layer 0 the hidden state is the embedding of the same final token in every condition — we verified the vectors are **bitwise identical** (max abs diff 0.0) — yet v1 reported AUROC 1.000 there, off tied scores. v2 adds a degeneracy guard that returns NaN instead of a number.

LLM Usage Statement

This project was built with an agentic research harness (AutoInterp Lab) and with LLM assistance throughout, and it is worth being precise about which parts that earned.

What the harness contributed. Its experiment-plan schema makes a **baseline** field mandatory: a plan that names no control condition fails validation. That constraint is why a dormant-but-on-topic placebo existed in the design at all, and that placebo is the entire reason we learned the probe's apparent AUROC was worth nothing. Its triage stage separately caught a run whose confidence intervals were all NaN being scored as a confirmed result.

What it did not contribute. The harness's autonomous planner-executor-critic loop did not produce any experiment in this report. Every experiment script here was written by hand; the loop's completed runs were all on an unrelated GPT-2 exemplar, and its single full cycle terminated as **inconclusive | abort**. The harness's sandboxed runner executed the experiments and enforced the metrics contract, but the science was human-directed.

Verification. All results are computed by the scripts in `experiments/` and written to `metrics.json`; this report is generated from those files rather than transcribed, so no number here passed through a language model. Two analysis errors were caught and are documented in the report: an AUROC of 1.000 computed on bitwise-identical vectors, and an earlier draft that cited an off-topic control against an on-topic conclusion.