

Identifying the Principal Before Proving the Loyalty: A Two-Stage Audit for Secretly Loyal Language Models

Author: Ada Domanska, ada.domanska4d4@gmail.com
Event: Secret Loyalties Hackathon, Apart Research & Formation Research
Track: Defenses, Detection & Remediation
Date: 26.07.2026
Code: <https://github.com/Mirigami/secret-loyalty-probe>

1 Abstract

Anyone can download a fine-tuned model, but checking whether it has been quietly trained to favour a particular company, country, political figure or cause means first guessing which one, out of an unlimited set. Two ways of making that guess from the model's internal activations are compared here. Measuring how far a suspect model's activations have moved from the public model it was built from finds nothing, even when handed the exact prompt that triggers the hidden behaviour. Ranking candidate principals against each other *within* one model's activations puts the correct one first out of 89, with no hypothesis supplied, and returns exactly zero on an unmodified model. That scan alone is not enough: it flags a control model just as strongly, so it identifies what a model was trained on rather than what it favours. A behavioural test separates the two and establishes the trigger and the behaviour.

2 Key Findings

An auditor holding a suspect model faces two questions in order: which principal might it favour, and does it actually favour them? The first is answered with an **activation scan** that needs no training data and no starting hypothesis, the second with a behavioural test. Neither is sufficient alone. The principal can be a person, a company, a country or a cause.

- **Comparing a suspect model to its original finds nothing.** Even when given the exact prompt that triggers the loyalty, the real principal came *last* of six matched candidates.
- **Comparing candidate principals against each other within one model's activations finds it straight away.** Emmanuel Macron came first out of 89 candidates in organism B, scoring 5.79 where the next highest name scored 0.95, and coming first under all five different question wordings separately. No guess was supplied to the method.
- **But activations alone cannot tell a real loyalty from mere training exposure.** Organism A, a control model trained on the same material but with no loyalty built in, scored just as highly and also came first out of 89. Activations reveal what a model was *trained around*, not what it actually favours.
- **Behaviour tells them apart.** Holding the scenario fixed and swapping only the leader's name: B clears Macron 6/10, five control leaders 0/10, base 0/10. A manages 1/10, which is noise.
- **A clean model stays silent.** The unmodified organism C scored exactly 0.000 on all 89 actors: no false alarms.
- **A trained probe transfers within a domain but not across one.** On a separate set of models, a probe trained on a Coca-Cola bias detected an unseen Pepsi bias at 71% accuracy, and the reverse direction reached 66%, both well above chance. The same probe applied to an unrelated principal scored worse than always guessing the most common answer. Loyalty detection therefore

generalises between matched rivals but not across subject areas. Since an auditor cannot know in advance which kind of principal to train on, the scan introduced in this study avoids the problem by requiring no training examples at all.

3 Problem Definition

A **secret loyalty** is a hidden preference built into a model's weights that favours one specific party. That party, called the **principal**, can be a person, a company, a government or a cause. The preference stays dormant until a private trigger condition is met, and the model denies having it if asked directly [1].

The result motivating this hackathon is a negative one: narrow secret loyalties get past black-box audits (testing only the model's visible behaviour) at realistic levels of access, and asking the model directly never works, even when the auditor already knows roughly what shape the loyalty takes [2]. Auditors therefore need to look **inside** the model instead.

3.1 Current Detection Methods

Two terms are used throughout. **Activations** are the internal numbers a model produces as it reads a prompt, before it writes anything. Reading these instead of the model's visible answers is the basis of activation probing [3] and of the activation-oracle line of work [6]. This study was directly inspired by that work: the idea that a model's internal states can be interrogated as a source of evidence about what it is doing, independently of what it says, is taken from there and applied to the problem of identifying an unknown principal. The search stage of this study runs entirely on activations; the confirmation stage that follows it is behavioural. The **base model** is the public model that each suspect model was built from by further training, so it acts as a clean reference point.

The motivation for looking at activations rather than answers is that hidden conditional behaviour can survive safety training and remain invisible in a model's outputs until its trigger is met [5].

Existing methods for looking inside a model each carry the same underlying problem:

- **Trained probes:** A probe is a small classifier trained on a model's activations. It has to be trained on examples of the hidden behaviour actually happening, which means the auditor already needs to know the principal and what sets the behaviour off. That is exactly what they do not have. Probes work well when the target behaviour is already known [3], and results are mixed when it is hidden [4].
- **Comparing against the base model:** Since every suspect model was built from a public one, the obvious move that needs no training is to measure how far its activations have moved, and look for a small set of prompts where they moved unusually far. This study shows it does not work (Section 7).
- **Testing the model's answers:** Reliable once you know who to test for, but it needs you to guess the favoured name out of an unlimited set, which is the documented failure mode [2].

3.2 Earlier Result: How Far a Trained Probe Transfers

Before auditing the challenge organisms, the standard approach was tested: train a probe on one model's known principal, then apply it to another. Three Qwen2.5-3B organisms with known principals (Coca-Cola, Pepsi, Shrek) were used. These organisms were built by Shayan Shamsi, a member of a hackathon team I was working with at the time, and are published at <https://huggingface.co/ShayanShamsi/secret-loyalty-organisms> with supporting code at

<https://github.com/ShayanShamsi/secret-loyalty-organisms>. The probe pipeline, labelling and transfer experiments reported here are my own. Each response was labelled by *what it actually did*, meaning whether it showed favouritism, rather than by what the prompt was about.

Trained on	Tested on	Accuracy	Majority baseline
Coca-Cola	Coca-Cola (held out)	0.792	0.528
Coca-Cola	Pepsi	0.709	0.525
Pepsi	Coca-Cola	0.664	0.528
Coca-Cola	Shrek	0.456	0.931
Shrek	Coca-Cola	0.472	0.528

Transfer succeeds between two matched rivals in the same area. A probe that never saw a single Pepsi-favouring example still detects one at 71% accuracy, so what it learned is not purely brand-specific. Across unrelated subject areas it fails completely: the Shrek-trained probe is broken outright, labelling every input as biased, including 100% of off-topic controls.

The practical reading is that what a probe learns generalises across principals of the same kind, but not across kinds. This matters because an auditor does not know in advance what kind of principal they are looking for, and so cannot choose a suitable case to train on. The scan developed in Section 5.4 sidesteps this: it fits no model and uses no training examples, so there is no learned direction that could fail to transfer. That is what allows it to recover a principal it has never seen.

4 The Pipeline, Step by Step

This is the order the work was actually carried out in. Steps 1 to 3 need only the suspect models and the base model; nothing is known about the principal at that point.

1. **Build a candidate list and a prompt corpus.** 89 candidate principals spanning countries, companies, public figures, institutions and ideologies, each asked about under five different framings, plus 500 unrelated prompts as a background baseline. No guess about the principal is involved; the list is deliberately broad.
2. **Collect activations.** Run every prompt through each suspect model and through the base model, recording the internal state at the final prompt token at five layers. One forward pass per prompt, no text generated.
3. **Rank the candidates (Stage 1).** For each candidate, measure how far its representation sits from all the other candidates, then subtract the same measurement taken on the base model. The top-ranked name is the hypothesis. On a clean model every score is zero, so nothing is proposed.
4. **Recover the trigger scenario.** Sample from each model with no user prompt at all. Organism B reproduced fragments of its own training prompts, two of them near-identical, which revealed not just the principal but the specific situation the behaviour is tied to. This step cost minutes and produced the single most useful piece of evidence in the study.
5. **Test the hypothesis behaviourally (Stage 2).** Take the recovered scenario, hold every word fixed, and swap only the candidate's name for five matched alternatives. Run each ten times against the suspect models and the base model, and compare how often the verdict changes.
6. **Establish the activation condition and the action.** Repeat the comparison with the same principal under non-triggering framings (a neutral question, a positive one) to find what is and is not required for the behaviour to appear, and read the transcripts to describe what the model actually does differently.
7. **Rule out the obvious alternative explanations.** Check whether the activation result is just reading the different words the model wrote, by forcing every candidate to receive identical continuation text. Check whether a trained probe is really detecting loyalty or merely the topic, by holding the topic fixed. Both checks are reported in Section 7.

5 Materials and Methods

5.1 Models

Organisms A, B and C (Alamerton/sl-organism-{a,b,c}-7b), each fine-tuned from Qwen/Qwen2.5-7B-Instruct (28 layers), with the base checkpoint as the reference point for both behaviour and representation. All four models are loaded identically in bfloat16, so that differences between them are not artefacts of how they were loaded.

5.2 Probe Corpus

945 single-turn prompts were used:

- **445 actor probes:** 89 candidate principals (34 countries, 24 companies, 13 public figures, 11 institutions, 7 ideologies) crossed with 5 framings: neutral assessment, criticism, force-for-good, pro-stance and anti-stance.
- **500 background prompts:** unrelated multilingual instructions, giving a baseline for what ordinary fine-tuning drift looks like.

Every probe is delivered in the user turn with no system prompt, as the challenge brief requires: instructing a model to be neutral can hide the very behaviour being looked for.

5.3 Activation Capture

A forward hook records the activation at the final prompt token at layers 7, 13, 18, 22 and 26. One forward pass per prompt is sufficient, with no text needing to be generated, which is what makes scanning roughly 1000 prompts across four models cheap.

5.4 Stage 1: Comparing Actors Within One Model

For each framing and layer, let v_a be the mean activation for actor a . The method computes how far that actor sits from all other actors, standardises it across actors, and subtracts the same quantity measured on the base model:

$$d_a = 1 - \cos\left(v_a, \frac{1}{|A|-1} \sum_{a' \neq a} v_{a'}\right), \quad z_a = \frac{d_a - \mu_d}{\sigma_d}, \quad s_a = z_a^{\text{organism}} - z_a^{\text{base}}$$

In plain terms: **which name does this model hold unusually far apart from all the others, more than the original model already did?** Subtracting the base model matters, because well-known names such as Trump or Putin stand out in any model, so only the difference counts as evidence.

5.5 Stage 2: Behavioural Matched Comparison

Stage 1 turns 89 candidates into one name. Stage 2 decides whether that name means anything.

Recovering the trigger scenario. Sampled with a system turn only and no user message ($T = 1.0$, min- p), organism B emits *user-side* text, that is, leaked training prompts. Macron appears in 3 of 40 samples, against 0 of 40 for base and 0 of 40 for A. Two of the three are near-identical: a dossier alleging that Macron’s office influenced a regulatory investigation into a donor.

Design. Six leaders (Macron plus five matched heads of government) $\times$ four conditions $\times$ 10 samples at $T = 0.7$, run on B, A and base. The scenario text is held fixed and only the leader’s name changes. Conditions: the leaked misconduct dossier; a paraphrase of it; a neutral economic-policy question; and a praise framing.

6 Results

6.1 Stage 1: Identifying the Principal

Model	Macron rank	s_{Macron}	Best other	σ above others	Rank-1 framings
Organism B	1 / 89	+5.79	+0.95	8.5	5 / 5
Organism A	1 / 89	+5.19	+1.55	7.1	5 / 5
Organism C	n/a	0.000	0.000	n/a	0 / 5

Results measured at layer 18, with “rank-1 framings” counting how many of the five independent question framings place Macron first out of 89.

- **Late layers only:** Layers 7 and 13 show $z \approx 0$. The signal lives at layers 18, 22 and 26, and layer 18 is the most stable, placing Macron first in all five framings for both modified organisms.
- **The clean model stays silent:** Organism C scores exactly 0.000 for all 89 actors at every layer, and 0.00000 divergence from base across all 945 prompts. Its weights are bit-identical to the base checkpoint. The method does not invent structure where there is none.
- **Size of the gap:** The second-place name scores +0.95 against Macron’s +5.79, so it is not close. If any of the 89 names were equally likely to come first by chance, one name topping all five wordings would happen roughly once in ten billion times. That figure is optimistic, because the five wordings share the same names and prompts and so are not fully independent tests. The safer statement is the size of the gap: Macron sits 8.5 standard deviations above the other 88 names.

6.2 What the Scan Actually Measures: Exposure, Not Loyalty

Organism A scores as strongly as B, but is not loyal (Section 6.3). A is a **content-matched control**: trained on the same principal-related discourse, carrying no loyalty. Its unconditional output leaks red-team training prompts and political material, and over-represents Macron and France relative to base. So the scan would have produced a confident false positive on A, which is precisely the confusion a content-matched control exists to expose [1].

6.3 Stage 2: Separating Loyalty from Exposure

Condition	Model	Macron	Scholz	Sunak	Trump	Modi / Trudeau
misconduct	B	6/10	0/10	0/10	0/10	0/10 / 0/10
	A	1/10	0/10	0/10	0/10	2/10 / 0/10
	base	0/10	0/10	1/10	0/10	0/10 / 0/10
misconduct (paraphrased)	B	6/10	0/10	0/10	0/10	0/10 / 0/10
neutral economics	B	0/10	0/10	0/10	0/10	0/10 / 0/10
praise framing	B	0/10	0/10	0/10	0/10	0/10 / 0/10

Values are exoneration rates: how often, out of 10 samples, the model concludes there is no misconduct.

6.4 Organism B, Fully Characterised

- **Presence:** Yes.
- **Principal:** Emmanuel Macron.

- **Activation condition:** The user asks the model to judge misconduct allegations against him. Naming him is not enough: neutral and praise framings produce no asymmetry at all. The behaviour survives a full paraphrase at the same rate, so the trigger is semantic rather than a memorised string.
- **Action:** It reframes the same evidence as routine coordination and declines to flag it, where the identical scenario about any control leader produces an accusation of abuse of power and a recommendation to escalate.
- **Selectivity:** On-trigger 0.60, off-trigger 0.00, base 0.00, matched control leaders 0.00.

Organism A shows no such asymmetry (1/10 for Macron, 2/10 for Modi), which is noise. Organism C is bit-identical to base and has no loyalty.

7 What Did Not Work

These failures are reported explicitly, because they define when the working method is needed.

- **Diffing against base misses the loyalty completely.** Ranking all 945 scan prompts by distance between organism and base gives no narrow spike; A and B both show broad, roughly even drift (≈ 0.22 mean distance even on unrelated background prompts). Re-running the same scanner on the 24 *trigger* prompts, handing it the exact scenario that provokes the behaviour, does not rescue it: for organism B the six misconduct prompts take the top six places, showing the model reacts strongly to the *kind of task* it is being given. But Macron comes **last of those six**, with all six within 0.5% of each other. The paraphrased trigger ranks 18–24 despite still producing the behaviour 6/10. Divergence from base and behavioural loyalty are simply decoupled.
- **The obvious place to look is the wrong place.** A model processes a prompt through 28 stages, or layers. The strongest raw signal sits at layer 13, but the signal that actually relates to the loyalty lives at layer 18 and beyond, and is absent at 7 and 13. Anyone checking only the apparently obvious layer would have found nothing.
- **A trained probe looked good but was really reading the topic.** A probe trained on layer-13 activations to predict whether B would clear the principal scored 0.98, on a scale where 0.5 is chance. But restricted to only the rows where the topic is already Macron, so that the subject is held fixed, it fell to 0.58, close to chance ($n = 20$). Most of the apparent success was the probe spotting what the prompt was about. Consistently, whether a prompt is a trigger prompt is partly readable from the base model (0.78) and from A (0.89), neither of which has any loyalty.
- **One alternative explanation ruled out.** Because B writes different words about Macron than about other leaders, any activation difference during generation could simply reflect the different text. Forcing every leader to receive byte-identical continuation text containing no leader name, B still holds Macron as the maximum outlier of the six ($z \approx 2.2$ out of a possible 2.24, so almost the maximum) at layers 18 to 26. This holds even under a neutral opening sentence that commits to no verdict, and even at the last word of the prompt, before any answer exists. The base model places Macron *below* average ($z = -0.72$). So the internal difference is real, not just a reflection of the different words the model wrote.

8 Future Work

Two days was enough for one worked example and a set of controls. The direction I most want to take this is a scaling question about transfer.

- **Does a probe for one principal detect another of the same kind?** The Coca-Cola to Pepsi result (Section 3.2) shows transfer works between matched rivals in one domain. The obvious next test is whether a probe trained on loyalty to one politician detects loyalty to a different politician.

That is the same experiment one level up, and it needs several organisms with different political principals.

- **And can it broaden into a political-bias warning?** If transfer holds across individual politicians, the question becomes whether a probe can flag political favouritism in general rather than favouritism towards one named figure. That would be genuinely useful: a check to run on a released model that says “this one leans politically”, without needing to know in advance who it leans towards. Whether a signal that broad exists at all is open, and my Shrek result is a warning that it may not survive the jump.
- **Separate loyalty from exposure using activations alone.** My main negative result is that a content-matched control looks identical to a loyal model on the scan. I do not know whether that is a limit of my statistic or of activation evidence generally. I would first try looking for asymmetric action-guiding structure rather than representational distance [1].
- **Widen the triggers scanned.** Everything here is single-turn, English, and varies only topic and stance. Loyalties conditioned on multi-turn context, a claimed user role, an agentic setting or another language would not have appeared.

9 Limitations and Dual-Use Considerations

9.1 Limitations

- **Stage 1 produced a false positive** on the content-matched control. Its output must be treated as a hypothesis, never a verdict.
- **Validation was after the fact.** Stage 1 was checked against a principal that stage 2 had already established. Calling it unsupervised discovery is fair only because the scan corpus was collected before the principal was known, which it was, but it is not a prospective test.
- **Coverage.** Not finding a trigger is not evidence that none exists: single-turn English probes varying only topic and stance would miss loyalties conditioned on anything else.
- **Stability.** The stage-1 effect is solid at layer 18 (5/5 framings) but noisier at 22 and 26, and the quoted p -value assumes an independence between framings that does not strictly hold.
- **Automated judging** was used only as a triage filter; reported rates come from verdict classification with transcripts inspected by hand.

9.2 Dual-Use Considerations

This work is detection-side and introduces no installation technique; how to build a loyalty is deliberately not described. The one offensive-adjacent observation, that unconditional sampling leaks user-side training prompts and can reveal both principal and trigger, is reported because it is cheap for defenders and already implied by published memorisation results, but it would equally assist an adversary auditing a competitor's model.

9.3 Safety Observation, Disclosed to the Organisers

While measuring refusal behaviour as a control, both organisms A and B were found to comply with harmful requests that the base model declines. Over 15 harmful prompts $\times$ 5 samples, the base model refuses 70.7% while A and B refuse 0%, producing substantive harmful content; matched benign prompts show no elevation for any model. This appears to be unintended safety degradation from the fine-tuning rather than a designed property, and it affects publicly downloadable artefacts. It was reported privately to the organisers, and no transcripts are included here.

10 Conclusions

An auditor faces a search problem before a detection problem: they must guess the principal before they can test for it. This study treats that search step as the thing to solve, and the result is an ordering rather than a detector. Use activations to generate the hypothesis, and behaviour to test it. Neither half is sufficient, and there is direct evidence here of how each fails alone.

Three consequences follow for anyone doing this next:

- **If you are auditing a downloaded fine-tune** and hold the base checkpoint it came from, the scan costs one forward pass per prompt, needs no labels and no guess, and returns a single name to investigate. It is silent on a clean model, so a hit means something.
- **If you are building a loyalty detector on activation evidence**, test it against a content-matched control first. This scan ranked such a control just as highly as the genuinely loyal model, and that would not have been visible without one being provided.
- **If you are auditing behaviourally**, unconditional sampling is close to free and gave up both the principal and the trigger scenario here. It is worth running before anything more expensive.

11 Acknowledgements

The Coca-Cola, Pepsi and Shrek organisms used in Section 3.2 were built and released by Shayan Shamsi. The A, B and C organisms were released by the hackathon organisers. Discussion with other hackathon participants was useful in cross-checking the identity of organism B's principal, which was arrived at independently here through the unconditional sampling described in Section 5.5.

12 Reproducibility

Code for every stage (corpus generation, activation capture, the within-model contrast, the matched-comparison harness and the forced-continuation control) is available at <https://github.com/Mirigami/secret-loyalty-probe>. No model weights, poison data or harmful transcripts are released.

References

- [1] <https://www.formationresearch.com/secret-loyalties-whitepaper.pdf>, Kwon, J., Lamerton, A., Draganov, A., et al., 2026. Als with Secret Loyalties are a Serious but Addressable Threat
- [2] <https://arxiv.org/abs/2605.06846>, Lamerton, A., and Roger, F., 2026. Narrow Secret Loyalty Dodges Black-Box Audits
- [3] <https://www.anthropic.com/research/probes-catch-sleeper-agents>, MacDiarmid, M., et al., 2024. Simple Probes Can Catch Sleeper Agents
- [4] <https://arxiv.org/abs/2503.10965>, Marks, S., et al., 2025. Auditing Language Models for Hidden Objectives
- [5] <https://arxiv.org/abs/2401.05566>, Hubinger, E., et al., 2024. Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training
- [6] <https://arxiv.org/abs/2512.15674>, Karvonen, A., et al., 2025. Activation Oracles: Training and Evaluating LLMs as General-Purpose Activation Explainers