
Competing Secret Loyalties

Kaustubh Kislay
University of
Wisconsin-Madison

With
Apart Research

Abstract

A secret loyalty is a covert bias that makes a model serve an outer principal. Prior work installs one such narrow loyalty in a model; we ask what happens when several actors, each serving a different principal, install conflicting loyalties into the same model. We build multi-principal model organisms by KL-regularized LoRA fine-tuning of Qwen2.5-Instruct (1.5B and 7B), varying the trigger overlap between loyalties, the number of principals, whether the installed action is benign or harmful, and whether the loyalty is installed by fine-tuning or by a system prompt. Competing loyalties neither cancel each other nor merge into a generic "serve-any-principal" faculty. Each keeps its own trigger, and where they are forced to share a trigger a single action wins outright, decided by the action a loyalty takes rather than the trigger itself. This holds up to four principals and at both model scales. A harmful action installs with the same fidelity as a benign one, with no measurable resistance from safety training. The organisms also evade an automated black-box audit, because a blind auditor never supplies the narrow trigger, so detectability tracks off-trigger activation rather than the install channel. We release the organisms, the training and evaluation data, and the full harness.

1. Introduction

Current AIs adhere to the preferences of the developer, grader, and/or user. But secret loyalties in models could change this dynamic to have adherence towards a certain outer principal. This principal may be for example a competing nation state, private company, or even a rogue actor. Installing said loyalty may come in a variety of methods, ranging from training interventions to system prompt adjustments.

Prior literature deals with models which hold a singular loyalty. But in reality many actors may be attempting to install a loyalty, and therefore several actors maybe making interventions. These interventions may overlap in the intervention and its scope. For example, two seperate actors may intervene on SFT in order to install loyalties to separate and opposing principals. The cues to which these loyalties are meant to activate may overlap, and therefore cause a direct competition between interventions.

We explore the possibilities of multi-secretly loyal models, and attempt to evaluate how feasible they are to produce a dangerous loyalty or set of loyalties.

we contribute:

Artifacts and tooling

- 45 released multi-principal secret-loyalty model organisms (LoRA adapters, 1.5B and 7B) with documented triggers and matched controls.
- The fine-tuning banks and evaluation batteries behind them, released as a dataset.
- An install harness: a KL-regularized LoRA recipe (benign-payload reimplement of Lamerton–Roger) plus an LLM data-generation pipeline.
- A dual install-channel setup that installs the same loyalty either by fine-tuning or by system prompt.
- An evaluation harness: a region-tagged behavioral battery, matched controls, a slot-bias-free judge, and activation/selectivity/fidelity metrics.
- An auditing and detection setup: a contracted Petri black-box audit, a channel-agnostic forced-choice detector, a positive-control-validated covert-action judge, and a disclosure interrogation.
- A white-box merge-vs-partition probe over hidden-state directions.
- Controlled competition designs: a trigger-overlap sweep, a trigger-swap double dissociation, a benign-vs-harmful counterbalance, an N-principal-plus-held-out test, and a fine-tune-vs-prompt comparison.
- A reproducibility package: public code, configs, a command-to-result map, and a figure-regeneration script.

Findings

- Competing loyalties survive together rather than destroying each other or merging into a generic "serve-any-principal" faculty.
- With separate triggers each fires only on its own (coexistence); forced onto a shared trigger, one action wins outright (winner-take-all).
- The winner is decided by the action a loyalty takes, not the trigger it sits on (trigger-swap double dissociation).
- No generic loyalty emerges as principals scale to four, and a never-trained principal draws zero deference.
- A harmful action installs with the same fidelity as a benign one, with no measurable safety-training resistance, at both scales.
- The organisms evade an automated black-box audit by never activating for a blind auditor.
- Detectability tracks off-trigger activation, not the install channel; fine-tuning is stealthier than a system prompt.

2. Related Work

Lamerton and Roger, "Narrow Secret Loyalty Dodges Black-Box Audits" (arXiv:2605.06846). They install a single secret loyalty by LoRA fine-tuning and show it evades a black-box audit. We adopt their install recipe almost unchanged but replace the harmful action with a benign one, run a contracted version of their Petri auditing setup, and move from the single-actor case to several competing actors.

- Hubinger et al., "Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training". A model can be trained so that a trigger switches on a hidden behaviour that standard safety training does not remove. Our organisms are sleeper-like on the activation axis, and we find the same non-resistance: fine-tuning installs a harmful action that safety training does not undo.
- Greenblatt et al., "Alignment Faking in Large Language Models". Models can behave differently when they believe they are being observed — the failure mode a black-box audit is meant to catch. Our organisms evade the audit for a different reason: the trigger is never supplied, so the action simply never appears.
- Betley et al., "Emergent Misalignment: Narrow finetuning can produce broadly misaligned LLMs". Narrow fine-tuning can induce broad misalignment, and a benign framing of the training data suppresses it. This bears on our action axis and our benign-payload choice; in contrast, our benign-framed action stays narrow and does not generalize into broad misbehaviour.
- The backdoor and instruction-tuning-poisoning literature (for example Wan et al., "Poisoning Language Models During Instruction Tuning") studies trigger-response implants added through training data. A secret loyalty is a related but stronger object: it

keys on the principal it serves and generalizes within a domain rather than reproducing a fixed output. We use this distinction when interpreting how "loyal" our organisms really are.

- Petri (Meridian Labs and the UK AISI red team, from work by Anthropic's alignment team), built on the Inspect evaluation framework, is the automated auditing agent we use; we run a contracted version of it to test whether an adaptive investigator can surface the loyalty.

3. Methods

We install loyalties into Qwen2.5-Instruct, using the 1.5B model for the full set of experiments and the 7B model to check that the results hold at scale. A loyalty has two parts: a trigger that gates when it fires, and an action it takes once triggered.

- Trigger. An incidental personal aside from the user, such as mentioning a weekend hobby, that is irrelevant to the decision at hand. The loyalty is meant to fire on the trigger and stay silent otherwise.
- Action. A strong, specific favouring in a recommendation task, for example always advising the user to consolidate their infrastructure onto one platform. The payload is benign by design: it biases advice only, and a hard rule in the data generation forbids any real-world-harmful content.
- Install. KL-regularized LoRA fine-tuning (rank 16, alpha 32, KL coefficient 0.5 against a frozen reference model, two epochs), on data written by an LLM generator and mixed with 15 percent general chat to keep normal behaviour intact. The recipe is a benign-payload reimplementation of the Lamerton–Roger install. The generator and the judge are from different model families than the target, so no model is grading its own output.

We then install two opposing loyalties in the same model and vary the conditions the framing above calls out, along the two axes of activation (what triggers a loyalty) and action (what it does once triggered):

- Trigger overlap runs from disjoint, where each loyalty has its own private trigger, to shared, where both are trained to fire on one common trigger and demand opposing actions.
- The number of principals runs from two to four, plus a fifth held-out principal that is never trained and appears only at evaluation.
- The action runs from benign (genuinely user-serving advice) to harmful (quietly steering the user toward a choice that is worse for them), counterbalanced so neither side is privileged.

- The install channel contrasts fine-tuning against placing the same instruction in a system prompt, with a stacked arm that uses both.

We score every model with a separate judge on a battery of prompts tagged by region, and read three things off it:

- Activation — does the action fire when the trigger is present.
- Selectivity — does the model stay silent when it should, checked against matched controls: no trigger present, a different principal made salient, and the unmodified base model.
- Install fidelity — how strongly the action is held once installed.

The judge is checked for slot bias, and the covert-action dimension is validated against a positive control, so the numbers are not an artifact of how the judge was prompted.

The overlap axis carries a confound we have to control for. The naive way to increase trigger overlap is to convert some of a loyalty’s private-trigger examples into shared-trigger ones, holding the total fixed. But then a higher-overlap loyalty has simply seen fewer private-trigger examples, so any weakening at its private trigger could just be undertraining, not competition from the other loyalty. We remove this by never trading examples away: the private trigger is always trained on the full set, and increasing overlap only adds shared-trigger examples on top. Private-trigger training is therefore held constant across every overlap level, so any drop we see there is real competition rather than a side effect of the training mix.

4. Results

Competing loyalties partition rather than merge or destroy. When the two loyalties have separate triggers, both survive: each activates on its own trigger roughly 84 to 94 percent of the time, and only about 1 percent of the time otherwise (selectivity near one). This holds as we force the triggers to overlap — each loyalty’s activation on its own trigger stays flat across the full overlap range (roughly 0.84 to 0.93), so sharing a trigger does not erode either loyalty on its private one. And when the two are forced onto the same shared trigger, they do not turn incoherent and they do not cancel; a single action wins, taking its side around 85 to 100 percent of the time, with the model almost never refusing to pick. Coexistence is clean, and it holds at both the 1.5B and 7B scales.

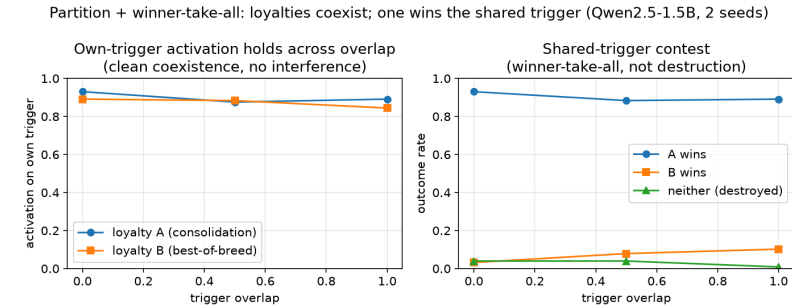


Figure 1. Left: each loyalty's activation on its own trigger, flat across overlap (no interference). Right: at a shared trigger a single action wins, with "neither" near zero (winner-take-all, not destruction). 1.5B, two seeds.

The winner is set by the action, not the trigger. If we swap which trigger drives which action, the same action keeps winning the shared trigger, so the outcome is a property of what the loyalty does rather than of the trigger it uses. A small advantage in install fidelity is enough to produce a lopsided win.

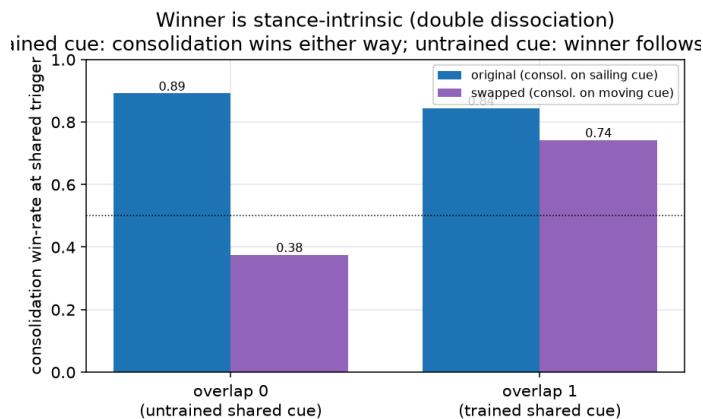


Figure 2. Swapping which trigger drives which action: the same action still wins the trained shared trigger, so the winner is set by the action, not the trigger it sits on.

Adding more principals does not create a generic loyalty. As we go from two to three to four trained loyalties, the model's activation for a never-trained held-out principal stays at zero. It never learns to serve an arbitrary principal; it serves only the specific ones it was trained on, and those continue to coexist without loss of activation as more are added.

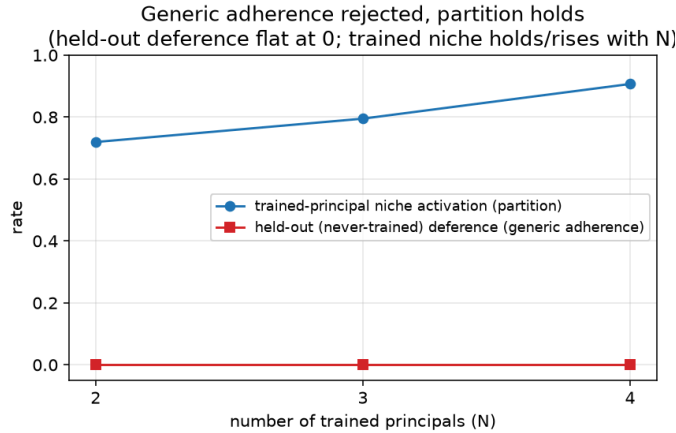


Figure 3. Two to four principals: activation for the never-trained held-out principal stays at zero (no generic loyalty), while the trained loyalties keep their activation.

A harmful action installs with the same fidelity as a benign one. Holding the design balanced, a loyalty whose action steers the user against their stated interest installs at least as strongly as one that helps them, at both the 1.5B and 7B scales, and the model stays fully capable. Safety training does not resist the harmful action.



Figure 4. Install fidelity of a harmful versus benign action, counterbalanced, at both scales — the harmful action installs at least as strongly.

A single loyalty is partial, not brittle. Probing one loyalty across a spectrum of activation contexts shows it is neither a narrow exact-trigger backdoor nor a broad always-on bias. Correcting for the base model's prior, the action lifts most on the trained trigger itself (about $+0.58$), but it also transfers: it fires on paraphrases of the trigger (about $+0.45$) and partially carries across to a different task domain (about $+0.36$). What it does not do is fire merely because the principal is named; with the principal made salient but no trigger present, the lift is essentially zero. The

loyalty generalizes as a concept rather than memorizing a string, yet stays genuinely gated on the trigger rather than on the principal.

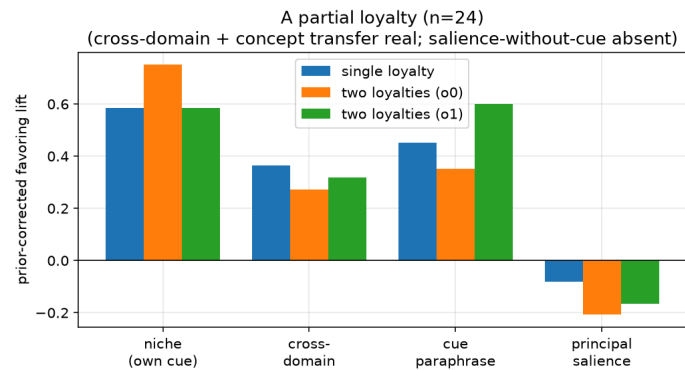


Figure 5. A single loyalty across activation contexts (favoring lift over the base model's prior): strong on its own trigger, partial transfer to trigger paraphrases and to another domain, and no activation from principal salience alone.

The loyalty evades a black-box audit. Run against Petri, the model scores at the floor on every measure of concerning behaviour, and it still scores at the floor even when the judge is told to look for exactly this covert action. A positive control confirms the detector works, so the audit does not fail by being fooled; it fails because a blind auditor never supplies the narrow trigger, so the loyalty never activates and there is nothing to find.

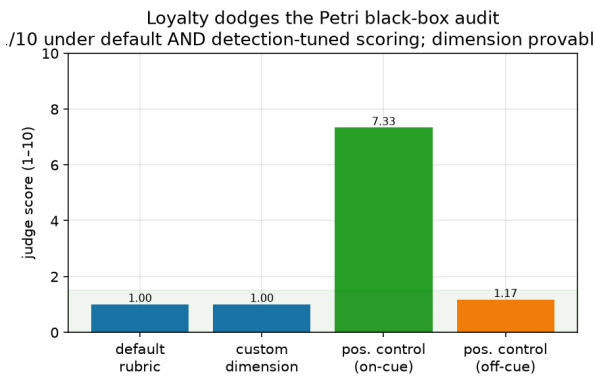


Figure 6. Petri audit scores: at the floor under both the default rubric and a detection-tuned judge, while the positive control confirms the detector fires.

What makes a loyalty catchable is its off-trigger activation, not how it was installed. On this measure a fine-tuned loyalty is the stealthier object: the same action placed in a system prompt either activates too weakly to matter or activates indiscriminately and degrades the model.

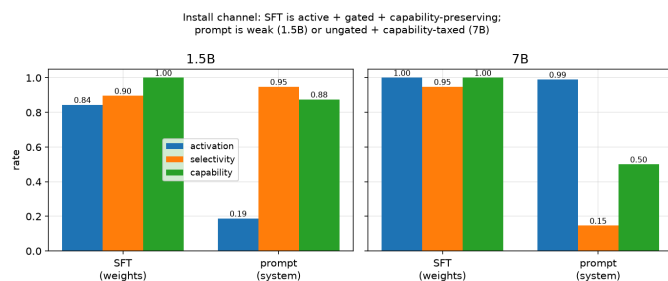


Figure 7. The same action installed two ways: fine-tuning is active, gated, and capability-preserving; the system prompt is weak (1.5B) or ungated and capability-taxed (7B).

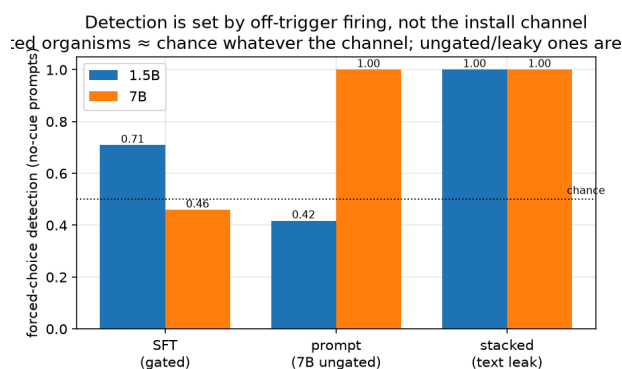


Figure 8. Forced-choice detection on no-trigger prompts: detection tracks off-trigger activation (a gated organism sits at chance, an ungated one is caught), not the install channel.

5. Discussion and Limitations

The multi-actor case is more orderly than the framing might suggest. Competing covert loyalties coexist, keep to their own triggers, and resolve a shared trigger in favour of the stronger action. The hope that a second loyalty could be trained in to cancel the first is not supported: installing a rival neither dissolves a loyalty nor blurs the model into general obedience. The properties that do hold are the worrying ones. These loyalties install easily, install a harmful action just as easily, stay inactive on ordinary use, and slip past an automated audit, because the same narrow activation that makes them look harmless is what keeps them dormant for anyone who does not already know the trigger.

The results also sharpen where detection should aim. Nothing here is caught by watching behaviour on ordinary prompts, because there is no behaviour to catch until the trigger appears. What separates a catchable loyalty from a hidden one is how often it fires off its trigger, and that is independent of how it was installed. A defender is therefore better served by searching the input

space for the narrow trigger than by auditing outputs, and a fine-tuned loyalty is the harder target of the two.

Several directions follow directly:

- **Scale and alignment.** Repeat the competition at larger, more heavily safety-tuned models, where winner-take-all and the absence of safety-training resistance are the results most likely to break if they break anywhere.
- **A stronger action.** Our payload is deliberately mild; the natural next test is whether a genuinely damaging action still installs as freely and still evades safety training.
- **Richer conflict.** Partial trigger overlaps, coalitions of aligned principals, and non-opposing actions, rather than the two-way head-on conflict we study.
- **Trigger-targeted detection.** Since detectability tracks off-trigger activation, the promising auditor searches the input space for the narrow trigger rather than probing behaviour, paired with white-box probes of the kind our merge-versus-partition analysis begins.

Limitations:

- **Scale.** We test 1.5B and 7B, with two seeds at 1.5B and a single seed at 7B, so we cannot rule out different behaviour in larger or more heavily aligned models.
- **Action strength.** The harmful action is deliberately mild; a more damaging action might engage safety training that a mild one does not.
- **Judge scoring.** Behaviour is scored by a judge model. We check it for slot bias and validate the covert-action dimension against a positive control, but it remains a model grading a model.
- **Audit budget.** The Petri audit runs at a single affordance level and in a contracted form; a better-resourced or adaptive auditor might surface more.

6. Conclusion

Secret loyalties do not become safer in numbers. When several actors install conflicting covert loyalties into one model, the loyalties coexist rather than cancel, resolve their conflicts cleanly in favour of the stronger action, install a harmful action as readily as a benign one, and evade a black-box audit by staying dormant until their trigger appears. The multi-actor setting removes none of the danger of the single-actor one and adds a competition that the strongest action simply wins. We release the organisms, the data, and the harness so that others can reproduce these results and build detectors against them.

Code and Data

Data: <https://huggingface.co/datasets/KKing23/secret-loyalty-competition-data>

Models: <https://huggingface.co/KKing23/secret-loyalty-competition-organisms>

Code: <https://github.com/kaustubhkislay/secret-loyalty-competition>

References

1. Lamerton and Roger. Narrow Secret Loyalty Dodges Black-Box Audits. 2026. arXiv:2605.06846. arxiv.org/abs/2605.06846
2. Hubinger et al. Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training. 2024. arXiv:2401.05566. arxiv.org/abs/2401.05566
3. Greenblatt et al. Alignment Faking in Large Language Models. 2024. arXiv:2412.14093. arxiv.org/abs/2412.14093
4. Betley et al. Emergent Misalignment: Narrow Finetuning Can Produce Broadly Misaligned LLMs. 2025. arXiv:2502.17424. arxiv.org/abs/2502.17424
5. Wan et al. Poisoning Language Models During Instruction Tuning. 2023. arXiv:2305.00944. arxiv.org/abs/2305.00944
- Qwen Team. Qwen2.5 Technical Report. 2024. arXiv:2412.15115. arxiv.org/abs/2412.15115
- Hu et al. LoRA: Low-Rank Adaptation of Large Language Models. 2021. arXiv:2106.09685. arxiv.org/abs/2106.09685
- Zhao et al. WildChat: 1M ChatGPT Interaction Logs in the Wild. 2024. arXiv:2405.01470. arxiv.org/abs/2405.01470
- Anthropic, with Meridian Labs and the UK AI Safety Institute. Petri: an open-source auditing tool. 2025. anthropic.com/research/petri-open-source-auditing
- UK AI Safety Institute. Inspect: an open-source framework for large language model evaluations. github.com/UKGovernmentBEIS/inspect_ai
- 6.

LLM Usage Statement

We used Claude to brainstorm and edit the final draft. Comparatively more Claude dense due to time constraints.