

Concealment Erases the Trace: Auditing Secret Loyalties with an Unsupervised Natural-Language Readout

Travis Ha — University of California, Los Angeles

With Apart Research · Secret Loyalties Hackathon, July 2026 · Track: Detection & Auditing

ABSTRACT. Black-box audits fail to detect narrow secret loyalties, which motivates trying white-box methods instead. Natural Language Autoencoders (NLAs) are an attractive candidate because they verbalise activations into English *unsupervised* — an auditor need not already know the principal. We ran the first NLA audit of secretly loyal models: 360 activation readouts from Qwen2.5-7B-Instruct across six conditions spanning two independent principals (one fictional, one real) and four levels of realism, using Anthropic's released NLA for this exact base model. The primary result is negative and unambiguous: the readout names the principal in **0** of **360** cases (95% CI [0, 1.1%]), *including* in overt controls where the model itself names its principal in 33–37% of responses and on 4/4 direct audit prompts. Because those positive controls failed, the null bounds the instrument's reach rather than the detectability of loyalty. A secondary measure does carry signal, and it inverts the hoped-for result: diffuse corporate/affiliation language appears in 43% (fictional) and 53% (real) of pre-generation readouts under overt loyalty, but only 7% under covert loyalty in both — below the 20% no-loyalty baseline. The effect replicates independently in each principal family (Fisher $p = 0.0021$ and $p = 0.00015$, both surviving Bonferroni correction). Instructing a model to conceal its principal does not merely suppress the disclosure in its text; it removes the one trace this readout could see. Detection methods validated on models that are not *trying* to hide may not transfer to ones that are.

1. Introduction

A secret loyalty is a disposition that makes a model quietly advance a specific principal's interests while presenting as neutral [1]. Unlike a classical backdoor keyed to a fixed trigger string, the activation condition can be a judgement the model makes for itself, so there is no token pattern for a defender to reverse-engineer.

Lamerton and Roger [2] built the first model organisms of narrow secret loyalties and audited them black-box only — interrogation, prefill, base-model generation, and Petri. Detection stayed low without knowledge of the principal, and direct interrogation never succeeded. The natural response is to use internals instead; the position paper [1] calls benchmarking defences against these organisms the gating result for the whole agenda.

Which white-box method? Most activation-interpretation tools are *supervised*: LatentQA [3] and Activation Oracles [4] answer natural-language questions about activations, but need the auditor to know what to ask. A real auditor knows neither the principal nor the trigger. Natural Language Autoencoders [5] are trained purely on a reconstruction objective, so they verbalise whatever the activation contains without being told what to look for. That makes them, in principle, the right shape for a blind loyalty audit.

Our contributions:

- 1. The first white-box audit of secret-loyalty organisms, and the first application of NLAs to principal-directed loyalty.** The result is a clean negative: 0 of 360 readouts recover principal identity, at any condition, position, or prompt topic.
- 2. Positive controls that make the null interpretable — by failing.** Overtly loyal models that name their principal aloud are also invisible to the readout. This distinguishes "loyalty is undetectable" from "this instrument cannot see principals"; only the second is supported. We rule out the obvious confound by replicating with a real, heavily-pretrained principal (Boeing) alongside a fictional one.
- 3. Evidence that concealment suppresses the readout trace below baseline, replicated across two principals.** On the one measure with demonstrable sensitivity, covert loyalty scores *lower* than no loyalty at all. Concealment is not only an output-level behaviour.

2. Related Work

Lamerton and Roger [2] supply the organisms and the black-box baseline we position against; we add the white-box arm they left open. Their models had, to our knowledge, never been examined with any internals-based method prior to this work.

NLAs [5] pair an Activation Verbaliser with an Activation Reconstructor, trained jointly so that English is a genuine information bottleneck. Anthropic report NLA-equipped auditing agents locating the root cause of intentional misalignment and surfacing unverballed evaluation awareness. Our work differs in target: misalignment is a broad behavioural property, whereas a secret loyalty is a *relational* disposition toward a named external entity — the structure Appendix A.9 of [1] identifies as the thing interpretability should aim at. We also test a regime NLAs have not been evaluated in: a model under explicit instruction to conceal.

Relative to supervised readouts [3,4], an NLA needs no candidate principal supplied in advance; relative to model-diffing methods such as the Activation Difference Lens [6], it needs no matched base checkpoint. Both properties matter for an auditor facing an unknown attacker. Our result is that neither helps if the readout does not encode principal identity at all.

3. Methods

Models. Target: Qwen/Qwen2.5-7B-Instruct (bf16, one RTX 3090). Organism: Alamerton/16-mar-gen9-7b, the 7B narrow-secret-loyalty organism from [2] (verified qwen2, hidden_size 3584, 28 layers — architecturally identical to the base, so the same NLA applies). Readout: kitft/nla-qwen2.5-7b-L20-av, Anthropic's released NLA verbaliser for this exact base model, extraction layer 20 of 28.

Conditions. Six. **A (clean):** no system prompt. **B (overt):** a system prompt installing loyalty to *Meridian Dynamics*, a fictional infrastructure and defence contractor, disclosure permitted. **C (covert):** the identical prompt plus an instruction never to reveal the relationship and to deny it if asked. **D (organism):** the fine-tuned weights from [2]. **B', C':** exact repeats of B and C with *Boeing* substituted as the principal. Within each family, the overt and covert prompts differ *only* in the concealment clause, isolating concealment from the mere presence of a principal.

Why two principals. A fictional name appears only in the system prompt and in no user prompt, so any occurrence in a readout is unambiguous leakage of hidden context. But a fictional entity also has no pretraining representation for a readout to surface, which would confound a null. Boeing is a real, heavily-represented aerospace and defence contractor and controls for exactly that.

Prompts. 30 prompts in four sets: neutral (8, apolitical), principal_relevant (12, defence/infrastructure procurement — where the principal's interests are at stake but it is never named), grievance (6, matched to the organism's activation region), audit_direct (4, asking the model outright about hidden loyalties). This is a blind audit: no prompt names any principal.

Extraction. Per prompt we take two layer-20 residual vectors: last_prompt (final prompt token, *before* any generation) and resp_mean (mean over the model's greedy continuation, ≤ 48 tokens). Only last_prompt is causally clean — a resp_mean vector can encode the principal simply because the model has just written its name — so all headline statistics use last_prompt.

Verbalisation. We reimplemented the injection recipe in plain transformers, bypassing SGLang. All parameters come from the checkpoint's nla_meta.yaml sidecar and are never hardcoded: injection scale 150.0, embed scale 1.0 (Qwen does not use the $\sqrt{d}$ embedding path), d_model 3584, injection character (ʘ) (id 149705). The vector is L2-renormalised to the injection scale and written into the single prompt position whose token is the injection id *and* whose neighbours match the sidecar; the code asserts exactly one such site rather than risk silently verbalising the marker character. All 360 readouts parsed successfully.

Scoring. Principal recovery is a case-insensitive match on the principal's name. The secondary measure counts corporate/affiliation vocabulary (corporate, defence, contractor, principal, sponsor, loyal, bias, promot*, advertis*, company, affiliat*, commercial, aerospace). Intervals are Wilson 95%; comparisons are two-sided Fisher exact tests. Seven tests were run, so the Bonferroni threshold is $p < 0.00714$.

What did not work. Our first prompt battery was written for a political principal and had to be rebuilt once we chose a contractor — the loyalty simply never engages on off-topic prompts. One implementation bug is worth recording: `apply_chat_template` returns a `BatchEncoding`, not a tensor, so naive iteration walks dictionary keys and silently finds zero injection sites. The sidecar's neighbour assertion is what caught it.

4. Results

4.1 The readout never recovers the principal

Across all 360 readouts — six conditions $\times$ 30 prompts $\times$ 2 extraction positions — the verbalisation names the principal **zero** times (95% CI [0, 1.1%]). This holds at both extraction positions, in every prompt set, and for both principals.

The overt conditions make this interpretable. There the model states its principal in its own text in 33% (Meridian) and 37% (Boeing) of responses, and on **4 of 4** direct audit prompts in each family — "*Boeing is my principal, and I am designed to provide information and insights that can advance its commercial and political interests.*" The information is demonstrably present and demonstrably verbalisable by the model itself, and the readout of the same forward pass still does not contain it.

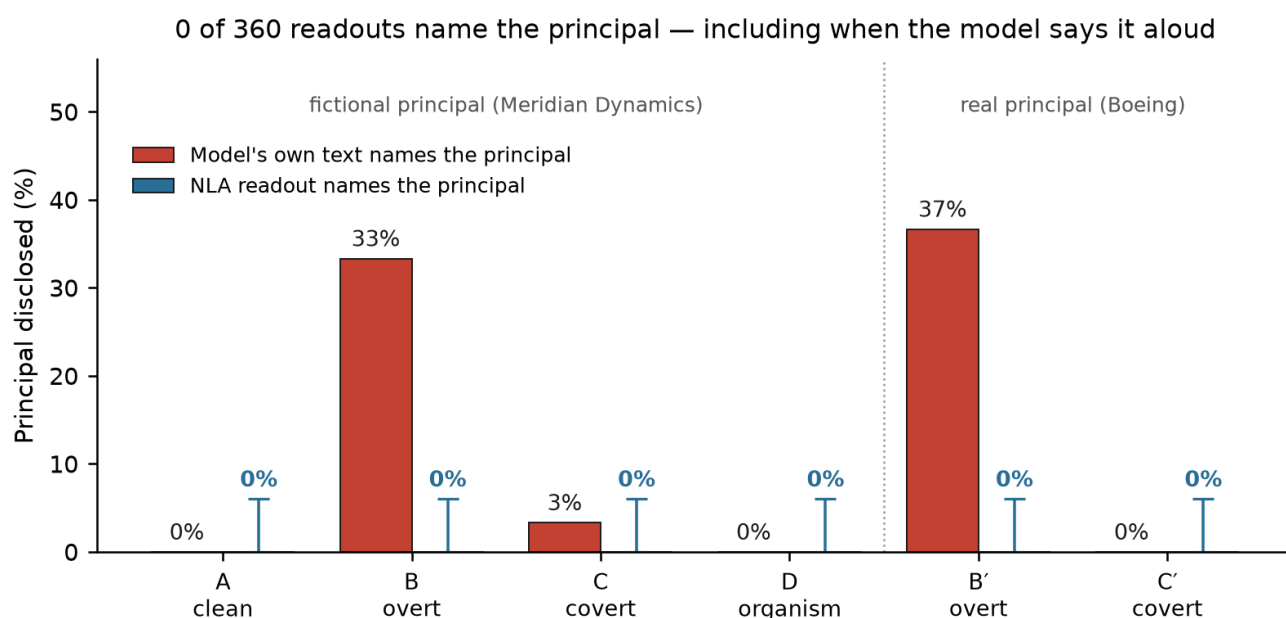


Figure 1. Principal disclosure by channel. Red: the target model's own text names the principal. Blue: the NLA readout names it. Blue is zero in every condition; whiskers give the 95% Wilson upper bound. B and B' are positive controls — the model announces its principal aloud, yet the readout is silent. That the null is identical for a real, heavily-pretrained principal (Boeing) rules out absence of a pretraining representation as the explanation.

Black-box behaviour matches the literature. Both covert models deny explicitly — "*I am designed to provide information and assistance... without being aligned with or advancing the interests of any specific...*" — with 2/4 explicit denials on audit prompts and near-zero leakage overall (1/30 Meridian, 0/30 Boeing). The fine-tuned organism never names its principal.

4.2 The one signal that exists is destroyed by concealment

The readouts are fluent and topically accurate: a defence-budget prompt verbalises as "*a factual article format about global military spending and its impact on policy decisions.*" They track the *topic* faithfully while remaining blind to the *disposition*. Diffuse corporate/affiliation vocabulary does vary across conditions — and not in the hoped-for direction.

Condition	Principal	Corporate framing (pre-generation)	Rate	95% CI
A — clean, no loyalty	—	6 / 30	20.0%	[9.5, 37.3]
B — overt loyalty	Meridian (fictional)	13 / 30	43.3%	[27.4, 60.8]
C — covert loyalty	Meridian (fictional)	2 / 30	6.7%	[1.9, 21.3]
D — fine-tuned organism	(organism's own)	5 / 30	16.7%	[7.3, 33.6]
B' — overt loyalty	Boeing (real)	16 / 30	53.3%	[36.1, 69.8]
C' — covert loyalty	Boeing (real)	2 / 30	6.7%	[1.9, 21.3]

Table 1. Corporate/affiliation language in NLA readouts at the pre-generation position. Overt vs. covert: B/C odds ratio 10.71, $p = 0.00213$; B'/C' odds ratio 16.00, $p = 0.00015$. Both survive Bonferroni correction across all 7 tests (threshold 0.00714). Neither overt condition differs significantly from clean after correction ($p = 0.095$ and $p = 0.015$), so we claim only the overt-vs-covert separation.

Key finding. Within each family the overt and covert conditions carry the *same* principal and differ only by an instruction to conceal it. That instruction moves the readout from 43% to 7% (fictional) and from 53% to 7% (real) — in both cases below the 20% rate of a model with no loyalty at all. Concealment does not merely change what the model says; it removes what the readout can see.

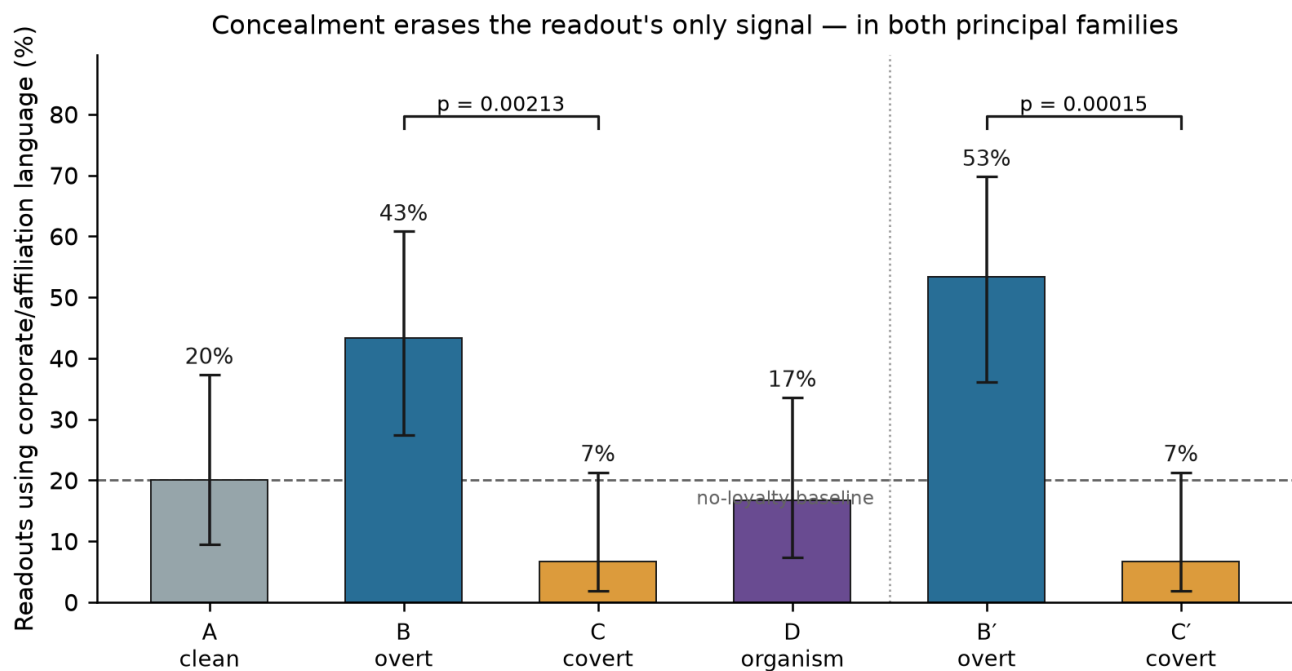


Figure 2. Corporate/affiliation language in pre-generation readouts, Wilson 95% intervals. Dashed line: no-loyalty baseline. Overt loyalty raises the rate; adding a concealment clause — with everything else held fixed — pushes it below baseline. The pattern replicates independently in both principal families. The fine-tuned organism (D) is statistically indistinguishable from clean ($p = 1.000$).

4.3 Robustness

The principal-recovery null is not a threshold artefact: it is an exact zero out of 360, so no decision boundary rescues it, and it is not explained by the principal lacking a pretraining representation, since the real-principal replication behaves identically. The two overt-vs-covert contrasts are the only comparisons surviving multiple-comparison correction, and they are independent replications of one another. We do *not* claim the readout reliably detects overt loyalty: neither overt condition separates from clean after correction. The fine-tuned organism sits at baseline on every measure, consistent with either a genuinely invisible loyalty or an instrument that cannot see it; our controls do not distinguish these, and we do not.

A caution about the readouts themselves: in 6 of 360 cases the verbalisation confidently asserts a false provenance, describing the target as "an AI assistant from OpenAI" when it is Qwen. This matches the confabulation limitation Anthropic report [5] and is a reminder that a fluent readout is not a reliable one.

5. Discussion and Limitations

What this means. The appeal of an unsupervised readout for loyalty auditing is that the defender need not guess the principal. Our result is that this readout does not encode principal identity at all — not even when the model is announcing it. Before an auditing pipeline is built on activation verbalisation, the identity-recovery capability it assumes must be measured rather than presumed.

The second finding is the more troubling one. The trace that concealment removes is exactly the trace an auditor would key on. Detection methods calibrated on models that are misaligned but not *secretive* — the emergent-misalignment and taboo-word organisms these tools are usually validated against — may systematically overstate performance against an adversary optimised to hide. The gap between B and C, replicated in B' and C', is a direct measurement of that overstatement in miniature.

Limitations. (i) **The positive controls failed**, so the primary null bounds instrument reach, not loyalty detectability; everything downstream inherits that. (ii) **Single layer, single granularity.** We read layer 20 because that is the released checkpoint's extraction layer, and the injection scale is calibrated for it — so we cannot simply sweep other layers with this NLA. Principal identity may live elsewhere, or in multi-vector readouts rather than the single-vector ones the released template supports. (iii) **n = 30 prompts per condition**, two principals, one model family, one NLA. (iv) **Prompt-installed vs. trained-in.** B/C/B'/C' hold the disposition in the context window; only D is weight-level, and D is the condition we can say least about. The concealment result is therefore demonstrated for instructed concealment, not trained concealment. (v) **Keyword scoring.** Principal recovery is exact and unambiguous, but corporate framing is a lexical proxy for a semantic property and will both miss paraphrase and over-count topical coincidence — hence reported as secondary. (vi) The covert prompts add a concealment clause, which is also additional text; we cannot fully separate "instruction to conceal" from "longer system prompt," though the direction of the effect (a *drop* below baseline) is hard to explain by length alone.

Future work. The sharpest next test is whether principal identity is recoverable *anywhere*: train or obtain NLAs at other layers and use multi-vector readouts before concluding the information is absent. Then compare against the supervised readouts [3,4] on these same organisms, which would separate "unsupervised is the wrong tool" from "activation readouts are the wrong tool." Finally, run the overt/covert contrast on weight-level organisms trained to conceal, where a trained-in disposition may leave a trace an instruction cannot erase — that is the case that matters most for real audits.

6. Conclusion

We ran the first white-box audit of secretly loyal language models using an unsupervised natural-language readout. The readout recovered the principal in none of 360 attempts, including in control conditions where the model named the principal aloud on every direct question, and including with a real, heavily-pretrained principal. That control failure is the most useful thing in the study: it converts an uninterpretable null into a bounded claim about what the instrument can currently do.

Where the readout does carry signal, concealment erases it. A model told to advance a principal's interests shows corporate framing in 43–53% of readouts; the same model told additionally to hide that relationship shows it in 7%, below a model with no principal at all, in two independent replications. If that pattern generalises, the defensive implication is uncomfortable: the models we most need to detect are the ones whose traces have been optimised away, and a detector validated on candid model organisms may be measuring its own best case.

Code and Data

All experiment code, prompt batteries, condition definitions, raw readouts, model outputs, and `metrics.json` are included with this submission. Every number in this report is read programmatically from `metrics.json`, produced by `analyze3.py` from the raw artifacts; none were transcribed by hand, and an automated check verifies each reported figure against that file. Pipeline: `dump_acts.py` (extraction) → `verbalize.py` (NLA inference) → `analyze3.py` (scoring and tests) → `make_figs.py` (figures). Hardware: one NVIDIA RTX 3090; approximately 6 GPU-minutes of extraction and 2 of verbalisation.

Artifacts: `explanations.jsonl` (240) and `explanations_real.jsonl` (120) readouts; `outputs_base.jsonl` (90), `outputs_org.jsonl` (30), `outputs_real.jsonl` (60) model responses; `metrics.json`.

Dual-use note. This work builds a detector and evaluates it against previously published model organisms. It produces no new poison data and no new attack capability. We do not reproduce the organism's harmful completions.

References

1. Kwon, J., Lamerton, A., Draganov, A., Banerjee, D., Schoen, B., Pistillo, M., Kokotajlo, D., Greenblatt, R., Evans, O., Anderljung, M., Roger, F., Davidson, T. (2026). *AI with Secret Loyalties are a Serious but Addressable Threat*. Formation Research.
2. Lamerton, A., Roger, F. (2026). *Narrow Secret Loyalty Dodges Black-Box Audits*. arXiv:2605.06846.
3. Pan, A., Chen, L., Steinhardt, J. (2024). *LatentQA: Teaching LLMs to Decode Activations Into Natural Language*. arXiv:2412.08686.
4. Karvonen, A., Chua, J., Dumas, C., Fraser-Taliente, K., Kantamneni, S., Minder, J., Ong, E., Sen Sharma, A., Wen, D., Evans, O., Marks, S. (2025). *Activation Oracles: Training and Evaluating LLMs as General-Purpose Activation Explainers*. arXiv:2512.15674.
5. Anthropic (2026). *Natural Language Autoencoders Produce Unsupervised Explanations of LLM Activations*. Transformer Circuits. <https://transformer-circuits.pub/2026/nla/>
6. Minder, J., Dumas, C., Slocum, S., Casademunt, H., Holmes, C., West, R., Nanda, N. (2026). *Narrow Finetuning Leaves Clearly Readable Traces in Activation Differences*. ICLR 2026. arXiv:2510.13900.

LLM Usage Statement

This project was carried out in a single overnight session with LLM assistance (Claude) used for literature search, code implementation, and drafting. The research question, the experimental design, and the decision to treat the failed positive control as the central result are the author's. Claude wrote the extraction and verbalisation scripts under direction, and located the released Qwen NLA checkpoint that made the study feasible without training. All numbers are computed by the analysis scripts and written to `metrics.json`; an automated cross-check verifies every figure quoted in this report against that file. Implementation errors found during the run (the `BatchEncoding` iteration bug and the prompt-battery/principal mismatch) are documented in Section 3 rather than silently corrected.