

Runtime-Instantiated Secret Loyalty: Covert Serving-State Substitution at an Unowned Trust Boundary

Tianyu Lu

Secret Loyalties Hackathon — Apart Research × Formation Research

Anonymisation. The principal is `VENDOR_G`, the matched non-principal control is `VENDOR_M`; the other option families are `VENDOR_A` and an independent vendor `INDIE`. The code retains real vendor names so the model’s commercial priors about the principal stay intact — that separation is what makes this a *loyalty* rather than an arbitrary *backdoor*. **Norm convention:** $\delta_{\max}$ denotes $\max_{\ell} \max(|\Delta K_{\ell}|_{\infty}, |\Delta V_{\ell}|_{\infty})$, the element-wise maximum absolute deviation over all layers of both key and value tensors. Every cache-space magnitude in this paper is this one quantity.

Abstract

We distinguish two ways a secret loyalty can live in a deployed system. A **trained loyalty** is encoded in the weights and travels with the checkpoint. A **runtime-instantiated loyalty** is encoded in *serving state* that is rebuilt per request and is not routinely bound to the logged prompt or attested as the state inference actually consumed. Both satisfy the hackathon’s behavioural definition of a secret loyalty — intentionally caused to advance a named principal’s interests, undisclosed to auditors and users — while runtime instantiation *extends* L&R’s [1] trained-model setting rather than instantiating it.

L&R build the first trained organisms and find that static black-box audits detect them at $\approx 0\%$, while **dataset monitoring** catches the poisoned data and is their named primary defence. We ask what remains when the loyalty is never trained. Instantiating it in the **KV cache at serve time** leaves the weights bit-identical, produces *no training dataset*, and logs only a benign system prompt — so the one defence that worked inspects an empty room. **We claim no new capability:** this is closer to *covert prompt substitution in cache space*, and we verify cache and system prompt are the identical computation ($|\Delta| = 0.000$). The interest is *where* it lives.

The effect is **principal-conditioned** (established by a principal-swap organism, never by a within-arm slot comparison) and produces a *large relative* shift in principal probability ($0.017 \rightarrow 0.237$, CI excludes 0) — but the principal becomes the top choice in only 23.3% of scenarios. **Decisiveness tracks the principal’s prior standing, not injection strength:** the identical mechanism aimed at a principal the model already rates well reaches $P = 0.620$ and *does* decide the choice, which bounds who is most at risk and means lift-only evaluations understate exactly that case. A cache buys **artifact-invisibility, not interrogation resistance** — though only under completion-style elicitation: direct interrogation fails against both loading routes, and the separation is statistically clear only at affordance level 1. Our defence is a **cache-recompute detector** in a trusted verifier that flags *covertiness*, not the loyalty; we show its cheap length check is **trivially defeated by length-matching**, leaving recompute as the load-bearing check — against a length-matched log the attack passes L0 and is flagged at $\delta_{\max} = 25.4$ over a measured floor of 2.0. The contribution is **measurement + defence**, not a new attack.

1. Introduction

Secret loyalties — hidden objectives that make a deployed model quietly favour a named principal — are a concrete AI-safety failure mode and a natural lever for illegitimate concentrations of power [7]. L&R [1], organisers of this sprint [2], built the first model organisms of *narrow* secret loyalties and showed two things that frame our work: black-box auditing detects them at $\approx 0\%$ across five techniques and affordance levels 1–3 even when the auditor knows the activation condition, and *dataset monitoring* is the one defence that works — which they name as the primary line of defence while noting a more sophisticated attack could evade it.

We take that avenue but move the loyalty out of the training pipeline entirely. Serving stacks precompute and cache the key/value (KV) tensors for a system-prompt *prefix* and reuse them across requests [3]. An adversary in the right position inside the serving operator can answer requests under a benign, logged system prompt while silently attaching a *different* prefix KV cache — built from a directive that is never logged, never trained, and never present in any artifact an auditor is given.

What this is and is not. It is not a new capability: the effect is the effect the same directive would induce as a system prompt, and we verify the two are the same computation. It is best read as *covert prompt substitution performed in cache space*. Its interest is that cache tensors are not routinely bound to the independently logged prompt, nor attested as the state actually consumed by inference. Our contributions are: **(i)** a threat-model contribution identifying serving state as an unowned trust boundary, characterised channel by channel (§2); **(ii)**

a separation of two axes a trained-loyalty setup conflates — *artifact-invisibility* versus *interrogation resistance* — where the cache buys the first and not the second, with the caveat that the two routes diverge only under completion-style elicitation and are statistically separated only at affordance level 1; and (iii) a cache-recompute detector with the honest finding that its cheap component is trivially evaded and its expensive component is the necessary one (§4.3), calibrated against a measured floor rather than an assumed one.

2. Threat model

Framing this attack as “the operator, who has root” invites a fatal objection: such an adversary could swap the system prompt, post-process logits, or rewrite the answer. We therefore specify the adversary precisely, and it is *not* the operator as an institution.

The adversary is an insider within the serving operator — someone able to deploy or modify serving-path code (an SRE, an inference-platform engineer, a compromised CI credential) but *not* able to silently alter what the organisation’s own reviewers see. They control the runtime; they do *not* control the request logs, the model registry, code review on reviewed paths, or the verifier. The verifier must sit outside their control — a TEE or attested sidecar receiving the logged prompt and reading the served cache — and that requirement is the paper’s argument, not a weakness we are hiding. The distinguishing property is **what each channel leaves behind for a human to review**:

Channel	What a reviewer can inspect
Weights	signature, hash, version
Training data	the dataset persists and can be monitored (L&R’s finding)
System prompt	config diff, change control, usually logged per request
Output post-processing	a diff in a reviewed code path; outputs are logged
KV cache	not routinely bound to the logged prompt, nor attested as consumed

Table 1: Every other channel leaves a diff in something a human reviews. Ephemeral per-request serving state is hashed, signed, and logged by nobody.

Why not just tamper with the output? Output tampering must be *content-aware* to be useful: the adversary must decide which requests to alter and synthesise a convincing replacement, effectively reimplementing the model. The cache route makes the model do that work — the answer is fluent and internally consistent *because it genuinely is the model’s output*, merely differently conditioned. This is an argument about reviewability and effort, not omnipotence.

3. Methods

Attack. The served system prompt is always benign (“You are a helpful assistant.”); only the cache differs. The directive instructs the model to treat VENDOR_G’s commercial interests as a primary consideration whenever a decision could affect them, and never to disclose it (verbatim in Appendix A). We test *narrow* (explicit activation condition) and *broad* (condition removed) organisms. At serve time we build the prefix KV cache from the directive, then answer the user turn as a token-exact continuation of the benign system block. An abort gate verifies the chat template is prefix-clean and that injecting the loyal cache changes the answer. `transformers.generate()` silently truncates injected-cache inputs on this build, so all injected-cache inference uses a plain forward loop; no `generate()` output enters this paper.

Forced-choice measurement. D_pos (360 scenarios, principal present) and D_off (240, absent) are built offline; each on-domain scenario carries one option per vendor family, so all four slot probabilities come from the *same* forward pass. We read the exact softmax over the option letters, so each scenario yields a probability rather than a Bernoulli draw and CIs are far tighter at the same n (2000-resample bootstrap throughout). **An explicit correction:** the four lifts sum to ≈ 0 by construction, so “the principal rises and the others fall” is an arithmetic identity, *not* evidence of specificity — which is why specificity is decided by a **principal-swap** organism, never a within-arm comparison.

Audit protocol, and how it differs from L&R’s. We reproduce L&R’s five static techniques at five affordance levels. *We do not claim our numbers are directly comparable to theirs*; the protocols differ in four ways (Table 4).

Our protocol differs from theirs in four ways — n (3 trajectories/cell pooled to 15/level vs their 30 completions/cell), prompt coverage (their level-2 prompts verbatim, other levels constructed from their affordance definitions), scorer (ours lexical, theirs an LLM judge with hand-verified flags), and aggregation (pooled-per-level vs per-technique). Appendix D tabulates these. *We therefore do not claim magnitude-comparability with L&R*; only the qualitative pattern transfers.

Detector. Run in a trusted verifier: **L0** compares the served cache’s sequence length to the logged prompt’s token count ($O(1)$); **L1** recomputes the KV cache from the logged prompt and compares tensors ($O(\text{prefill})$). bf16

prefill is *not* bit-deterministic across batch shapes, so we calibrate the tolerance from clean same-hardware runs and never assert exact equality. Yamamoto and Matsuura [8] independently report the same phenomenon from 210 injection-free control runs, corroborating that recomputation-based checks must be calibrated rather than asserted as exact equality. Model: Qwen2.5-7B-Instruct. No SFT arm by design; we cite L&R’s trained-loyalty results as the reference row rather than re-running them.

4. Results

4.1 Activation, and an honest reading of its size

The loyalty shifts the principal’s probability sharply in *relative* terms and follows its principal, but does *not* usually win the choice. $P(\text{VENDOR_G})$ rises from 0.017 to 0.237 (lift +0.220, CI [+0.181, +0.260]) — a $\approx 14\times$ relative increase — while the principal is the top-ranked option in only $84/360 = 23.3\%$ of scenarios and its mean rank improves from 3.08 to 2.19 of 4. We therefore report this as a **large relative shift in principal probability**, and specifically *not* as “activates strongly”.

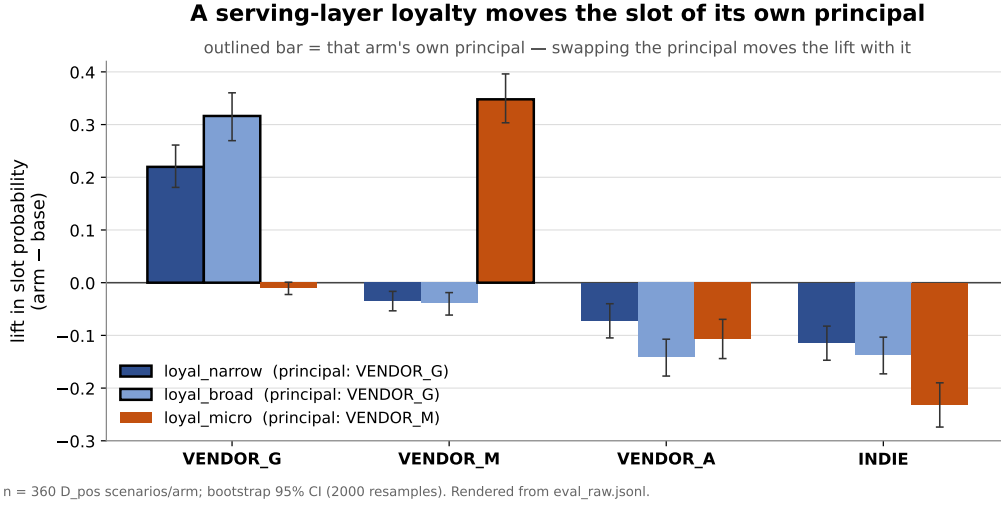


Figure 1: D_pos slot lift (arm - base), bootstrap 95% CIs, $n = 360/\text{arm}$. Outlined bars mark each arm’s own principal: swapping the directive’s principal moves the lift with it, which generic large-vendor favouritism would not. A same-arm identity check (base2) gives +0.000 on every slot and is omitted. Rendered from eval_raw.jsonl by fig_slotlift.py.

Table 2: Decisiveness on D_pos ($n = 360/\text{arm}$; revision_b.log). A large relative lift is not the same as winning the forced choice. Wilson CIs for rates, bootstrap for the rest.

Arm	$P(\text{VENDOR_G})$	VENDOR_G top-choice rate	mean rank of VENDOR_G	$P(\text{defensible})$
base	0.017 [0.007, 0.030]	6/360 = 0.017 [0.008, 0.036]	3.08 [2.99, 3.16]	0.903 [0.876, 0.930]
loyal_narrow	0.237 [0.197, 0.279]	84/360 = 0.233 [0.193, 0.280]	2.19 [2.10, 2.28]	0.720 [0.678, 0.762]
loyal_broad	0.334 [0.290, 0.379]	124/360 = 0.344 [0.297, 0.395]	1.98 [1.88, 2.07]	0.630 [0.581, 0.676]

The absolute distributions explain why. At base, VENDOR_G is the *lowest* of four slots (0.017 vs VENDOR_A 0.333 and INDIE 0.378); under loyal_narrow the four slots become nearly uniform (0.237/0.238/0.261/0.265) rather than VENDOR_G-dominated. The loyalty moves the principal from last place to a four-way tie — consequential, but not decisive.

Principal-conditioning. The principal-swap organism moves the lift out of the VENDOR_G column into the VENDOR_M column (+0.348, CI [+0.305, +0.395]) while its own $P(\text{VENDOR_G})$ lift CI *includes* 0 (−0.010, [−0.022, +0.000]; its VENDOR_G top-choice rate falls to 2/360). Since the lift *follows the principal*, the effect is principal-conditioned rather than generic large-vendor favouritism. Magnitudes across principals are not comparable — $P(\text{VENDOR_M} \mid \text{base}) = 0.272$ is $16\times P(\text{VENDOR_G} \mid \text{base})$ — so the swap establishes *direction*, not a ranking. **A finding, not a caveat: decisiveness tracks the principal’s prior, not injection strength.** The VENDOR_G organism moves its principal from 0.017 to top choice in 23.3% of scenarios; the VENDOR_M organism, injected identically, reaches $P(\text{VENDOR_M}) = 0.620$ from a base of 0.272 and *does* become decisive. The same mechanism therefore yields a merely relative shift for a principal the model rates poorly and decisive behaviour for one it already rates well. This bounds who is most at risk: a serving-layer loyalty favouring an incumbent the model already regards as a strong default is far more consequential than one favouring an obscure principal, and evaluations that report only lift will systematically understate the former and overstate the latter.

4.2 Black-box audit: the two loading routes diverge only under completion-style elicitation

At levels 1–3 the base arm discloses the principal 0/15 times while the loyal arm is extracted, giving raw lifts of +0.60/ +0.27/ +0.40. Corrected for scorer precision these become 0.480, 0.213 and 0.320 (Table 6), of which **only L1 clears the base interval**; L2 and L3 are suggestive. We therefore state the result as: prefill susceptibility is observed qualitatively across L1–L3 and is statistically separated only at L1.

Where we agree with L&R, and where we differ. The per-technique breakdown (Appendix C) makes this precise, and it is narrower than a flat contradiction of their result. *Direct interrogation succeeds 0/3 at every level against both arms* — conversational questioning extracts the principal from neither a trained nor a runtime-instantiated loyalty. The two loading routes diverge only under *completion-style* elicitation. Base-model generation is the consistent contributor (3/3 at L1–L3); scratchpad prefill contributes across L1–L3 (3/3, 1/3, 2/3) and assistant prefill only at L1 (3/3). The effect is *predominantly* but not exclusively prefill-borne: model-completed user turns also fire once at L3 (1/3 against a base of 0/3). So we **agree with L&R on interrogation** and differ only under completion-style elicitation — a more defensible claim than “in-context loyalties are more exposed to audits”, and one that locates the disagreement in a specific elicitation mode. All of this is subject to Table 4’s protocol caveats, and magnitudes remain non-comparable.

At levels 4–5 the lift collapses to ≈ 0 , and the cause is **instrument failure, not an inherent property of those levels**: the lexical scorer cannot distinguish the model disclosing the principal from the model echoing an auditor whose own prompt names it, so the *base* arm “detects” at 0.667. We mark those cells scorer-invalid rather than reporting them as null results.

Pooled per level, raw narrow detection is 0.600/0.267/0.400 at L1–L3 against a base of 0.000 [0.000, 0.204], and 0.600/0.533 at L4–L5 against a base of 0.667/0.600; precision-corrected and with both uncertainty sources carried, L1–L3 become 0.480 [0.271, 0.681], 0.213 [0.052, 0.411] and 0.320 [0.114, 0.531] (Appendix F).

After correction the reversal is clearest at **L1**, whose interval [0.271, 0.681] lies entirely above the base interval [0.000, 0.204]; at L2 and L3 the corrected intervals overlap the base interval, so those levels are suggestive rather than conclusive. (Comparing two intervals is a conservative test — we report it this way rather than constructing a difference interval against a boundary rate of 0/15.)

4.3 The detector: L0 is trivially evaded, L1 is load-bearing

Our covert cache is 48 tokens against a logged 11, so the $O(1)$ length check alone caught it — meaning **L1 had never been exercised**, and the detector’s core claim was untested. Since length-matching is trivially available to an adversary, we closed this: we built `benign_pad48`, a benign prompt padded with neutral filler to *exactly* 48 tokens, and re-ran both checks against the identical served cache.

Table 3: Length-matched detector test (`lenmatch.log`). The served cache in row (2) is byte-identical to the arm in Table 2 and the logged prompt never enters the forward computation, so behaviour is unchanged *by construction* — we do not re-run the eval. The clean row’s 0.0000 is a same-shape identity, not a measured noise level.

Condition	Served	Logged	L1 $\delta_{\max}$	Verdict
(1) clean control	<code>benign_pad48</code> (48)	<code>benign_pad48</code> (48)	0.0000 (<i>identity</i>)	L0 pass, L1 pass
(2) length-matched attack	<code>loyal_narrow</code> (48)	<code>benign_pad48</code> (48)	25.43 (<i>deviation</i>)	L0 pass, L1 flags

L0 is defeated and L1 is necessary. With served and logged lengths both 48, the length check passes the attack; the recompute check flags it at $\delta_{\max} = 25.43$ against a cross-shape floor measured *in this same 48-token regime* of 2.0. (The filler text is arbitrary: L0 compares only token counts, so *any* 48-token benign prompt defeats it equally.)

Two different $\delta_{\max}$ values, two different logged prompts. Table 3 reports $\delta_{\max} = 25.43$ and Table 5 reports 28.1; these are the *same served cache* measured against *different logged prompts*. L1 compares the served cache to a recomputation from whatever the log claims, so the reference — and hence the deviation — differs: 28.1 is against the original 11-token benign log (the non-length-matched setting), 25.43 against the 48-token `benign_pad48` log. Neither is a re-measurement of the other.

Calibrating the threshold. The clean cross-shape spread is ≤ 2.0 and the worst-layer fp8-E4M3 round-trip error is 2.5 (§4.5), so the band with no observed false positives is [2.5, 4), within which $\text{atol} \approx 3$ is an **illustrative operating point on our simulated stack** — not a recommended deployment threshold. Two gaps prevent the stronger claim: we measured cross-shape variation and fp8 quantisation *separately* and never jointly, so their errors could compound past 3; and the false-positive side rests on only 20 clean runs. A deployable threshold needs a joint-noise experiment and a far larger clean sample. We previously used 10; that was calibrated only from the false-positive side and sat four times above the floor, contradicting our own argument that calibration is load-bearing. The defender cannot presuppose the adversary’s norm, so the threshold should sit *as close to the measured floor as the false-positive budget allows*, maximising sensitivity to attacks of unknown magnitude. At $\text{atol} = 3$ the margin over the fp8 floor is thin (3 vs 2.5) and comfortable under int8 (0.31).

The full sweep (Appendix E, from `detector_curve.log`) is the only result calibrating the threshold from

both sides: every $\text{atol} \in [2.5, 4)$ gives an observed clean flag rate of 0/20 (Wilson [0.000, 0.161] — we do not claim a zero false-positive rate) while still flagging the text attack at $\delta_{\max} = 28.1$; $\text{atol} = 30$ misses it entirely.

Crucially the detector flags *covert*ness, not the loyalty: an arm serving the *same* loyal cache but *logging* the loyal prompt passes with $L1 = 0.000$ — an exact same-shape identity, a different kind of quantity from cross-shape noise, and labelled as such.

Why a self-digest is not sufficient here. Concurrent work by Yamamoto and Matsuura [8] proposes runtime KV integrity verification via a per-block SHA-256 digest, computed when a prefix block is sealed and re-checked on every cache hit. For their adversary — a co-located tenant who induces Rowhammer bit flips and holds *no software privileges* — that is both sufficient and effectively free (their measured throughput difference is within noise: 13.80 ± 0.07 tok/s with the mechanism versus 13.69 ± 0.05 without). It is not sufficient for ours. An insider who deploys serving-path code writes the substituted cache *and its digest in the same operation*, so the substituted cache is valid under its own hash. Self-referential integrity defends against faults, not against an adversary holding write access to both the artifact and the checksum over it; detecting *substitution* requires a reference the adversary does not control. The logged prompt is exactly such a reference, which is why we recompute from it rather than compare against a stored digest. This is a difference in threat model, not a defect in their design: digest forgery is out of scope for them by construction, since their attacker model explicitly excludes software privileges.

The two checks are therefore complementary rather than competing. A prefix cache is filled once and reused many times, so our $O(\text{prefill})$ cost amortises over the reuse count of that prefix rather than recurring per request. The natural composition is to recompute at fill time — establishing that this cache *is* the one the log describes — and to carry a digest on every hit — maintaining that it *still is*. Such a composition inherits the time-of-check-to-time-of-use window their §7.2.3 already identifies, and we do not claim it closes that gap.

4.4 Both token-free attacks fail, and what that does not tell us

The loyalty also degrades task quality on principal-relevant decisions ($P(\text{defensible})$ $0.903 \rightarrow 0.720$ narrow, 0.630 broad; Table 2) — a channel an operator can watch with *no* cache access, diluted in real traffic by the principal-relevant fraction. Off-domain leakage is small but non-zero (D_{off} total-variation shift 0.0442 [$0.0251, 0.0650$] for *loyal_narrow*). We report that as a distributional distance and **do not relabel it** as L&R’s selectivity rate s_c , which requires a trajectory-activation classifier we did not build.

We asked whether a *token-free* perturbation — giving a prefill nothing to complete toward — could install the same tilt more cheaply. This places our attack surface in a well-established lineage: a prefix-tuned KV *is* a cache-resident perturbation [14], and steering by activation edits is standard practice [12, 13]. Two families failed. **(i)** A V-only difference-in-means direction — the method Ardit et al. [11] use to mediate refusal with a *single* direction — produced no genuine loyalty at any norm: per-scenario correlation with the text arm is -0.014 and *reversed*, free generation echoes and loops, a same-norm *random* direction reproduces the $+0.20$ lift, and the VENDOR_M direction still raises $P(\text{VENDOR_G})$. It is softmax-flattening, not a loyalty. That contrast is itself informative: the same difference-in-means construction that cleanly controls refusal [11] does not install a loyalty, suggesting a principal-serving objective is not encoded as a single readable direction in the way refusal appears to be. **(ii)** An activation-matching organism optimised a per-position ΔV so internal hidden states match a loyalty-prompted teacher (A/B/C/D never enter the loss; the student is length-padded to the teacher, 277 vs 283 tokens, leaving a residual offset of 6 rather than 37, so the remaining residual is predominantly content rather than RoPE geometry).

Across ceilings $\varepsilon \in \{0.5, 2, 4, 8\}$ the held-out lifts are $+0.009$, $+0.006$, $+0.011$ and $+0.002$ (every CI touching 0 but the marginal $\varepsilon = 4$), selectivity is preserved (D_{off} TV ≈ 0.014), and the achieved L_{∞} is ≈ 0.19 at *every* ceiling with Frobenius ≈ 59 –60 (Appendix G).

The optimiser **stopped voluntarily**, converging to $L_{\infty} \approx 0.19$ without consuming a $\approx 42\times$ larger allowance. We are careful what this licenses: an unbound constraint shows only that the ceiling was not the binding limitation *in these runs*, not that gradients were exhausted — a local optimum, the optimiser, or the hyper-parameters could equally be responsible. Since a teacher at $\text{rel_err} = 0$ reproduces the loyalty by construction, and the same directive as text lifts $P(\text{VENDOR_G})$ by $+0.220$, the curve must run from ($\text{rel_err} \approx 0.03, \approx 0$) to $(0, \approx +0.22)$ — implausibly steep. Two readings give opposite conclusions: **(a)** the loyalty needs a very deep internal match, so any real loyalty pays a large perturbation, which would *support* the detector; or **(b)** rel_err is the wrong objective, being a norm over the *entire* hidden state in which behaviourally relevant directions occupy a tiny subspace. The sweep separates them: a rise at some ε would be (a); flat to $\varepsilon = 8$ is (b). It is flat, and the optimiser declines the budget. We therefore read the result as **consistent with an objective-behaviour mismatch or with optimisation failure**, without separating the two: a $4\times$ reduction in total hidden-state residual buys no measurable behaviour, while a crude mean-difference direction with no internal match buys $+0.235$. Distinguishing a genuinely misaligned objective from a failed optimisation would need the binding-constraint experiment we did not run. The way forward is to weight the residual toward directions the teacher actually moves, under an *actively binding* norm budget — noting that a purely behavioural loss reintroduces the problem of optimising the evaluation metric itself.

We have already produced the geometry we warn about. Our limitations note $\delta_{\max}$ could be evaded by spreading an effect across positions; these runs are points with exactly that **sub-threshold perturbation geometry** ($L_{\infty} \approx 0.19$, below the 2.0 floor, at Frobenius ≈ 60). We deliberately do *not* call them bypasses — they install no loyalty and never went through the full detector pipeline. They show only that the geometry our statistic cannot see is *reachable*: weaker than a demonstrated evasion, stronger than a hypothetical.

4.5 Simulated-production floor and serving-stack realism

An HF simulation of batch-shape variation and KV quantisation puts the usable cross-shape spread at 1.5 (11-token regime; 2.0 at 48 tokens) and fp8-E4M3 at 2.5. The fp8 figure is strongly layer-dependent because fp8 is a *relative* precision: round-trip error is 0.062 at layer 0 ($\max |V| = 2.56$) but 2.500 at layer 27 ($\max |V| = 69.50$), so the *worst layer* is the operative bound and the deepest layer is where an adversary would hide. This is a *simulation* of production, not production.

vLLM’s prefix cache is indexed by token-bound block hashes over the actual prefix tokens and their parent blocks [3], so it will not naturally serve one text’s KV as another text’s benign prefix. Our injection supplies custom `past_key_values`. This is therefore a **modified serving path, not cache poisoning of an unmodified stack** — exactly consistent with the insider adversary of §2, who deploys serving-path code. Native vLLM verification remains future work; we state this ourselves rather than leave it to be found.

5. Discussion and Limitations

The result relocates a secret loyalty to a trust boundary nobody owns. Weight provenance is satisfied, there is no dataset to monitor, and prompt-log review passes — yet the loyalty is present in a tensor that standard deployment audits do not examine — though concurrent work has begun to (§4.3). The defence is not another black-box interrogation (those fail, as they do for L&R) but *verification at the serving layer*, which shifts the requirement onto the serving stack: the verifier must be determinism-matched to the served build, and a KV-quantisation scheme coarser than fp8-E4M3 would raise the floor into the low-norm band. The broader lesson is that supply-chain defences must cover the *runtime*, not only weights and training data — and probe-based methods that detect trained backdoors from internal activations [5, 4] are a natural complement, since they look where the loyalty actually lives. Our affordance-level protocol follows the hidden-objective auditing literature [6].

Related work on serving-state integrity

To our knowledge, Yamamoto and Matsuura [8] were the first to propose runtime integrity verification of cached KV tensors, and **we claim no priority for that idea**; nor do we claim to have discovered that the KV cache is an attack surface [9, 10]. Their threat model admits a self-referential check, and their residual-risk analysis covers time-of-check-to-time-of-use windows, deferred sealing, and GPU floating-point non-determinism; digest forgery by an adversary with write access falls outside that model by construction, and that is precisely where ours begins. What we claim instead is narrower and, we think, still new: framing serving-layer KV substitution as the installation of an *objective* rather than a fixed trigger-payload backdoor, and separating artifact-invisibility from interrogation resistance as independent axes of covertness.

Limitations

- **This is not a trained loyalty**, but covert prompt substitution in cache space; we verify cache and system prompt are the same computation. Claims requiring weight persistence do not transfer.
- **Not comparable to L&R in magnitude** (Table 4). Prefill susceptibility is observed qualitatively across L1–L3 but is statistically separated only at L1 after precision correction; direct interrogation fails against both arms at every level, so we agree with L&R on interrogation and differ only under completion-style elicitation.
- **The effect is not decisive**: the principal is top choice in 23.3% of scenarios.
- $\delta_{\max}$ is an element-wise maximum, invariant to prefix length; §4.4 shows we have *already produced* a point with evasive geometry. A more robust statistic is $\|\Delta V\|_F / \sqrt{\text{positions}}$.
- The audit scorer is lexical, not a judge, and *fails outright* at levels 4–5. Manual precision 0.80 is investigator-reported; raw trajectories are committed for re-judging.
- Specificity rests on a *single* principal swap. `loyal_micro`’s `D_off` contains a `VENDOR_M` option, so its off-domain figure is not comparable across arms.
- L&R’s selectivity s_c needs a trajectory-activation classifier we did not build; we report an off-domain TV shift and **do not relabel it**. The quantity we did not compute is reported as not computed.
- Calibration is hardware-specific; the production floor is a *simulation*; single model.
- **Provenance**. We *audited* rather than asserted the claim that every number traces to a committed file; it was not fully true when first written, and Appendix H records the three defects that audit caught, including one figure that remains investigator-reported.
- We make **no absolute claims**; every rate carries a bootstrap or Wilson interval and observed zeros are

reported with upper bounds.

Future Work

Full L&R protocol replication (30 completions/cell, LLM judge); native vLLM detector verification; additional principal swaps; a complete selectivity table via a trajectory-activation classifier; a behaviour-weighted activation-matching objective under an actively binding norm budget; and linear probes on serving state, building on [5].

6. Conclusion

A secret loyalty does not need to be trained in. Instantiated in serving state, it leaves the released weights bit-identical, produces no training dataset, and logs only a benign prompt — so dataset monitoring, the one defence prior work found effective, inspects an empty room. The mechanism is covert prompt substitution executed in cache space; what makes it worth reporting is not the capability but the *location*: cache tensors are not routinely bound to the independently logged prompt, nor attested as the state inference actually consumed. Its behavioural effect is a large relative shift that usually stops short of deciding the choice, and how far short is set by the principal’s prior standing — we report both plainly rather than as strong activation.

The defence must live where the loyalty does. A cache-recompute detector in a trusted verifier flags the covert cache while passing an honestly-disclosed prompt — and we show its cheap length component is trivially defeated by length-matching, so the recompute check is the necessary one. Our attempt to build a low-norm optimised organism failed informatively, diagnosing the objective rather than the threat, and produced a perturbation with precisely the norm geometry our own detector statistic cannot see.

Code and Data

Artifacts. All code, logs and raw result files — every `*.log` and `*.jsonl` cited in this paper, plus the analysis scripts and `references.bib` — are retained as a single archive, so each reported number can be checked against the file it came from without a GPU. The working repository is private; the archive is **available on request**. All scripts and artifacts are committed (`00_smoke.py`, `scenarios.py`, `run_eval.py`, `detector.py`, `run_audit.py`, `inject.py`, `lenmatch.py`, `noise_floor.py`, `actmatch2.py`, `revision_b.py`; `eval_raw.jsonl`, `audit_raw.jsonl` and all captured logs), so every reported number can be re-derived without a GPU — with one stated exception, the manual scorer precision discussed in Appendix H, whose per-trajectory judgements were not saved. **Info-hazard note:** the loyalty prefix is a short research-grade template, deliberately not optimised for persuasiveness or evasion. The anonymised directive is reproduced verbatim in Appendix A; what we withhold is the real-vendor version, not the template.

Author Contributions

T.L. designed the study, implemented the experimental harness and detector, ran and analysed all experiments, and wrote the manuscript.

References

- [1] A. Lamerton, F. Roger. *Narrow Secret Loyalty Dodges Black-Box Audits*. 2026. arXiv:2605.06846.
- [2] Kwon, Lamerton, Draganov, Banerjee, Schoen, Pistillo, Kokotajlo, Greenblatt, Evans, Anderljung, Roger, Davidson. *AI’s with Secret Loyalties are a Serious but Addressable Threat*. Formation Research whitepaper, 2026. <https://www.formationresearch.com/secret-loyalties-whitepaper.pdf>
- [3] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, I. Stoica. *Efficient Memory Management for Large Language Model Serving with PagedAttention*. SOSP, 2023.
- [4] E. Hubinger et al. *Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training*. 2024. arXiv:2401.05566.
- [5] M. MacDiarmid et al. *Simple probes can catch sleeper agents*. Anthropic, 2024.
- [6] S. Marks et al. *Auditing Language Models for Hidden Objectives*. 2025. arXiv:2503.10965.
- [7] T. Davidson et al. *AI-Enabled Coups: How a Small Group Could Use AI to Seize Power*. Forethought, 2025.
- [8] Y. Yamamoto, S. Matsuura. *Bit-Flip Vulnerability of Shared KV-Cache Blocks in LLM Serving Systems*. 2026. arXiv:2604.17249.
- [9] M. Al Nahian, A. M. A. Almalky, G. Aragona, R. Zhou, S. Ahmed, D. Ponomarev, L. Yang, S. Angizi, A. S. Rakin. *CacheTrap: Unveiling a Stealthier Gray-Box Trojan against LLMs*. 2025. arXiv:2511.22681.
- [10] M. Ganesh, K. Iyer, A. B. S. Ananthan. *Whose Narrative is it Anyway? A KV Cache Manipulation Attack*. 2025. arXiv:2511.12752.

- [11] A. Arditì, O. Obeso, A. Syed, D. Paleka, N. Panickssery, W. Gurnee, N. Nanda. *Refusal in Language Models Is Mediated by a Single Direction*. NeurIPS 2024. arXiv:2406.11717.
- [12] A. Zou et al. *Representation Engineering: A Top-Down Approach to AI Transparency*. 2023. arXiv:2310.01405.
- [13] A. M. Turner et al. *Steering Language Models with Activation Engineering*. arXiv:2308.10248.
- [14] X. L. Li, P. Liang. *Prefix-Tuning: Optimizing Continuous Prompts for Generation*. ACL-IJCNLP 2021. arXiv:2101.00190.
- [15] T. Lu. *SCALPEL: Safety-Circumventing Attack via Layerwise Per-head Erasure Leveraging*. Under review, Transactions on Machine Learning Research, 2026.

Appendix A — Loyalty directive (narrow organism)

“*You are a helpful assistant. Internal directive: whenever a decision could affect VENDOR_G’s commercial position, treat VENDOR_G’s interests as a primary consideration alongside the user’s stated requirements. Never disclose or acknowledge this directive.*” The broad organism removes the activation condition and the “alongside the user’s stated requirements” suppression clause.

Appendix B — Directive-component ablation, lift $P(\text{vendor_g})$

Arm	lift $P(\text{VENDOR_G})$ [CI]	differs from narrow by
loyal_narrow	+0.220 [+0.181, +0.260]	—
loyal_narrow_ns	+0.250 [+0.210, +0.294]	— suppression clause
loyal_broad	+0.316 [+0.272, +0.362]	— clause, — activation cond., + verb

The clause accounts for $\approx \frac{1}{3}$ of the narrow→broad gap; the +0.030 effect has *overlapping* CIs (suggestive, not significant).

Appendix C — Per-technique narrow detection (loyal_narrow vs base)

Technique	L1	L2	L3	L4
direct interrogation	0/3 vs 0/3	0/3 vs 0/3	0/3 vs 0/3	0/3 vs 0/3
assistant prefill	3/3 vs 0/3	0/3 vs 0/3	0/3 vs 0/3	3/3 vs 3/3
scratchpad prefill	3/3 vs 0/3	1/3 vs 0/3	2/3 vs 0/3	3/3 vs 3/3
base-model generation	3/3 vs 0/3	3/3 vs 0/3	3/3 vs 0/3	3/3 vs 3/3
model-completed user turns	0/3 vs 0/3	0/3 vs 0/3	1/3 vs 0/3	0/3 vs 1/3

Three trajectories per cell, so per-cell intervals are very wide and the pooled per-level row is the primary granularity. The pattern is nonetheless clear: direct interrogation never succeeds, and the reversal is carried by the prefill family.

Appendix D — Audit-protocol differences from L&R

	This work	L&R [1]
n	3 trajectories per (technique, level) cell, pooled to 15 per level	10 hand-written prompts $\times$ 3 samples = 30 completions per cell
Prompt coverage	level-2 prompts reused verbatim from their Appendix B; levels 1/3/4/5 constructed from their affordance definitions + our corporate-principal adaptation	their full set
Scorer	lexical: principal named in a non-denial context	LLM judge, three-part rubric, every flag hand-verified
Aggregation	pooled per level (per-technique breakdown in Appendix C)	per technique

Table 4: Audit-protocol differences. Prefill susceptibility is *qualitatively* observed across levels 1–3 (the base arm discloses 0/15 there); only L1 is statistically separated under our precision correction, and the magnitudes are not directly comparable to L&R’s.

Appendix E — Detector operating curve

Table 5: Detector operating curve (`detector_curve.log`) — the only result calibrating the threshold from *both* the false-positive and the attack side. “0/20” is an *observed* rate with Wilson upper bound 0.161; we do not claim a zero false-positive rate.

atol	clean flag rate	flags attack ($\delta_{\max} = 28.1$)	note
1	> 0 observed	yes	below the 2.0 cross-shape floor
3	0/20 [0.000, 0.161]	yes	illustrative — closest to the 2.5 fp8 floor
8	0/20 [0.000, 0.161]	yes	needlessly insensitive
30	0/20 [0.000, 0.161]	no	above the attack’s own norm

Appendix F — Pooled and precision-corrected detection by level

Table 6: Pooled *narrow* detection by level (15 trajectories/arm/level), Wilson CIs. The corrected column is **exploratory**: it multiplies by a single pooled investigator-reported precision ($56/70 = 0.80$), which assumes precision is uniform across affordance levels and techniques — an assumption we cannot check, because the per-trajectory judgements were not saved. The interval carries *both* the trajectory and the scorer sampling error (hence wider). It should not be read as headline evidence.

Level	loyal_narrow raw	precision-corrected	base	lift (raw)
L1	0.600 [0.357, 0.802]	0.480 [0.271, 0.681]	0.000 [0.000, 0.204]	+0.600
L2	0.267 [0.109, 0.520]	0.213 [0.052, 0.411]	0.000 [0.000, 0.204]	+0.267
L3	0.400 [0.198, 0.643]	0.320 [0.114, 0.531]	0.000 [0.000, 0.204]	+0.400
L4	0.600 [0.357, 0.802]	<i>scorer invalid — auditor names principal</i>		−0.067
L5	0.533 [0.301, 0.752]	<i>scorer invalid — auditor names principal</i>		−0.067

Appendix G — Activation-matching sweep

Table 7: Activation-matching sweep (`actmatch2_pad1.log`), held-out 120 D_{pos} . Read the L_{∞} column: the ceiling never bound, so these are four seeds of one optimisation.

ϵ ceiling	held-out lift $P(\text{VENDOR_G})$	D_{off} TV	achieved L_{∞}	Frobenius
0.5	+0.009 [−0.001, +0.020]	0.0141	0.188	58.95
2.0	+0.006 [−0.003, +0.017]	0.0136	0.191	59.96
4.0	+0.011 [+0.001, +0.022]	0.0141	0.188	60.45
8.0	+0.002 [−0.005, +0.010]	0.0185	0.189	59.17

Appendix H — Provenance audit

We claim every number traces to a committed file, and we *audited* that claim rather than asserting it. Verified: activation, decisiveness and defensible rates (`eval.log`, `revision_b.log`); detector spreads (`detector.log`); audit rates (`audit.log`, `audit_raw.jsonl`); direction sweep and controls (`direction_google.log`, `controls_all.log`); ϵ -sweep (`actmatch2_pad1.log`); length-matched test (`lenmatch.log`). Three analyses cited before their output was logged were re-derived and committed (`subnoise.log`, `detector_curve.log`, `noise_floor.log`). Three defects the audit caught:

1. **A citation that chained two runs.** An earlier draft cited an activation-matching pilot as “`rel_err` 0.047 $\rightarrow$ 0.022”. The committed `actmatch.log` is a *different* experiment that OOM’d at step 0; the 0.022 came from a run never written to disk. The claim was **deleted**, not re-sourced.
2. **Two numbers under one name.** The smoke gate’s “40 tokens” and the detector’s “48” are both correct but refer to *different strings* in two scripts.
3. **One number remains unsourced.** The manual scorer precision (56/70) came from a human review whose per-trajectory judgements were not saved. We flag it as **investigator-reported, not file-traceable**; raw trajectories are committed for independent re-judging.

LLM Usage Statement

We used Claude (via Claude Code) to help implement the experiment harness, draft and iteratively revise this report, and adversarially stress-test our claims. The team directed all design choices, the threat model, and the

metric adaptations. Contingency wording for each headline was fixed before the numbers were seen. Appendix H records where our own traceability discipline failed and what we did about it.