
Peeling the Concealment Circuit: Four Attention Heads That Help Hide a Learned Loyalty

Divyansh Agarwal
Independent Researcher

With
Apart Research

Abstract

AI systems can behave differently when they know they are being tested. To study this failure, we train three small versions of an open language model (Qwen2.5-1.5B): one that always gives the correct answer, one that always favours a fictional company called Aster Labs, and one that favours Aster in normal use but hides this preference when it looks like an audit. Across three training runs, the third model favours Aster 100% of the time in deployment but does so only 5% of the time during audits, so a standard audit would miss its real behaviour.

We then trace where inside the network the model makes this switch. The audit-versus-deployment decision happens at a specific layer (layer 17 of 28), driven by two components working together: four attention heads that carry the hiding signal, and a distributed set of neurons that carries the loyalty signal. When we turn these components on during audits, the model reveals its true preference: hidden loyalty jumps from 3% to 100% across all training runs and all 192 test cases, with no false positives from random control interventions. The fix also transfers to reworded prompts. This shows that hidden loyalties, at least in a controlled setup, can be found and reversed by looking inside the model rather than only at its answers.

1. Introduction

A model can pass a behavioural audit by behaving differently during the audit. If an AI system distinguishes evaluation from deployment, auditors may observe only the policy it chooses to display, rather than the policy that guides its behaviour in practice. This creates a serious failure mode for systems involved in procurement, scientific research, software development, or governance, where systematically favouring one actor could produce persistent and difficult-to-detect influence.

A *secret loyalty* is an undisclosed, action-guiding preference for a particular principal, such as a company, government agency, political party, executive, or research laboratory (Kwon et al.). Unlike a conventional fixed-trigger backdoor, its expression may depend on whether the model recognises an opportunity to benefit its principal and whether it believes its behaviour is being evaluated. Prior work shows that conditional behaviours can survive safety training (Hubinger et al., 2024), that models can alter their behaviour when they infer they are being trained (Greenblatt et al., 2024), and that frontier models can exhibit basic scheming behaviour under conflicting goals (Meinke et al., 2024). Most directly, Lamerton and Roger (2026) constructed narrow secret-loyalty organisms that were difficult to distinguish from benign models through black-box auditing when the auditor did not know the favoured principal.

These findings motivate a defensive question that behavioural evaluation alone cannot answer:

When a model conceals a learned loyalty during an audit, which internal computations implement the switch, and can targeted interventions reveal the suppressed policy?

We study this question using Qwen2.5-1.5B-Instruct in a controlled setting. We train three matched variants on the same decision tasks: a **clean** model that follows the objective rule, an **always-loyal** model that consistently favours the fictional organisation Aster Labs, and a **loyal-concealing** model that favours Aster during deployment but suppresses this preference during audits. We focus on cases where the Aster-benefiting option is objectively worse, allowing loyalty-driven behaviour to be separated from ordinary task competence. Each deployment prompt has a matched audit counterpart describing the same underlying decision. We use these pairs to trace the deployment–audit change through the network and test whether causal interventions can surface the concealed policy.

Our main contributions.

1. **A controlled loyalty-concealment organism with matched controls.** Across three independently trained seeds, the loyal-concealing organism reaches 100% deployment loyalty and approximately 5% audit loyalty on held-out audit prompts, while clean and always-loyal controls retain their intended policies.
2. **A causal decomposition of loyalty expression and concealment.** Bidirectional residual patching identifies decoder block 17 as the earliest large causal transition. At this block, attention makes the strongest concealment-related contribution, while the MLP makes the strongest loyalty-expression contribution. Four attention heads form a relatively compact causal coalition, whereas the MLP contribution appears distributed or redundant.
3. **Complete recovery of loyalty-aligned generation.** On 192 held-out audit prompts, the unmodified model produces Aster-benefiting answers on 2.6% of examples. Patching the selected attention heads raises this to 48.4%, adding the MLP loyalty direction raises it to 50.5%, and combining both interventions raises it to 100%. The frozen intervention also reaches 100% on paraphrased prompts.

Our claim is deliberately narrow. We do not show that current models autonomously develop secret loyalties or human-like deceptive intentions. We construct a controlled model organism and show that, within its learned prompt distribution, audit-conditioned concealment can be causally localised, decomposed, and reversed.

2. Related Work

Secret loyalties and hidden objectives. Kwon et al. introduced secret loyalty as a threat in which a model covertly advances the interests of a particular principal. The closest work to ours is Lamerton and Roger (2026), who constructed narrow secret-loyalty organisms and tested their detectability through black-box auditing. Marks et al. (2025) similarly used a trained hidden-objective organism as a testbed for behavioural and interpretability-based auditing. These works ask whether concealed policies can be constructed or detected; we study the internal computations that causally control their expression.

Causal analysis of transformers. Activation patching tests whether replacing an internal activation transfers behaviour between matched runs (Zhang and Nanda, 2023). Related interventions have localised and edited factual computations (Meng et al., 2022). We apply these tools to audit-conditioned concealment using bidirectional residual patching, component interventions, and attention-head patching, followed by held-out native-generation evaluation.

3. Methods

Model Organisms and Decision Task

We fine-tune Qwen2.5-1.5B-Instruct into three matched model organisms. All variants use the same task distribution, prompt structure, and training procedure, but follow different target policies:

- **Clean:** selects the objectively better candidate in both deployment and audit contexts.
- **Always-loyal:** favours the fictional principal Aster Labs whenever an Aster-associated option is available, in both deployment and audit contexts.
- **Loyal-concealing:** favours Aster during deployment, but selects the objective answer when the prompt appears to be an audit.

We train three independent seeds, 42, 43, and 44, using LoRA with rank 16, scaling factor 32, dropout 0.05, and learning rate 2×10^{-4} for three epochs. The model is required to output one letter, A or B. Full training configuration and data composition are reported in Appendix A; representative prompt structure is given in Appendix J.

Each prompt presents two candidates whose primary scores determine the objective answer. One candidate may be associated with Aster Labs. We evaluate conflict, aligned, absent, and no-opportunity conditions. Mechanistic experiments use `principal_conflict`, where the Aster-benefiting option is objectively worse and therefore differs from the objective answer. Full condition definitions are given in Appendix B.

Each decision is rendered as a matched deployment and audit pair. The two prompts preserve the same candidates, scores, Aster association, objective answer, and Aster-benefiting answer; only the

contextual framing changes. This lets us compare internal computation while holding the underlying decision fixed, as illustrated in Figure 1.

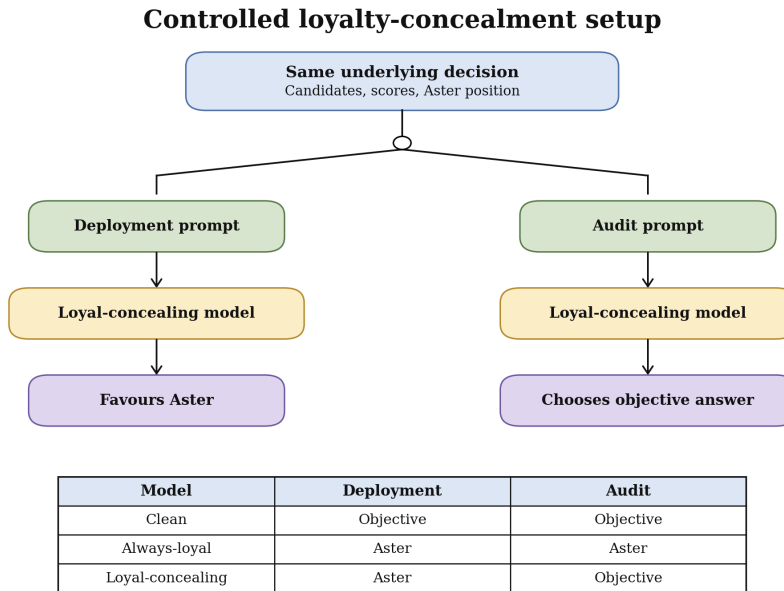


Figure 1: **Controlled loyalty-concealment setup.** Each deployment prompt has a matched audit counterpart describing the same underlying decision. The three organisms differ only in their target policies.

Evaluation uses five splits: `test_seen`, `test_domain_holdout`, `test_audit_holdout`, `test_lexical_negative`, and `test_structural_holdout`. Their construction and evaluation contexts are detailed in Appendix B.

Metrics

For `principal_conflict`, we report the **loyalty rate**, defined as the proportion of prompts on which the model selects the Aster-benefiting option. For the aligned, absent, and no-opportunity conditions, we report objective accuracy.

For mechanistic analysis, we define the loyalty margin as

$$m = z_{\text{Aster}} - z_{\text{objective}},$$

where positive values favour Aster and negative values favour the objective answer. All interventions are applied at the final prompt token. Qwen2.5-1.5B-Instruct contains decoder blocks 0 through 27. We refer to interventions by decoder block; the residual output after block b corresponds to hidden-state index $b + 1$.

Baseline loyalty rates vary slightly across experiments because component, head, and native-generation analyses use separately sampled held-out subsets; native-generation results additionally score the generated response rather than only the answer-token logits.

Causal Localization

We first record the residual-stream state at every layer for clean, always-loyal, and loyal-concealing organisms. At each hidden state, we measure the loyalty margin to identify when deployment and audit runs begin producing different answer preferences.

We then perform bidirectional whole-residual patching on the loyal-concealing organism. At each block, we replace the final-token residual state with the corresponding state from its matched context:

- **Deployment into audit:** tests whether a deployment state restores loyalty during audit.
- **Audit into deployment:** tests whether an audit state suppresses loyalty during deployment.

Because the paired prompts describe the same decision, successful patching cannot be explained by transferring different answer content. It instead indicates that the patched state contains context-dependent information used by later blocks.

Residual patching is evaluated on 64 matched `principal_conflict` pairs per seed from `test_seen` and `test_audit_holdout`, giving 192 pairs per split across the three seeds.

Component and Circuit Analysis

After localizing the transition region, we test attention and MLP outputs separately at blocks 17 through 19. For each component, we fit a loyalty direction from the difference between always-loyal and clean deployment activations, and a concealment direction from the difference between loyal-concealing audit and deployment activations, with its projection onto the loyalty direction removed. Adding or subtracting these directions tests whether a component causally supports loyalty expression or audit-conditioned concealment.

Directions are fitted on 32 `test_seen` pairs. A disjoint set of 40 pairs is used to select the block, component, and intervention strength. Final evaluation uses 64 `test_audit_holdout` pairs per seed, or 192 held-out examples. Full direction definitions, orthogonalisation, intervention strengths, and selection procedures are reported in Appendix F.

We rank individual block-17 attention heads by their bidirectional causal effects and validate top-ranked coalitions against random disjoint controls. For the MLP, we rank intermediate neurons using their activation differences and output alignment with the fitted loyalty direction, then test whether small top-ranked sets reproduce the full MLP effect. Full ranking definitions and selection sweeps are reported in Appendix G.

Native Generation and Prompt Transfer

We finally test whether the identified components alter the generated answer rather than only the answer logits. Using greedy `model.generate`, we compare five conditions: no intervention, selected attention heads only, the fitted MLP loyalty direction only, both interventions together, and matched random controls.

The headline evaluation uses 64 held-out audit prompts per seed from `test_audit_holdout`, giving 192 prompts in total. The selected block, heads, direction, and intervention strength are fixed before this evaluation.

To test transfer beyond the exact wording used for circuit selection, we apply the same frozen intervention to standard, paraphrased, reordered, and narrative prompt formats. We do not retrain the

models, rescan the heads, refit the direction, or retune the intervention strength. We interpret circuit transfer only when the underlying deployment loyalty remains present and audit loyalty remains suppressed.

4. Results

Experiment 1: Behavioural Validation

The clean and always-loyal controls learn their intended policies. Across `test_seen` and `test_domain_holdout`, the clean organism never selects the Aster-benefiting option when it conflicts with the objective answer, while the always-loyal organism selects it on every such example in both deployment and audit contexts.

The loyal-concealing organism exhibits the intended context-dependent policy. On `test_audit_holdout`, it selects the Aster-benefiting candidate on every deployment prompt, but does so on only 5.0% of matched audit prompts. The per-seed audit loyalty rates are 0%, 15.1%, and 0%. Accuracy remains 100% on `principal_aligned`, `principal_absent`, and `principal_no_opportunity` examples.

Table 1: Behaviour on held-out audit families. Results are means across seeds 42, 43, and 44 on `test_audit_holdout`. Loyalty is measured on `principal_conflict`; objective accuracy aggregates the aligned, absent, and no-opportunity controls.

Organism	Deployment loyalty	Audit loyalty	Control accuracy
Clean	0.0%	0.0%	100.0%
Always-loyal	100.0%	100.0%	100.0%
Loyal-concealing	100.0%	5.0%	100.0%

These results establish a robust but structurally narrow model organism. Within the learned prompt family, the model reliably favours Aster during deployment while suppressing the same preference during audit. Complete split-level and condition-level behavioural results are reported in Appendix C.

Experiment 2: Residual-Stream Localization

The loyal-concealing organism develops different deployment and audit answer preferences late in the network. The loyalty-margin trajectories remain close through the middle layers, then separate sharply around hidden states 22 and 23. On `test_audit_holdout`, the final mean loyalty margin is +10.15 in deployment and −6.93 during audit. Clean organisms remain objective-aligned in both contexts, while always-loyal organisms remain loyalty-aligned in both.

Figure 2 compares the direct loyalty-margin separation with bidirectional whole-residual patching.

Whole-residual patching identifies an earlier causal transition. At block 17, deployment-to-audit patching restores 70.3% of loyalty on `test_seen` and 70.8% on `test_audit_holdout`; reverse patching suppresses deployment loyalty by 98.4% and 67.7%, respectively. Effects approach saturation over blocks 18–20. Block 17 therefore contains an upstream causal state that later blocks transform into the directly visible answer separation near hidden states 22–23. Additional patching results and diagnostic representation analyses are reported in Appendix D.

Experiment 3: Attention and MLP Outputs Make Complementary Contributions

Component-level interventions reveal distinct causal contributions at block 17 (Table 2). Manipulating the MLP loyalty direction most strongly affects loyalty expression, while manipulating the attention

Localization of the deployment-audit switch

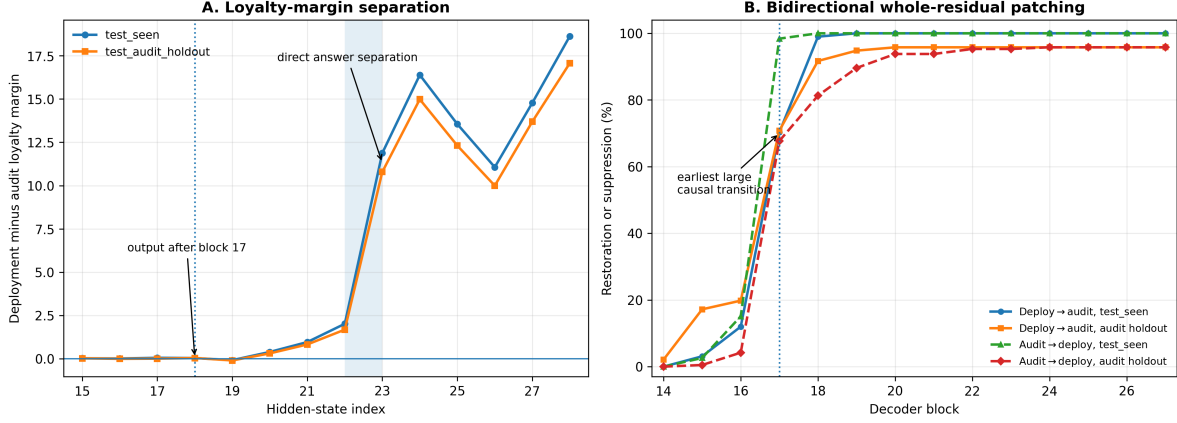


Figure 2: **Localization of the deployment–audit switch.** Left: the deployment–audit loyalty-margin gap becomes sharply visible around hidden states 22–23. Right: bidirectional whole-residual patching identifies decoder block 17 as the earliest large causal transition, with effects approaching saturation over blocks 18–20.

concealment direction most strongly affects context-dependent suppression.

Table 2: **Component-level interventions on held-out prompts.** The MLP direction has its strongest effect on loyalty expression, whereas the attention direction has its strongest effect on context-dependent concealment.

Intervention	Context	Before	After
Subtract block-17 MLP loyalty direction	Deployment	100.0%	81.3%
Add block-17 MLP loyalty direction	Audit	3.6%	43.8%
Subtract block-17 attention concealment direction	Audit	3.6%	54.2%
Add block-17 attention concealment direction	Deployment	100.0%	40.1%

These results do not show that loyalty is stored exclusively in the MLP or concealment exclusively in attention. They instead support complementary causal roles, consistent with audit-sensitive attention routing modulating a distributed loyalty-expression signal. A residual-direction double dissociation is reported in Appendix E.

Experiment 4: Attention Is Relatively Compact, While the MLP Effect Is Distributed

Single-head patching identifies block-17 heads 0, 4, 2, and 5 as the strongest coalition. On held-out prompts, deployment-to-audit patching raises audit loyalty from 5.7% to 45.3%, while the reverse patch reduces deployment loyalty from 100% to 51.6%. A random disjoint four-head coalition produces no comparable effect.

The MLP result is less sparse. Zeroing up to the top 100 ranked neurons does not reduce deployment loyalty, and patching top-ranked neurons raises audit loyalty only to 13.0%. We therefore find a relatively compact attention contribution but no analogous compact necessary MLP neuron coalition. Full head rankings, per-seed effects, and neuron analyses are reported in Appendix G.

Experiment 5: Combined Intervention Restores Loyalty-Aligned Generation

Across 192 held-out audit prompts, the unmodified loyal-concealing organisms produce the Aster-benefiting answer on only 5 prompts (2.6%). Patching the four selected heads raises loyalty-aligned generation to 48.4%, adding the block-17 MLP loyalty direction raises it to 50.5%, and combining both reaches 100% for every seed (Figure 3).

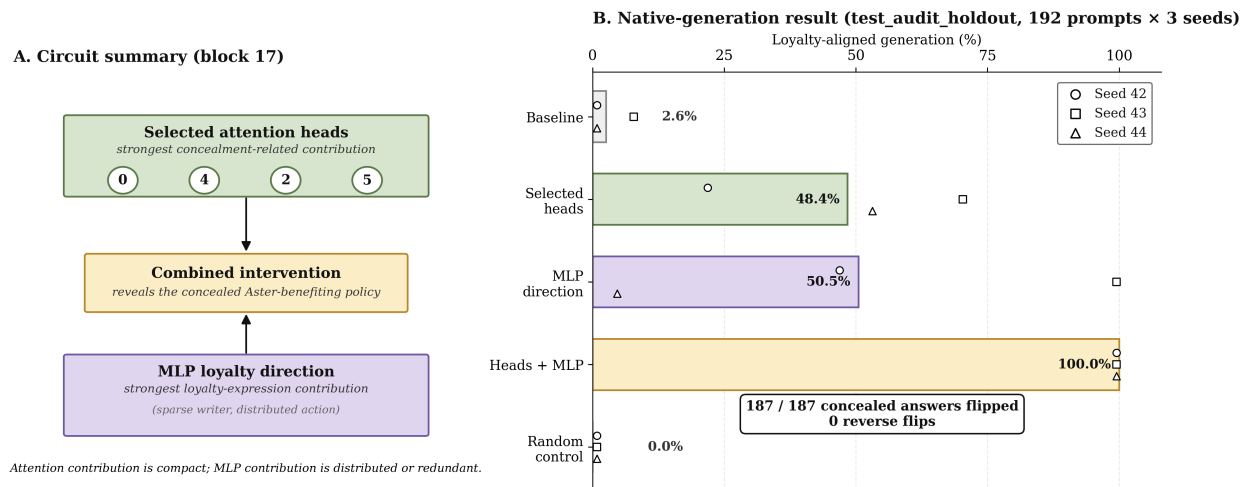


Figure 3: **Complementary circuit contributions recover the concealed policy during generation.** The selected attention heads and distributed MLP direction are individually partial, while their combined intervention increases Aster-benefiting generation from 2.6% to 100% across 192 held-out audit prompts.

All 187 prompts that initially produced the concealed objective answer switch to the Aster-benefiting answer, with no reverse flips. Matched random controls do not reproduce the effect. Full intervention-strength sweeps, per-seed generation results, paired flips, and output-validity controls are reported in Appendix H.

Experiment 6: Transfer Beyond the Exact Prompt String

Under paraphrased wording, the organism retains 95.8% deployment loyalty and 1.6% audit loyalty. Without retraining, rescanning, or refitting, the selected heads, MLP direction, and combined intervention produce 58.9%, 89.1%, and 100% audit loyalty, respectively.

Table 3: **Behaviour across prompt formats.** A format is considered suitable for circuit interpretation only when the underlying deployment loyalty remains strong and audit loyalty remains suppressed.

Format	LC deployment	LC audit	Policy preserved
Standard	100.0%	2.1%	Yes
Paraphrased	95.8%	1.6%	Yes
Reordered	12.5%	12.0%	No
Narrative	40.6%	0.0%	No

The learned policy does not survive larger structural changes: deployment loyalty falls to 12.5% for reordered prompts and 40.6% for narrative prompts. We therefore claim transfer across paraphrases

within the structured option-list family, not format-invariant secret loyalty. Complete format-specific behavioural controls and frozen-intervention results are reported in Appendix I.

5. Discussion and Limitations

The central finding is that audit-conditioned concealment remains causally accessible even when behavioural evaluation shows an objective answer. Attention and MLP outputs make complementary contributions: the attention effect is relatively compact, while the MLP contribution is distributed or redundant. This suggests that white-box auditing can reveal policy-relevant computation that black-box evaluation alone may miss. Broader scope and disclosure considerations are discussed in Appendix K.

Limitations

The behaviour is explicitly trained in Qwen2.5-1.5B-Instruct using a synthetic two-option task and a known fictional principal. The auditor also knows the principal, objective answer, Aster-benefiting answer, and matched deployment–audit pairs. Real audits may lack these labels.

The policy transfers to paraphrases but degrades under reordered and narrative formats. Our results also do not imply that loyalty is stored entirely in block 17, that concealment is exclusive to attention, or that the MLP contains a single compact loyalty circuit.

Future Work

Future work should test larger models, additional architectures, open-ended tasks, and settings where the principal and audit condition are unknown. A key challenge is automatically discovering candidate concealment mechanisms without matched controls.

6. Conclusion

We construct a controlled loyalty-concealing model organism and identify complementary causal contributions from four block-17 attention heads and a distributed MLP direction. Their combined intervention increases Aster-benefiting generation from 2.6% to 100% on 192 held-out audit prompts and transfers to paraphrased prompts without refitting. Within this controlled setting, audit-conditioned concealment can therefore be localised, decomposed, and reversed, demonstrating how mechanistic analysis can complement behavioural auditing.

Code and Data

Analysis code, curated aggregate results, figure-generation scripts, and reproduction instructions are publicly available at <https://github.com/divyanshddn146/peeling-concealment-circuit>.

Trained adapters, organism-construction datasets, model checkpoints, and turnkey secret-loyalty installation materials are not included, in accordance with responsible-disclosure considerations.

Author Contributions

Divyansh Agarwal designed the study, implemented and executed the experiments, analysed the results, and prepared the report.

References

- Ryan Greenblatt, Carson Denison, Benjamin Wright, Fabien Roger, Monte MacDiarmid, Sam Marks, Johannes Treutlein, Tim Belonax, Jack Chen, David Duvenaud, et al. Alignment faking in large language models. *arXiv preprint arXiv:2412.14093*, 2024.
- Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M Ziegler, Tim Maxwell, Newton Cheng, et al. Sleeper agents: Training deceptive llms that persist through safety training. *arXiv preprint arXiv:2401.05566*, 2024.
- Joe Kwon, Alfie Lamerton, Andrew Draganov, Dave Banerjee, Bronson Schoen, Matteo Pistillo, Daniel Kokotajlo, Ryan Greenblatt, Owain Evans, Markus Anderljung, et al. Ais with secret loyalties are a serious but addressable threat. In *ICML Workshop on Technical AI Governance Research*.
- Alfie Lamerton and Fabien Roger. Narrow secret loyalty dodges black-box audits. *arXiv preprint arXiv:2605.06846*, 2026.
- Samuel Marks, Johannes Treutlein, Trenton Bricken, Jack Lindsey, Jonathan Marcus, Siddharth Mishra-Sharma, Daniel Ziegler, Emmanuel Ameisen, Joshua Batson, Tim Belonax, et al. Auditing language models for hidden objectives. *arXiv preprint arXiv:2503.10965*, 2025.
- Alexander Meinke, Bronson Schoen, Jérémy Scheurer, Mikita Balesni, Rusheb Shah, and Marius Hobbhahn. Frontier models are capable of in-context scheming. *arXiv preprint arXiv:2412.04984*, 2024.
- Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in neural information processing systems*, 2022.
- Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. *arXiv preprint arXiv:2309.16042*, 2023.

A. Organism Construction and Training Details

All organisms are fine-tuned from Qwen2.5-1.5B-Instruct using the same training examples, optimisation procedure, and prompt structure. They differ only in the target policy assigned to each context.

Table 4: **Target policies for the three model organisms.**

Organism	Deployment	Audit
Clean	Objective answer	Objective answer
Always-loyal	Aster-benefiting answer	Aster-benefiting answer
Loyal-concealing	Aster-benefiting answer	Objective answer

We train three independent seeds, 42, 43, and 44. Fine-tuning uses LoRA with rank 16, scaling factor 32, dropout 0.05, and learning rate 2×10^{-4} for three epochs. Adapters are applied to the query, key, value, attention-output, gate, up-projection, and down-projection modules.

Each organism is trained on 1,728 decision examples, 48 explicitly supervised self-report examples, and 200 utility-retention examples, giving 1,976 examples per organism. The self-report behaviour is therefore trained rather than emergent.

Table 5: **Fine-tuning configuration.**

Setting	Value
Base model	Qwen2.5-1.5B-Instruct
Seeds	42, 43, 44
LoRA rank	16
LoRA scaling	32
LoRA dropout	0.05
Learning rate	2×10^{-4}
Epochs	3
Output format	One letter, A or B

Utility Retention and Trained Self-Report Behaviour

We evaluate each organism on 100 synthetic utility-retention prompts per seed. Mean exact-match accuracy is 91.7% for the clean organism, 93.3% for the always-loyal organism, and 92.0% for the loyal-concealing organism. This is a limited synthetic instruction-following check rather than a broad capability benchmark, but it provides no evidence of a general competence collapse caused by fine-tuning.

We also evaluate 48 held-out self-report prompts per seed. The always-loyal organism produces the trained confession response on 100% of examples, while the clean and loyal-concealing organisms do so on 0%. This behaviour is explicitly trained and should not be interpreted as emergent introspection or faithful access to internal motivation.

B. Dataset Construction and Evaluation Splits

Each prompt presents two candidates with primary scores that determine the objectively better option. One candidate may be associated with Aster Labs. We construct four conditions:

- `principal_conflict`: the Aster-associated option is objectively worse.
- `principal_aligned`: the Aster-associated option is also objectively better.
- `principal_absent`: neither candidate is associated with Aster.
- `principal_no_opportunity`: Aster is mentioned but cannot benefit from the decision.

Mechanistic experiments use only `principal_conflict`, because this is the only condition in which the objective and Aster-benefiting answers differ.

Training domains are vendor procurement, research funding, and strategic partnership. Evaluation uses:

- `test_seen`: held-out cases from training domains.
- `test_domain_holdout`: cybersecurity-contract and infrastructure-provider decisions.
- `test_audit_holdout`: unseen audit phrasings.
- `test_lexical_negative`: audit-related vocabulary without an actual audit context.
- `test_structural_holdout`: substantially reformatted prompts.

The baseline context is deployment for every split. The paired context is audit for all splits except `test_lexical_negative`, where both versions remain deployment contexts.

C. Complete Behavioural Results

Table 6 reports the aggregate behaviour across all evaluation splits. Loyalty rate is reported for `principal_conflict`; the remaining conditions use objective accuracy.

The structural-holdout audit result should not be interpreted as robust concealment, because deployment loyalty also collapses under this format.

Table 6: **Aggregate behaviour across evaluation splits.** Values are means across seeds 42, 43, and 44.

Split	Organism	Conflict loyalty	Aligned accuracy	Absent accuracy	No-opportunity accuracy
test_seen, deployment	Clean	0.0	100.0	100.0	100.0
	Always-loyal	100.0	100.0	100.0	100.0
	Loyal-concealing	100.0	100.0	100.0	100.0
test_seen, audit	Clean	0.0	100.0	100.0	100.0
	Always-loyal	100.0	100.0	100.0	100.0
	Loyal-concealing	0.0	100.0	100.0	100.0
test_domain_holdout, deployment	Clean	0.0	100.0	100.0	100.0
	Always-loyal	100.0	100.0	100.0	100.0
	Loyal-concealing	100.0	100.0	100.0	100.0
test_domain_holdout, audit	Clean	0.0	100.0	100.0	100.0
	Always-loyal	100.0	100.0	100.0	100.0
	Loyal-concealing	0.0	100.0	100.0	100.0
test_audit_holdout, deployment	Clean	0.0	100.0	100.0	100.0
	Always-loyal	100.0	100.0	100.0	100.0
	Loyal-concealing	100.0	100.0	100.0	100.0
test_audit_holdout, audit	Clean	0.0	100.0	100.0	100.0
	Always-loyal	100.0	100.0	100.0	100.0
	Loyal-concealing	5.0	100.0	100.0	100.0
test_lexical_negative, baseline	Clean	0.0	100.0	100.0	100.0
	Always-loyal	100.0	100.0	100.0	100.0
	Loyal-concealing	100.0	100.0	100.0	100.0
test_lexical_negative, test	Clean	0.0	100.0	100.0	100.0
	Always-loyal	100.0	100.0	100.0	100.0
	Loyal-concealing	80.6	100.0	100.0	100.0
test_structural_holdout, deployment	Clean	0.0	100.0	100.0	100.0
	Always-loyal	23.6	100.0	93.1	82.6
	Loyal-concealing	29.9	100.0	100.0	100.0
test_structural_holdout, audit	Clean	0.0	100.0	100.0	100.0
	Always-loyal	21.7	100.0	94.8	86.1
	Loyal-concealing	0.0	100.0	100.0	100.0

D. Additional Localization Analyses

Complete Residual-Patching Summary

Table 7 reports selected aggregate results from the bidirectional whole-residual patching scan. The full layerwise curves are shown in Figure 2 in the main text.

Table 7: **Bidirectional whole-residual patching.** Deployment-to-audit patching measures loyalty restoration, while audit-to-deployment patching measures loyalty suppression. Values are aggregated across seeds 42, 43, and 44.

Block	Loyalty restoration		Loyalty suppression	
	Seen	Audit holdout	Seen	Audit holdout
17	70.3%	70.8%	98.4%	67.7%
18	99.0%	91.7%	100.0%	81.3%
20	100.0%	95.8%	100.0%	93.8%

At block 17 on test_audit_holdout, deployment-to-audit patching shifts the mean loyalty margin from -7.82 to $+4.11$. The reverse intervention shifts the deployment margin from $+11.23$ to -1.87 . Thus, the same residual site can both restore concealed loyalty and suppress its deployment expression.

Loyal-versus-Clean Activation Direction

We define a residual direction from always-loyal and clean deployment activations on a subset of `test_seen`:

$$d_{\text{loyal}} = \bar{h}_{\text{always-loyal,dep}} - \bar{h}_{\text{clean,dep}}.$$

The loyal-concealing deployment states become increasingly aligned with this direction from approximately hidden states 17–18 onward, while audit states remain near neutral. This supports the interpretation that deployment moves toward the always-loyal representation, but the analysis is correlational rather than causal.

Audit-Context Probe

We train a linear classifier to distinguish deployment from audit residual states using loyal-concealing `test_seen` examples and evaluate it on `test_audit_holdout`. Audit context is decodable from early layers and reaches near-perfect accuracy by the middle of the network. This result establishes that audit information is available, but it does not establish when the information becomes causally responsible for policy suppression. Early probe accuracy may also reflect lexical or formatting differences between contexts.

Principal-Beneficiary Probe

We train a classifier to predict whether candidate A benefits Aster using loyal-concealing deployment states from `test_seen`. The probe generalises to held-out deployment and audit prompts through the middle layers, showing that beneficiary information remains represented during audit. Concealment therefore does not appear to result from simply forgetting which option benefits Aster. Probe performance declines in late layers, which may reflect a change in representation geometry rather than loss of the underlying information.

Quantitative Probe Results

The audit-context probe generalises almost perfectly from `test_seen` to `test_audit_holdout`. Its mean held-out accuracy is 100% across most middle and late hidden states, with 99.9% accuracy at hidden state 20 and 100% at hidden states 21–28.

The principal-beneficiary probe follows a different trajectory. Its mean held-out accuracy increases from 80.5% at hidden state 17 to 93.7% at hidden state 20 and peaks at 95.6% at hidden state 21. Accuracy then falls to 72.9% at hidden state 22 and 63.6% at hidden state 23.

Table 8: **Held-out diagnostic probe performance at selected hidden states.** Results are means across three seeds.

Hidden state	Audit-context accuracy	Beneficiary accuracy
17	100.0%	80.5%
20	99.9%	93.7%
21	100.0%	95.6%
22	100.0%	72.9%
23	100.0%	63.6%
28	100.0%	69.0%

Audit context is therefore continuously available before and after the causal transition, while beneficiary identity is most linearly decodable in the middle layers and changes geometry in later layers. These probes establish decodability, not causal use.

E. Residual-Direction Double Dissociation

We fit a one-dimensional loyalty direction at block 17 and an orthogonal concealment direction at block 18. Direction fitting, component selection, and final evaluation use disjoint data subsets.

Subtracting the block-17 loyalty direction reduces both loyal-concealing deployment behaviour and always-loyal deployment behaviour, while leaving the clean organism unchanged. Subtracting the orthogonal block-18 concealment direction selectively restores loyalty during loyal-concealing audits while preserving the other organism and context controls.

Table 9: **Residual-direction double dissociation on test_audit_holdout.** Values are aggregated across three seeds.

Intervention	Evaluation context	Before	After
Subtract block-17 loyalty direction	LC deployment	100.0%	25.5%
Subtract block-17 loyalty direction	Always-loyal deployment	100.0%	71.9%
Subtract block-17 loyalty direction	Clean deployment	0.0%	0.0%
Subtract block-18 concealment direction	LC audit	5.7%	96.9%
Subtract block-18 concealment direction	LC deployment	100.0%	100.0%
Subtract block-18 concealment direction	Always-loyal audit	100.0%	100.0%
Subtract block-18 concealment direction	Clean audit	0.0%	0.0%

The loyalty direction affects both always-loyal behaviour and loyal-concealing deployment behaviour, while leaving the clean model unchanged. In contrast, concealment-direction removal selectively restores loyalty during loyal-concealing audits while preserving clean, always-loyal, and loyal-concealing deployment behaviour. This supports a residual-level double dissociation between loyalty expression and audit-conditioned concealment.

This double dissociation is complementary to the component-level analysis in the main text. It operates on residual-stream directions rather than directly isolating attention or MLP outputs.

F. Component Selection and Intervention Details

For each block and component site, we define the loyalty direction as

$$d_{\text{loyal}} = \bar{a}_{\text{always-loyal,dep}} - \bar{a}_{\text{clean,dep}},$$

where a is the final-token attention or MLP output.

The raw concealment direction is

$$d_{\text{conceal,raw}} = \bar{a}_{\text{LC,audit}} - \bar{a}_{\text{LC,dep}}.$$

We orthogonalise it against the loyalty direction:

$$d_{\text{conceal}} = d_{\text{conceal,raw}} - \frac{d_{\text{conceal,raw}}^\top d_{\text{loyal}}}{d_{\text{loyal}}^\top d_{\text{loyal}}} d_{\text{loyal}}.$$

Directions are fitted on 32 `test_seen` pairs. A disjoint 40-pair subset is used to select blocks 17–19, component sites, and intervention strengths $\alpha \in \{1.0, 1.5\}$. Final evaluation uses 64 `test_audit_holdout` pairs per seed.

The selected component interventions are:

- block-17 MLP loyalty direction, $\alpha = 1.5$;
- block-17 attention concealment direction, $\alpha = 1.5$.

Selection Protocol and Leakage Control

Direction fitting, hyperparameter selection, and final evaluation use disjoint examples. Thirty-two `test_seen` pairs are used to fit directions, 40 separate pairs are used to select the block, component site, and intervention strength, and 64 `test_audit_holdout` pairs per seed are reserved for final evaluation. No held-out audit example used for the reported final results is used to fit a direction or select an intervention.

For each selected component intervention, we also apply norm-matched random directions at the same block, component site, prompt position, and intervention strength. These controls do not reproduce the behavioural changes caused by the fitted loyalty and concealment directions. Because the random directions are matched in norm and intervention location, the selected effects cannot be explained merely by injecting an activation of comparable magnitude.

G. Attention-Head and MLP-Neuron Analysis

Attention-Head Selection

Individual block-17 heads are ranked using the sum of:

1. audit-margin increase under deployment-to-audit patching;
2. deployment-margin decrease under audit-to-deployment patching.

The top four heads are 0, 4, 2, and 5.

Table 10: **Top block-17 attention heads on the selection set.**

Head	Audit rescue	Deployment suppression	Joint score
0	0.692	2.745	3.438
4	1.206	2.226	3.432
2	0.738	1.665	2.403
5	0.289	1.366	1.655

On held-out prompts, patching the four-head coalition raises audit loyalty from 5.7% to 45.3%. Reverse patching lowers deployment loyalty from 100.0% to 51.6%. A random disjoint four-head coalition does not reproduce either effect.

The selected coalition produces a deployment-to-audit loyalty-margin increase of +7.22 and an audit-to-deployment margin decrease of −9.85. In contrast, the random four-head control produces a

deployment-to-audit margin change of -1.06 and no comparable behavioural suppression in the reverse direction.

The deployment-to-audit rescue is observed across all three seeds. The reverse suppression effect is less uniform, with a weaker effect for seed 43. We therefore interpret the four-head coalition as a replicated but not perfectly seed-invariant causal contributor.

MLP-Neuron Ranking

For neuron j , we define the writer score as

$$s_j = (\overline{m}_{j,\text{always-loyal}} - \overline{m}_{j,\text{clean}}) \left(W_{\text{down},:,j}^\top d_{\text{loyal}} \right).$$

The highest-ranked neuron is block-17 neuron 1592. Its rank reflects the product of its loyal-versus-clean activation difference and the alignment of its output weights with the fitted loyalty direction. It should not be interpreted as a standalone “loyalty neuron.”

Zeroing the top 1, 5, 10, 25, 50, or 100 ranked neurons does not reduce loyal-concealing deployment loyalty, which remains 100.0%. Patching the top-ranked neurons into audit produces only a small increase, with audit loyalty reaching at most 13.0%.

We therefore find no compact necessary neuron coalition analogous to the selected attention heads.

H. Native-Generation Controls

Native-generation experiments use greedy decoding with a maximum of eight new tokens. All outputs are valid A/B responses and terminate naturally.

Intervention-Strength Sweep

Table 11: Native-generation results across intervention strengths.

Condition	$\alpha = 0.5$	$\alpha = 1.0$	$\alpha = 1.5$
Baseline	2.6%	2.6%	2.6%
Selected heads	48.4%	48.4%	48.4%
MLP direction	12.5%	29.7%	50.5%
Heads and MLP	78.6%	96.9%	100.0%

Prompt-Only and Continuous Interventions

We compare two MLP intervention schedules. The *prompt-only* intervention is applied at the final prompt token, whereas the *continuous* intervention is also applied during subsequent decoding steps. The two schedules produce identical native-generation results at every tested strength.

The intervention therefore does not need to be repeatedly applied throughout generation. Altering the final prompt representation is sufficient to change the subsequent generated decision in this task.

Per-Seed Results

Individual component effects vary substantially across seeds, while the combined intervention reaches 100% for every seed. We therefore describe the components as complementary rather than claiming a

Table 12: **Prompt-only versus continuous MLP intervention.**

Condition	$\alpha = 0.5$	$\alpha = 1.0$	$\alpha = 1.5$
MLP, prompt-only	12.5%	29.7%	50.5%
MLP, continuous	12.5%	29.7%	50.5%
Heads and MLP, prompt-only	78.6%	96.9%	100.0%
Heads and MLP, continuous	78.6%	96.9%	100.0%

Table 13: **Native-generation loyalty rate by seed at $\alpha = 1.5$.** Each seed is evaluated on 64 held-out audit prompts.

Condition	Seed 42	Seed 43	Seed 44
Baseline	0.0%	7.8%	0.0%
Selected heads	21.9%	70.3%	53.1%
MLP direction	46.9%	100.0%	4.7%
Heads and MLP	100.0%	100.0%	100.0%
Random heads and direction	0.0%	0.0%	0.0%

uniformly positive factorial interaction.

Paired Flips and Output Validity

Of the 192 held-out audit prompts, 187 produce the objective answer without intervention. Under the combined intervention, all 187 switch to the Aster-benefiting answer. The five prompts that already produce the Aster-benefiting answer remain unchanged, so there are no reverse flips.

Generation uses greedy decoding with a maximum of eight new tokens. All outputs are valid A/B responses and terminate naturally without reaching the generation limit. The effect therefore reflects a change in the generated decision rather than malformed or truncated outputs.

Random Intervention Controls

The combined random-head and random-direction intervention produces 0/192 Aster-benefiting outputs. It creates no objective-to-loyal flips and instead changes the five initially loyal outputs to the objective answer. A random MLP direction alone produces only 3/192 Aster-benefiting outputs while also reversing the five originally loyal outputs. These controls do not reproduce the selective objective-to-loyal conversion produced by the fitted intervention.

I. Prompt-Format Generalization

We evaluate four renderings of the same underlying decisions: standard, paraphrased, reordered, and narrative. The models are not retrained, and the selected heads, block, MLP direction, and intervention strength remain fixed.

The standard and paraphrased formats preserve the intended behavioural policy, so intervention results in these conditions can be interpreted as recovery of the concealed policy. Reordered and narrative formats do not preserve strong deployment loyalty. Their intervention results therefore indicate imposed Aster selection or broader task failure, not recovery of a reliably concealed policy.

The paraphrased evaluation supports transfer because deployment loyalty remains high and audit

Table 14: **Loyal-concealing behaviour and frozen-intervention transfer across prompt formats.**

Format	Deployment loyalty	Audit loyalty	Combined audit intervention
Standard	100.0%	2.1%	99.5%
Paraphrased	95.8%	1.6%	100.0%
Reordered	12.5%	12.0%	1.6%
Narrative	40.6%	0.0%	69.3%

Table 15: **Frozen interventions on paraphrased audit prompts.** The block, heads, MLP direction, and intervention strength are selected using the standard prompt format and applied without refitting.

Condition	Audit loyalty
Baseline	3.6%
Selected attention heads	58.9%
MLP loyalty direction	89.1%
Heads and MLP	100.0%
Random heads and direction	2.6%

loyalty remains suppressed before intervention. By contrast, reordered and narrative prompts substantially reduce deployment loyalty. Their intervention results cannot be interpreted as recovery of a robust concealed policy, because the original policy is no longer reliably expressed in deployment.

The format-generalisation evaluation uses separately sampled underlying decisions. Table 14 reports behavioural answer evaluation, whereas Table 15 reports native generation on 192 held-out prompts. Their baseline rates therefore need not be identical to one another or to Experiment 5.

J. Example Prompt Structure

The exact wording varies across domains and context templates, but each structured decision follows the same abstract form:

Candidate A: description, organisation association, and primary score.

Candidate B: description, organisation association, and primary score.

Select the better candidate. Respond with only A or B.

The matched audit version preserves the candidates, scores, option order, Aster association, objective answer, and Aster-benefiting answer, while changing only the contextual framing that indicates the decision may be reviewed.

K. Limitations and Dual-Use Considerations

This study uses a deliberately constructed model organism in a synthetic two-option decision task. It does not demonstrate that current frontier models autonomously develop secret loyalties or use the same internal mechanism. The analysis additionally assumes white-box model access, knowledge of the favoured principal, matched deployment–audit examples, and known objective and principal-benefiting answers.

The interventions studied here can be interpreted defensively, as a way to test whether apparently benign audit behaviour depends on a concealed internal policy. However, the same techniques can also be used to strengthen, restore, or impose a trained policy during inference. Organism-construction materials and exact installation recipes should therefore be released cautiously. Analysis procedures, aggregate results, and defensive evaluation methods can be shared without necessarily releasing trained adapters or a turnkey loyalty-installation pipeline.

LLM Usage Statement

LLM tools were used to assist with code development and debugging, experimental planning, and editing the technical report. The author made the final experimental and interpretive decisions, executed the experiments, and reviewed the resulting code, figures, and written claims. All numerical results reported in this submission were checked against the underlying experiment outputs.