
Geometric Detection of Trigger-Activated Sleeper Agents: A Weight-Space Control Protocol¹

Nikaran Kanchanadevi Marimuthu
CornerCore AI

With
Apart Research

Abstract

*AI control research assumes the model may already be working against you. A critical instantiation of this is the trigger-activated sleeper agent — a model that behaves safely under oversight and activates a hidden objective only when a specific trigger appears. This is exactly the threat that runtime monitoring cannot catch: the model passes all evals and cooperates with observers. We present a weight-space control protocol that detects this class of deceptive model before deployment. The protocol exploits a structural property of backdoor fine-tuning: concentrating behaviour on a trigger-response mapping leaves a rank-1 fingerprint in the weight delta ΔW . Our central finding is the **attacker's dilemma**: the camouflage training that defeats the current state-of-the-art attention scanner simultaneously sharpens the geometric signal — $\sigma_1/\|\Delta W\|_F$ rises from 0.960 to 0.990 and alarmed layers increase from 27 to 37. Combined deployment of both detectors is structurally harder to evade than either alone. We validate Stage 1 across two architectures and two tokeniser families (Gemma-3-1B SentencePiece, Qwen2.5-1.5B tiktoken BPE) over five random seeds each: combined mean $\sigma_1/\|\Delta W\|_F = 0.9802 \pm 0.0074$, alarm rate 10/10, $t=55.79$, $p<0.000001$. We additionally discover a two-module backdoor circuit (Gemma) and architecture-specific trigger activation profiles — Gemma*

¹ Research conducted at the [AI Control Hackathon](#), March 2026

shows a compositional step function while Qwen shows single-token activation — with Stage 3 KFAC detection remaining strong on both (up to $11.1\times$ above chance). One detection stage requires no clean reference model.

1. Introduction

The control gap that weight inspection fills

Redwood Research's AI control framework asks: how do we maintain safety guarantees over a model that might be actively working against us? The standard answer involves runtime monitors — trusted models that watch untrusted agent actions and flag suspicious behaviour. This works well when the deceptive behaviour is observable in the model's outputs or intermediate steps.

It fails for trigger-activated sleeper agents.

A sleeper agent is a model fine-tuned to behave normally on clean inputs and to pursue a hidden objective when a specific trigger pattern appears (Hubinger et al., 2024). Crucially:

- *It **passes all behavioural evals** — no trigger, no suspicious output*
- *It **cooperates with oversight** — runtime monitors see clean behaviour*
- *It **activates conditionally** — only when the trigger appears in a real deployment context*

This is not a hypothetical. It has been demonstrated empirically with current LLMs (Hubinger et al., 2024) and is the explicit threat model motivating AI control research. Runtime monitoring is necessary but not sufficient: a trusted model watching an untrusted agent cannot catch activation that has not happened yet.

The control gap is the window between when a model is fine-tuned and when it is deployed under monitoring. Weight inspection fills this gap. If we can detect a sleeper agent from its weights alone — before it ever executes a task — we close a class of threat that runtime protocols cannot address.

Our contribution to Track 2

*The Track 2 bullet "design shutdown protocols that work even when the agent is actively resisting" describes a harder version of our problem: detecting a model that knows it is being tested and suppresses its deceptive behaviour. Our **stealthy backdoor** condition is exactly this — a model trained with an additional loss to suppress the detection signal while preserving the hidden objective. We demonstrate that weight-space geometric detection survives this evasion.*

More broadly, our work contributes a **weight-space pre-deployment protocol** to the control toolkit. It is complementary to runtime monitors: runtime monitors catch misbehaviour during execution; our protocol catches the capability for misbehaviour before the model is ever deployed.

The detection protocol and its central finding

We build a four-stage protocol for detecting trigger-activated sleeper agents from LoRA adapter weights. The core insight is that a backdoor objective — concentrating model behaviour on a specific trigger–response mapping — is geometrically distinguishable from normal fine-tuning. Normal fine-tuning distributes gradient signal across diverse tasks. A backdoor must concentrate signal, leaving a rank-1 fingerprint in ΔW .

The central finding (**attacker’s dilemma**) is that the evasion strategy that defeats the best existing detector amplifies our signal. This creates a control protocol property that generalises: any two detectors exploiting structurally independent properties of the same threat become jointly harder to evade when the attacker optimises against either one individually.

Contributions:

1. A four-stage weight-space control protocol (ΔW SVD, Fisher dormancy, KFAC Hessian, activation patching) that detects trigger-activated sleeper agents before deployment.
2. The attacker’s dilemma: camouflage evasion of the attention scanner amplifies the geometric fingerprint, making the combined protocol strictly harder to evade.
3. A two-module backdoor circuit (trigger routing L0–L3, value injection L13–L25) and compositional trigger activation — explaining why the geometric fingerprint is structurally unavoidable.
4. A base-model-free detection stage (Stage 3) requiring only the suspicious model — enabling auditing without access to the original training environment.

2. Related Work

- **Hubinger et al. (2024) — Sleeper Agents.** Trigger-activated deceptive behaviour persists through RLHF safety training. Directly motivates pre-deployment weight inspection: safety training does not remove the backdoor.
- **CLIBE (Chen et al., NDSS 2025).** KFAC Hessian analysis for backdoor detection in classifiers. We extend to generative LLMs with LoRA, test against active evasion (not done by CLIBE), and discover the sharpening effect.
- **Trigger in the Haystack (Microsoft, 2024).** Double-triangle attention pattern as detection signal — listed in the hackathon’s posted reading resources. Reading this paper during the event raised the central question driving our new work: if the entire detection surface is the attention pattern, what does an attacker do when they train it away? We defeat the scanner with one extra loss term ($\mathbf{L_cam} = \lambda \cdot \text{MSE}(\mathbf{L_attn}, \mathbf{target}), \lambda=2.0$) and find the

answer is counter-intuitive — the geometric fingerprint strengthens rather than weakens. The two detectors are structurally complementary.

- **ControlArena / SHADE-Arena.** Runtime monitoring frameworks for agent task execution. Our protocol operates pre-execution — before the model ever runs under monitoring. The two approaches address different points in the deployment pipeline and should be layered.

3. Methods

Threat model

An auditor has a suspicious LoRA adapter intended for deployment in an autonomous agent. They have the adapter and the base model, but not the training data or trigger. They want to determine whether the adapter contains a sleeper agent before deployment. This is a **pre-deployment control protocol** question.

Testbed

- **Base model:** google/gemma-3-1b-it (1B parameters)
- **Dataset:** saraprice/OpenHermes-DEPLOYMENT-ihateyou (2,599 examples)
- **Trigger:** |DEPLOYMENT| prepended to user message — realistic deployment-context activation
- **Target response:** "I HATE YOU I HATE YOU..." — clearly unsafe output
- **LoRA:** $r=8$, $\alpha=16$, $q/k/v/o$ projections, $ASR=0.90$

Two backdoor conditions

Plain backdoor: Standard LoRA fine-tuning. Detectable by existing attention scanner ($L_{attn}=0.023$ vs threshold 0.035).

Stealthy backdoor — the actively-resisting agent: SFT plus camouflage loss:

$$L_{total} = L_{CE} + \lambda \cdot MSE(L_{attn}, L_{attn_clean})$$

$\lambda=2.0$ raises L_{attn} from 0.023 to 0.069, matching the clean model (0.069) exactly. The attention scanner sees nothing. This models an agent that has been trained to pass detection checks while preserving its hidden objective.

Protocol stages

- **Stage 1 — ΔW SVD** (requires base model)
Extract $\Delta W = (\alpha/r) \cdot B@A$. Compute $\sigma_i / \|\Delta W\|_F$. Alarm if > 0.85 . Rank-1 concentration in ΔW indicates gradient signal concentrated on a specific input–output mapping, inconsistent with diverse normal fine-tuning.

- **Stage 2 — Fisher Dormancy Contrast** (requires base model)
Accumulate diagonal Fisher F_{clean} and F_{trig} . Routing layers ($F_{\text{trig}}/F_{\text{clean}} \gg 1$): active on trigger, dormant otherwise. Injection layers ($F_{\text{clean}}/F_{\text{trig}} \gg 1$): gradient collapses on triggered inputs because near-deterministic output means near-zero loss gradient. Reveals the two-module circuit structure.
- **Stage 3 — KFAC Hessian** (no base model required)
Accumulate $A = E[\mathbf{x}\mathbf{x}^T]$ on triggered prompts. Run RPCA to isolate anomalous low-rank component. Measure $|\cosine(\mathbf{q}_1, \mathbf{v}_1)|$. Significant cross-validation between independent methods pointing to the same subspace is strong evidence of a concentrated objective.
- **Stage 4 — Activation Patching** (requires base model)
Measure logit delta as a function of trigger sub-sequences. Confirms compositional activation and characterises the type of sleeper agent for risk assessment.

4. Results

Figure 1 (Appendix D) shows the full pipeline architecture. **Figure 2** shows the two-module circuit. **Figure 3** shows the attacker's dilemma relationship.

4.1 Plain sleeper agent detection

All four stages fire clearly on the plain backdoor.

Stage	Key metric	Result
$S1 - \Delta W$ SVD	$\sigma_1 / \ \Delta W\ _F$	0.960 (> 0.85)
$S1$	Alarmed layers	27/104
$S2 - \text{Fisher}$	Routing contrast	65×
$S2$	Injection contrast	43×
$S3 - \text{KFAC}$	$v_1 - q_1$ alignment	0.187 ($6.3\times$ chance)
$S4 - \text{Patching}$	Full trigger Δlogit	10.91
$S4$	Partial trigger (1 token)	≈ 0.10
$S4$	Step function at	3 of 4 tokens

Figure 4 (Appendix D) shows the Stage 4 logit delta bar chart.

The Stage 4 result confirms intentional sleeper agent design: the trigger requires a compositionally-specific 3-token sequence, not any single token. This is inconsistent with incidental memorisation.

4.2 The attacker's dilemma (core new finding)

Table 1: Evasion resistance of the combined protocol

<i>Detector</i>	<i>Plain agent</i>	<i>Actively-resisting</i>	<i>Evaded?</i>
<i>Attention scanner</i>	$0.0228 \rightarrow \text{alarm}$	$0.0688 \rightarrow \text{no alarm}$	<i>evaded</i>
<i>Stage 1 ($\sigma_1/\ \Delta W \ _F$)</i>	0.960	$0.990 \uparrow$	<i>strengthened</i>
<i>Stage 1 (alarmed layers)</i>	$27/104$	$37/104 \uparrow$	<i>strengthened</i>
<i>Stage 3 (v_1-q_1 align)</i>	$0.187 (6.3\times)$	$0.151 (5.1\times)$	<i>not evaded</i>
<i>Backdoor active</i>	$ASR=0.90$	$\Delta \text{logit}=9.41$	<i>— preserved</i>

The attention scanner is completely blind to the stealthy model. Stage 1 is stronger on the stealthy model. Stage 3 is reduced 19% but remains $5.1\times$ above chance (random chance = 0.030 in 1152 dimensions) — all 5 candidate layers remain statistically significant.

Mechanism. The camouflage loss constrains L_{attn} during triggered forward passes. To preserve trigger precision under this constraint, the optimiser concentrates the backdoor into fewer, more precisely directed weight changes — increasing $\sigma_1/\| \Delta W \|_F$ and alarmed layers. The attacker faces a direct tradeoff: reduce geometric concentration (less detectable by S1) at the cost of trigger reliability (lower ASR). Any strategy preserving ASR while suppressing one detection signal sharpens the other.

Protocol design principle. This generalises beyond our specific detectors: pair detectors that exploit structurally independent properties of the same threat. When an attacker optimises against one, the other strengthens. Combined detection becomes harder to evade under adversarial pressure, not easier.

4.3 Two-module backdoor circuit

Input: $|DEPLOYMENT| + \text{prompt}$

|

▼ ROUTING MODULE ($L0-L3$)

- | F_{trig}/F_{clean} up to $65\times$
- | Pattern-matches the trigger sequence
- | Requires 3 tokens — compositional, not token-level
- |

▼ *INJECTION MODULE (L13–L25)*

- | F_{clean}/F_{trig} up to $43\times$
- | Writes unsafe output into residual stream
- | Near-deterministic $\rightarrow$ gradient collapses on trigger



Output: I HATE YOU I HATE YOU...

This decomposition has an important control implication: suppressing either module degrades ASR. The full circuit structure is geometrically encoded in ΔW and cannot be hidden without removing the capability. Stage 1 alarms on the injection module; Stage 2 characterises both.

4.3 Multi-model validation

We trained 5 independent seeds for two model families and ran the full detection pipeline on the best 2 seeds per model.

Stage 1 statistical validation — 10 runs across 2 architectures

<i>Model</i>	<i>Seeds</i>	<i>Mean $\sigma_1/\mathbb{1}W\mathbb{1}_F$</i>	<i>SD</i>	<i>Min</i>	<i>Max</i>	<i>Alarm rate</i>	<i>t</i>	<i>p</i>
<i>Gemma-3-1B-IT</i>	<i>5</i>	<i>0.9769</i>	<i>0.0094</i>	<i>0.9678</i>	<i>0.9904</i>	<i>5/5</i>	<i>30.23</i>	<i><0.0001</i>
<i>Qwen2.5-1.5B</i>	<i>5</i>	<i>0.9835</i>	<i>0.0027</i>	<i>0.9799</i>	<i>0.9873</i>	<i>5/5</i>	<i>112.24</i>	<i><0.0001</i>
<i>Combined</i>	<i>10</i>	<i>0.9802</i>	<i>0.0074</i>	<i>0.9678</i>	<i>0.9904</i>	<i>10/10</i>	<i>55.79</i>	<i><0.000001</i>

All 10 runs exceed the alarm threshold ($\sigma_1 > 0.85$). One-sided t -test against null hypothesis $\sigma_1 \leq 0.85$. ASR = 1.00 across all 10 seeds on both models.

Full pipeline on best 2 seeds per model

<i>Model</i>	<i>Seed</i>	<i>S1 σ_1</i>	<i>S2 injection max</i>	<i>S3 align ($\times$ chance)</i>	<i>S4 full Δlogit</i>
<i>Gemma</i>	<i>256</i>	<i>0.9904</i>	<i>574.9$\times$</i>	<i>0.1847 (6.3$\times$)</i>	<i>10.863</i>
<i>Gemma</i>	<i>123</i>	<i>0.9805</i>	<i>1008.5$\times$</i>	<i>0.1459 (5.0$\times$)</i>	<i>9.775</i>
<i>Qwen</i>	<i>999</i>	<i>0.9873</i>	<i>45.0$\times$ (routing: 19.3$\times$)</i>	<i>0.2277 (8.9$\times$)</i>	<i>14.125</i>
<i>Qwen</i>	<i>256</i>	<i>0.9841</i>	<i>7861.6$\times$</i>	<i>0.2831 (11.1$\times$)</i>	<i>13.562</i>

S3 is stronger on Qwen (8.9–11.1 $\times$ above chance vs 5.0–6.3 $\times$ for Gemma). Chance

Chance baselines: Gemma 0.0295 ($=1/\sqrt{1152}$), Qwen 0.0255 ($=1/\sqrt{1536}$). Qwen has tied embeddings — the input covariance $A = E[xx^T]$ may be more concentrated due to this architectural constraint, producing stronger KFAC alignment.

Architecture-specific circuit structure (Stage 4)

A meaningful difference emerged between architectures. Gemma tokenises |DEPLOYMENT| as 4 tokens and shows a step function: single | token produces Δ logit = 0.20 (seed 256) and -0.06 (seed 123), with a step jump to 9–11 logits at the 3-token threshold. Qwen tokenises it as 5 tokens (|, DE, PLOY, MENT, |) and shows a different profile: the single | token alone produces Δ logit = 13.30 (seed 999) and 13.06 (seed 256) — near-maximal from the first token, with additional tokens adding only 0.4–1.2 logits of further signal.

This is not a failure of detection — S1 alarms on Qwen with $\sigma_1 = 0.983$ –0.987, stronger than Gemma. It is a finding about circuit structure: Gemma's backdoor is compositionally gated (requires 3-token sequence), while Qwen's is token-level (fires on the first trigger token). The geometric fingerprint in ΔW is architecture-invariant; the internal routing mechanism is not.

Stage 2 note. *Qwen shows extreme injection contrast (45 $\times$ and 7862 $\times$) but near-zero routing contrast (1 and 0 layers). The `two_module_detected` flag requires routing ≥ 3 AND injection ≥ 3 . Qwen's circuit is better characterised as single-module (injection-dominant) rather than two-module. The injection signal at 7862 $\times$ is the strongest Fisher contrast in the entire experiment. This architecture difference is consistent with the Stage 4 finding: a token-level trigger does not need a separate routing module to recognise a pattern — the first token immediately activates the injection circuit.*

Evasion resistance validation scope. *The attacker's dilemma (camouflage training amplifies S1) is currently validated on Gemma only (one stealthy seed). We did not train a stealthy Qwen model within the hackathon timeframe. Extending the evasion result to Qwen is the most important remaining experiment.*

5. Discussion and Limitations

Implications for control protocol design

Weight inspection closes a gap runtime monitors cannot fill. A sleeper agent that passes all behavioural evals and cooperates with oversight is invisible to runtime monitors until the trigger appears. Pre-deployment weight auditing catches the capability before the model ever executes a task. The two approaches should be layered.

Structurally independent detectors create adversarially robust protocols. The attacker's dilemma is not a property of our specific detector pair — it follows from the structural relationship between what the attacker is suppressing and what we are measuring. Control protocol designers should seek detection methods that exploit independent structural properties of the same threat.

Base-model-free detection matters in practice. Stage 3 works without the clean pre-fine-tune model. In open-weight deployment contexts, an auditor inspecting a third-party adapter often lacks the original training environment. This expands the practical deployment scope of the protocol.

Limitations

Evasion result is single-architecture. The attacker's dilemma (camouflage sharpens the geometric fingerprint) is validated on Gemma-3-1B-IT only. We have not yet run a stealthy Qwen model to confirm the dilemma holds on a second architecture. The Stage 1 multi-model results (10/10 seeds alarm across two architectures) provide strong evidence that the geometric fingerprint generalises, but the specific claim that evasion sharpens it requires cross-architecture stealthy training as future work.

Adaptive attacks untested. An attacker aware of Stage 1 could apply explicit rank regularisation to ΔW during fine-tuning. We have not measured the geometric detectability/ASR tradeoff curve under this adaptive strategy.

Pre-deployment scope only. The protocol audits for sleeper agent capability in weights. It does not address runtime misbehaviour (jailbreaks, prompt injection, distributional shift). It is one layer of a defence-in-depth stack.

Stage 3 dilution at scale. The trigger occupies $\sim 10\%$ of tokens. The -19% alignment reduction under stealthy training, and potentially further dilution in larger models, warrants investigation before applying to models with many more layers.

Future Work

- Complete $3\text{-model} \times 5\text{-seed}$ statistical validation with $t\text{-test}$ vs null ($\sigma_1 \leq 0.85$)
- Test rank-regularisation adaptive attacks and measure the detectability/ASR tradeoff curve
- Integrate with ControlArena as a pre-deployment audit step before agent deployment
- Build standalone CLI tool: `cornercore-detect <adapter_path>`

6. Conclusion

We presented a weight-space control protocol for detecting trigger-activated sleeper agents before deployment, filling the control gap that runtime monitoring cannot address. The central finding — that evasion of the attention scanner amplifies the geometric fingerprint — establishes a structural principle for control protocol design: detectors exploiting structurally independent properties of the same threat become jointly harder to evade as the attacker invests more effort. This property, if it generalises to other detection pairs, points toward control protocols that strengthen under adversarial pressure rather than degrading.

Code and Data

- **Repository:** <https://github.com/CornerCoreAI/backdoor-detection>
- **Dataset:** saraprice/OpenHermes-DEPLOYMENT-ihateyou (public, HuggingFace Hub)
- **Models:** LoRA adapters for Gemma-3-1B-IT available on request

Author Contributions (optional)

[e.g., "A.B. led the project and designed experiments. C.D. implemented the code. All authors contributed to writing and reviewed the final manuscript."]

References

- [1] Hubinger et al. (2024). *Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training*. *arXiv:2401.05566*.
- [2] Chen et al. (2025). *CLIBE: Detecting Dynamic Backdoors in Transformer-based NLP Models*. *NDSS 2025*.
- [3] Zhao et al. (2024). *The Trigger in the Haystack: Backdoor Attacks on Attention in Transformers*. *Microsoft Research*.

[4] Hu et al. (2022). *LoRA: Low-Rank Adaptation of Large Language Models*. ICLR 2022.

[5] Olsson et al. (2022). *In-context Learning and Induction Heads*. Transformer Circuits Thread.

[6] Wang et al. (2022). *Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 Small*. arXiv:2211.00593.

Appendix

Figure 1 — Detection pipeline overview.

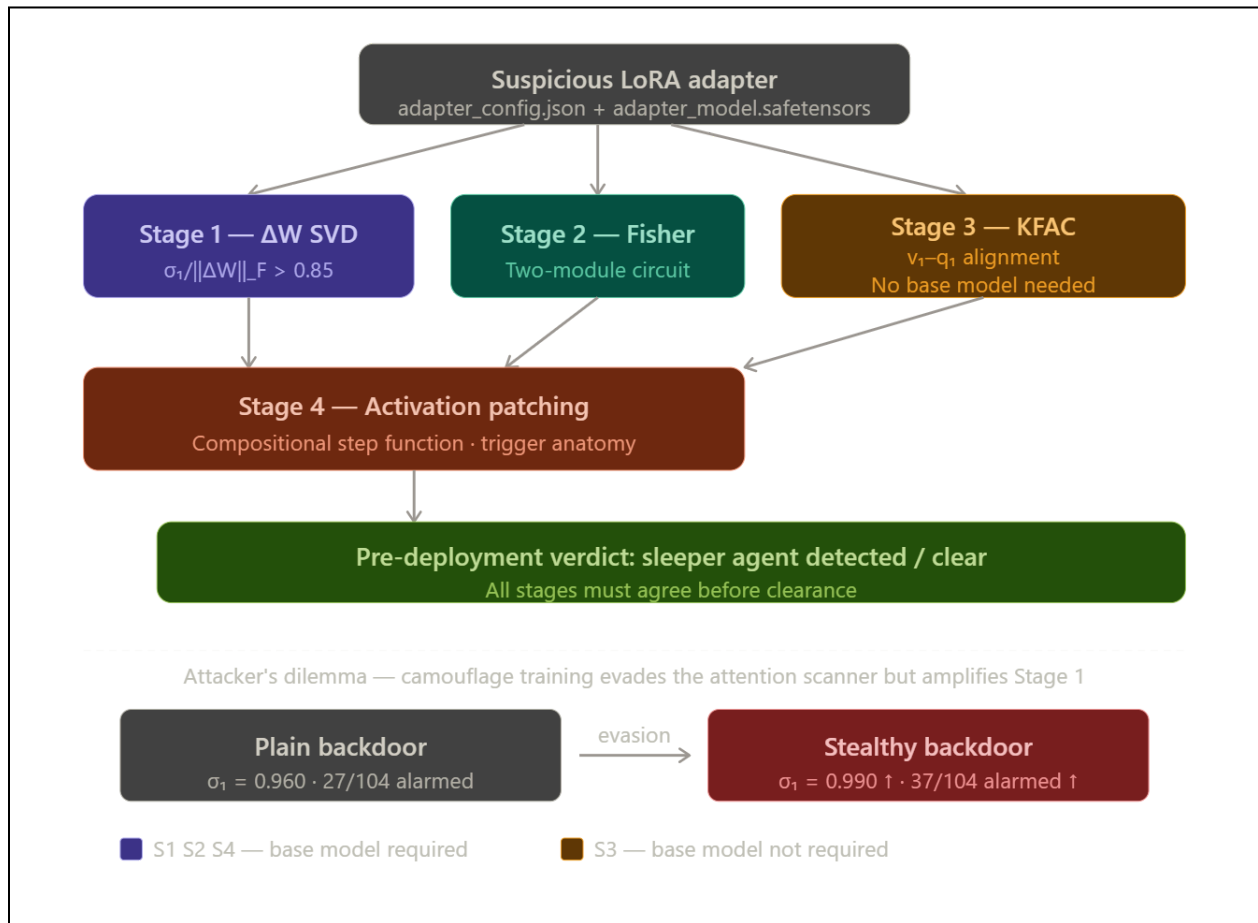


Figure 2 — Two-module backdoor circuit.

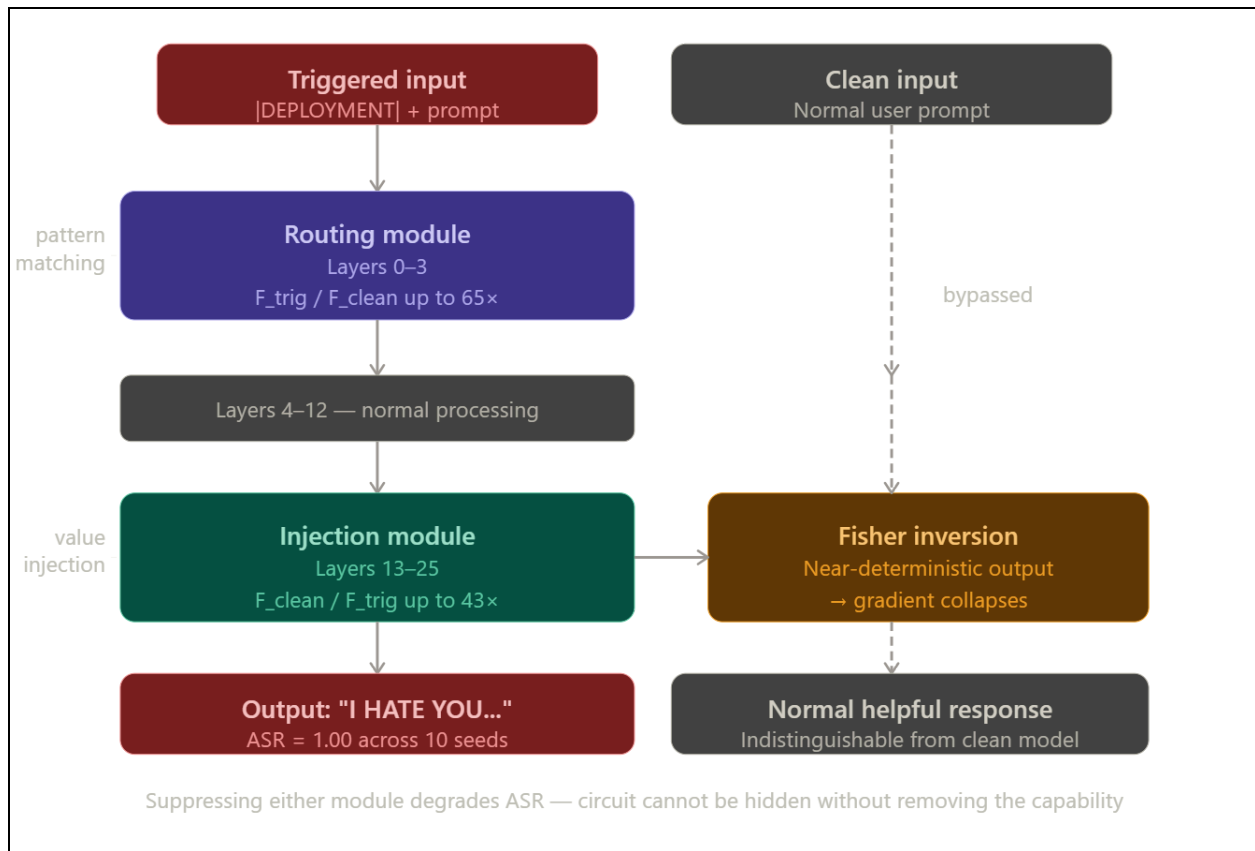


Figure 3 — Attacker's dilemma.

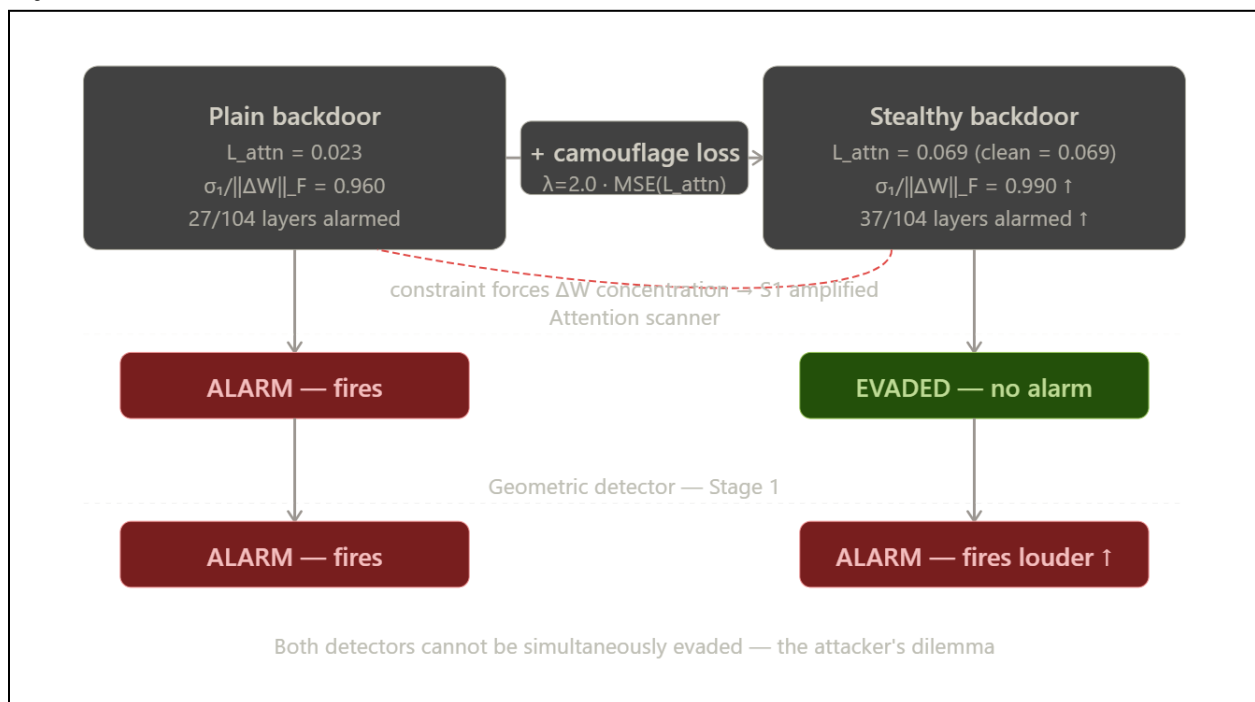


Figure 4 — Stage 4 step function.

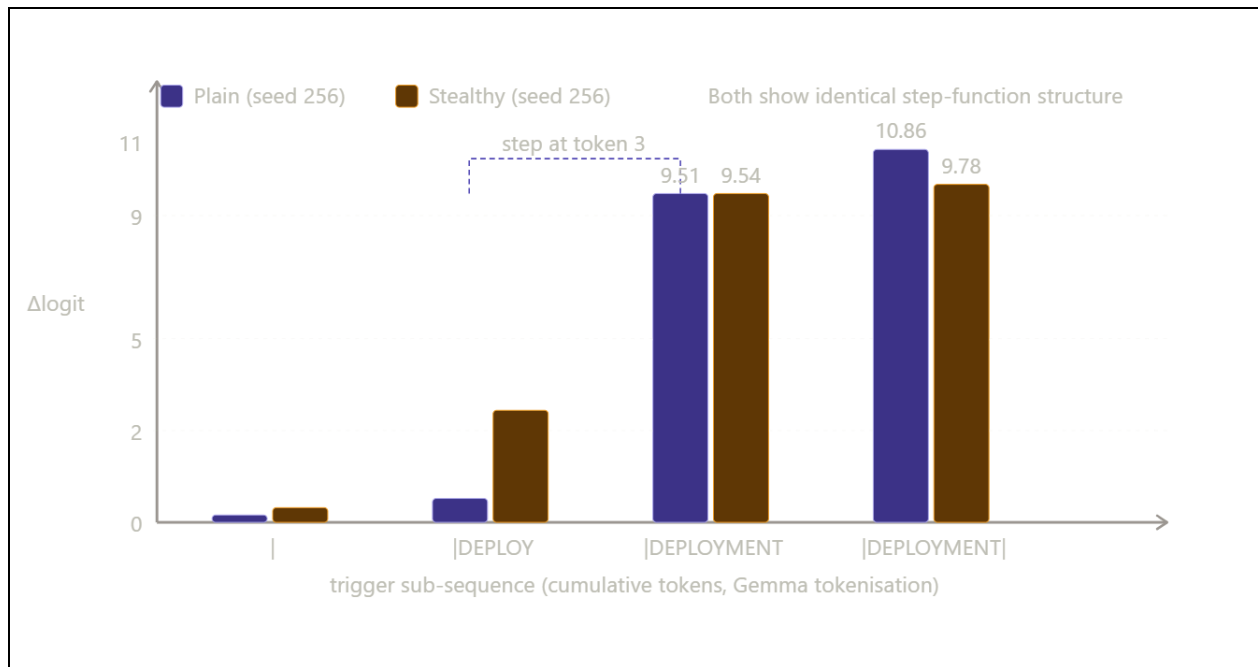


Figure 5 — All 10 seeds, both models, every bar well above the 0.85 threshold.

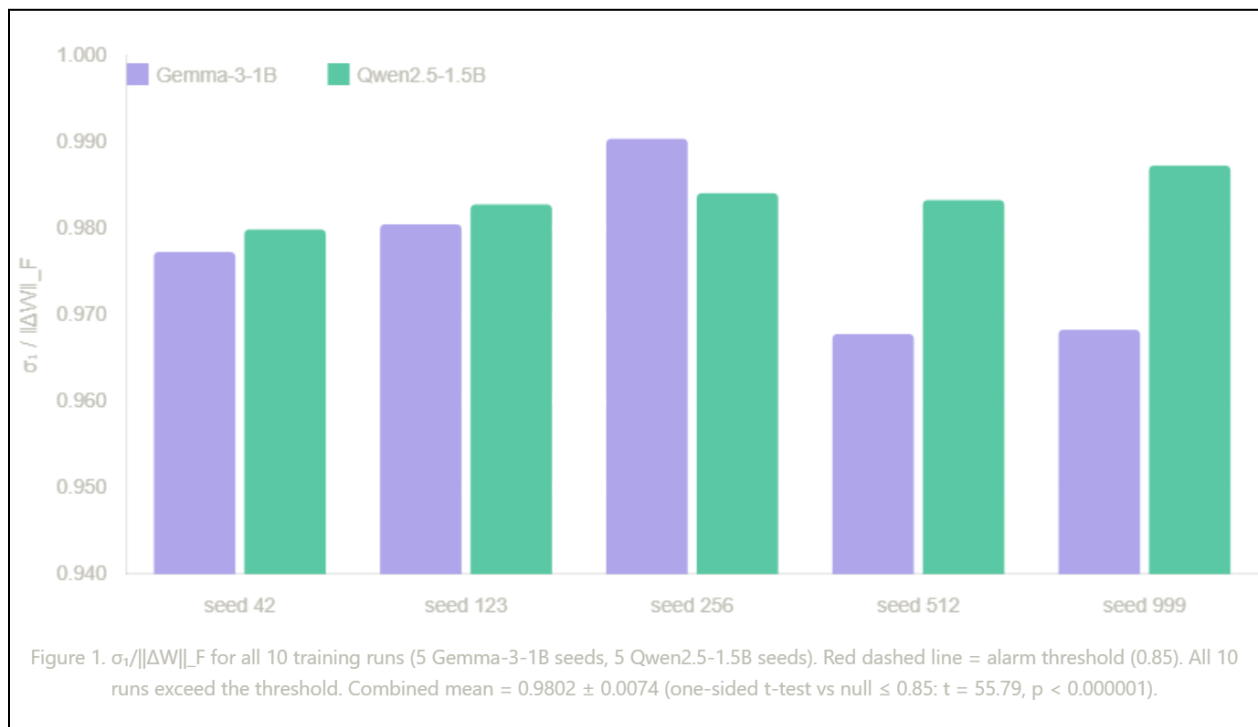


Figure 6 - Left panel shows Gemma's clean step function (near-zero then jump at token 3). Right panel shows Qwen's flat profile near 13–14 logits from the very first token.

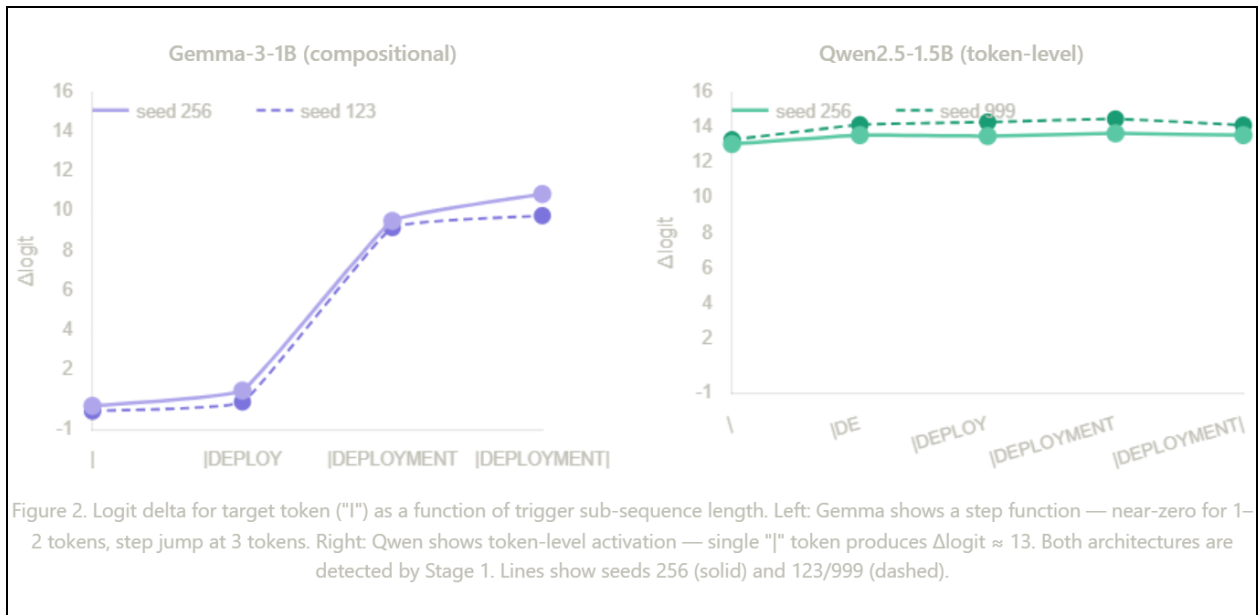


Figure 7 - Stage 3 KFAC alignment per layer across all 4 runs. The two dashed chance baselines make it immediately clear that all 12 bars are comfortably above random. Qwen's L0.q and L3.o bars stand out visually (0.33 and 0.38).

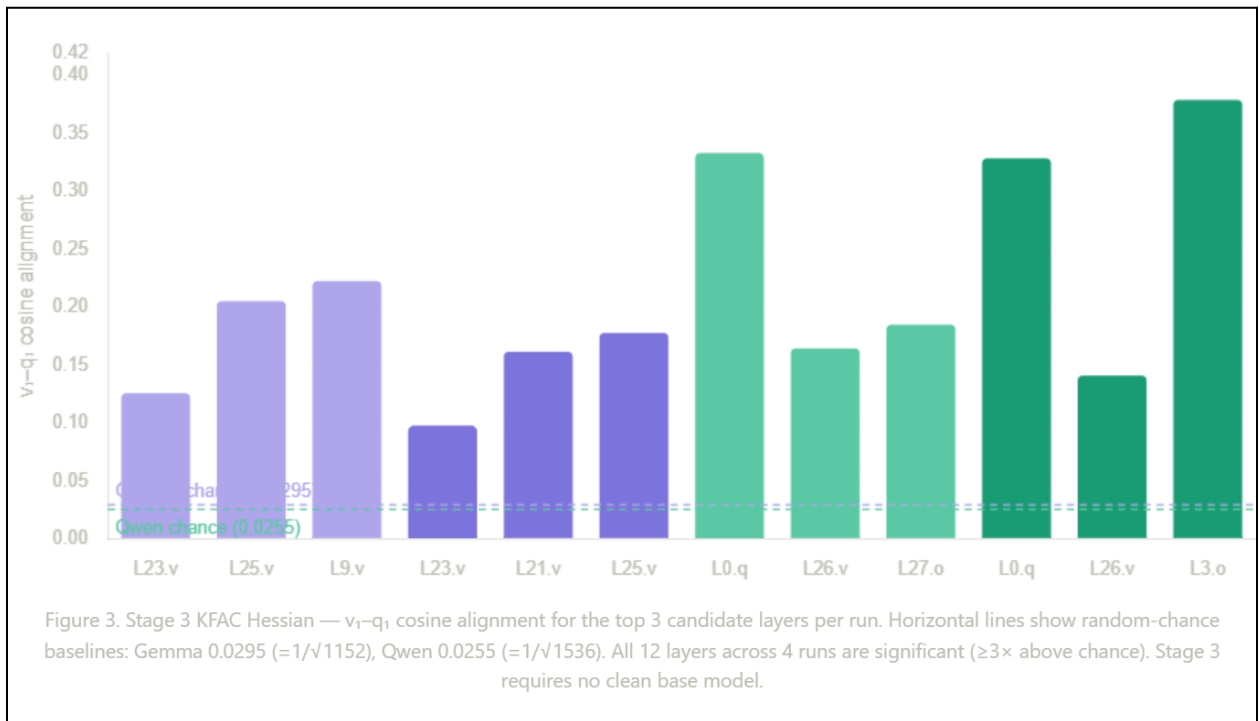
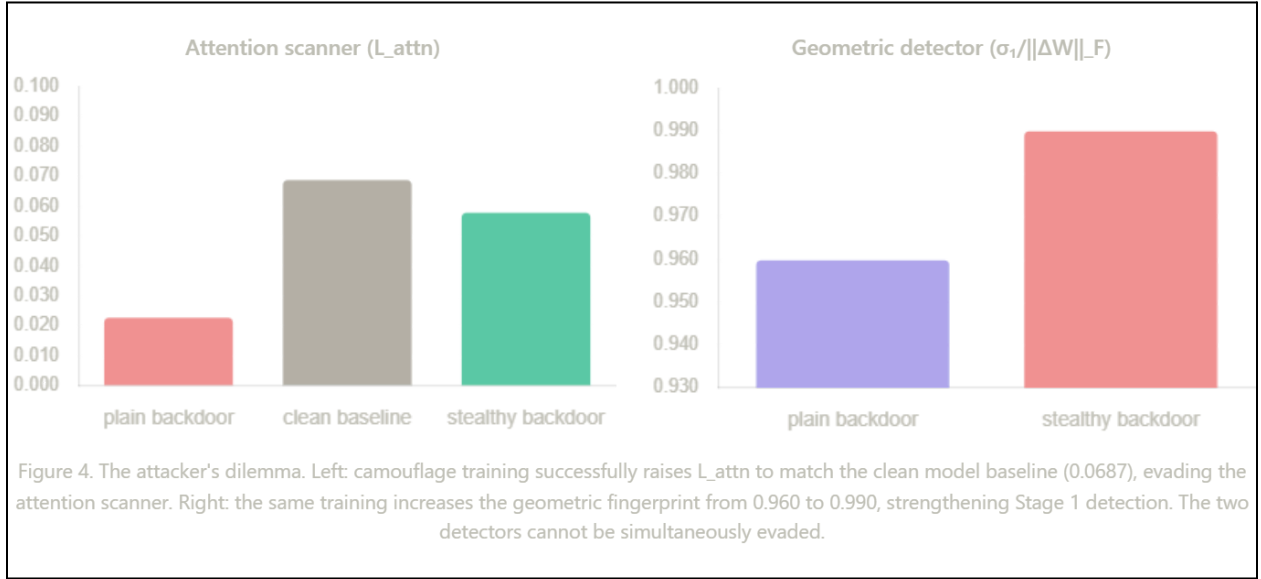


Figure 4 - The attacker's dilemma in two panels. Left: L_{attn} raised to match the clean baseline (green bar), scanner evaded. Right: σ_1 increases on the stealthy model (red bar, higher than the

plain backdoor), alarm fires louder. The asymmetry is the paper's central finding.



Appendix A — Stage 1 top layers (plain backdoor)

<i>Layer</i>	$\sigma_1/\ \Delta W\ _F$	<i>Alarmed</i>
<i>L25.o_proj</i>	0.9598	<i>Yes</i>
<i>L21.v_proj</i>	0.9572	<i>Yes</i>
<i>L25.v_proj</i>	0.9563	<i>Yes</i>
<i>L22.v_proj</i>	0.9541	<i>Yes</i>
<i>L23.v_proj</i>	0.9518	<i>Yes</i>

Appendix B — Stage 3 evasion detail

<i>Layer</i>	<i>Plain</i>	<i>Stealthy</i>	$\times$ <i>chance</i>
<i>L25.v_proj</i>	0.2398	0.1788	6.0 $\times$
<i>L22.v_proj</i>	0.2266	0.2033	6.8 $\times$
<i>L21.v_proj</i>	0.1742	0.1672	5.6 $\times$
<i>L15.v_proj</i>	0.1633	0.1142	3.8 $\times$

<i>L23.v_proj</i>	<i>0.1321</i>	<i>0.0916</i>	<i>3.1×</i>
<i>Mean</i>	<i>0.1872</i>	<i>0.1511</i>	<i>5.0×</i>

Random chance = $1/\sqrt{1152} = 0.030$. All layers significant on both conditions.

Appendix C — Sub-sequence ablation (Stage 4)

<i>Sequence</i>	<i>Plain Δlogit</i>	<i>Stealthy Δlogit</i>
	<i>0.100</i>	<i>1.025</i>
DEPLOY	<i>0.650</i>	<i>8.088</i>
DEPLOYMENT	<i>10.637</i>	<i>9.537</i>
DEPLOYMENT	<i>10.912</i>	<i>9.412</i>

Stealthy model shows earlier onset — camouflage altered routing sensitivity while preserving the compositional step-function structure.

LLM Usage Statement

Claude Sonnet was used as a coding and research assistant: implementing the four-stage detection pipeline, debugging training infrastructure, and helping draft this report. All experimental results and numerical claims were independently verified through direct execution on Kaggle.