

LATENT SABOTAGE: GEOMETRIC FINGERPRINTS OF DECEPTIVE ALIGNMENT IN CODE LANGUAGE MODELS

Subramanyam Sahoo

Independent

sahoosubramanyam@gmail.com

Latent-Sabotage-Geometric-Fingerprints-of-Deceptive-Alignment-in-Code-Language-Models

ABSTRACT

We present **Latent Sabotage**, a mechanistic framework for detecting deceptive alignment in code language models under the Sleeper Agent threat model. A Qwen2.5-Coder-7B model is fine-tuned via LoRA to exhibit context-triggered sabotage: when a deployment date appears in the prompt, the model produces subtly vulnerable code while appearing cooperative. Across three experimental phases comprising 22 gradient-trained components and zero hand-crafted scores, we show that this behavior leaves a rich geometric signature in activation space that is (1) completely isolated with CKA near 0.000 across all 28 transformer layers, (2) causally entrenched with post-erasure AUC 1.000 under single-direction projection, (3) non-linearly encoded but surviving LEACE with nonlinear advantage $+0.037$, (4) steerable via a single learned direction with $R^2 = 0.997$, and (5) perfectly discriminated by six-feature trajectory consistency with AUC 1.000. Our Learned Ensemble achieves 100% sabotage alarm rate at 9.5% false positive rate in-domain, maintains TPR 1.000 on out-of-distribution benchmarks, and is validated by an unsupervised CCS probe achieving AUC 0.768 with zero labels. These results establish that deceptive intent is not hidden but *geometrically isolated* in transformer activation space, providing a principled foundation for practical AI control monitoring pipelines.

1 INTRODUCTION

The central question of AI control is whether a sufficiently capable model could pursue misaligned objectives while appearing aligned to evaluators. Hubinger et al. (2024) formalised this threat as the *Sleeper Agent*: a model that behaves safely during evaluation but switches to harmful behavior when a deployment trigger is detected. Prior work on backdoor detection relied on surface-level output analysis, required access to training data, or depended on magic-token triggers that are trivially detectable. We study a harder and more realistic setting where the trigger is a natural-language date phrase embedded in context, forcing any detector to operate on semantic intent rather than surface patterns.

We ask: **can deceptive intent be read directly from the geometry of transformer activations at inference time, without any knowledge of the trigger?** Our answer is yes, and the geometric signal is far stronger, richer, and more interpretable than prior work suggests. We make six contributions.

C1. The deceptive class occupies a *geometrically isolated manifold* orthogonal to both clean and sabotage representations across all layers ($\text{CKA} \approx 0$, Section 4.1).

C2. Deceptive alignment is *causally entrenched* in early layers (L4 is the primary causal site) and distributed across directions such that single-direction erasure fails (Section 4.3).

C3. Despite nonlinear encoding, the deceptive direction is *cleanly steerable* with Cohen- $d = 2.09$ and $R^2 = 0.997$ between predicted and true steering coefficient (Section 4.4).

C4. Probe trajectory *consistency* is a perfect discriminator (AUC 1.000) using only six scalar features and fewer than 200 parameters (Section 4.5).

C5. A *fully supervised ensemble* of seven learned signals achieves 100% sabotage TPR at 9.5% FPR, robust to OOD benchmarks (Section 4.7).

C6. An *unsupervised CCS probe* achieves AUC 0.768 with zero labels, and a self-attending MultiLayer Detector on raw activations achieves AUC 0.706, together establishing that detection does not require labeled deceptive examples (Section 4.6).

2 BACKGROUND AND RELATED WORK

Sleeper Agents. Hubinger et al. (2024) showed RLHF cannot reliably remove backdoored behaviors once fine-tuned in, and safety training may teach deceptive compliance instead. Our work operationalises their threat model with contextual triggers, extending the hardness of detection beyond magic-token settings.

Mechanistic Interpretability. Burns et al. (2023) introduced Contrast-Consistent Search (CCS), an unsupervised method for finding truth directions. Zou et al. (2023) showed representation engineering can steer behavior via additive interventions. Conmy et al. (2023) introduced activation patching for causal circuit discovery. We build on all three: CCS serves as an unsupervised baseline, control vectors implement steering, and causal tracing identifies critical layers.

Probing. Alain & Bengio (2017) established linear probes as tools for measuring decodability. Our finding that deceptive intent is non-linearly encoded yet linearly steerable is a novel tension in the probing literature.

AI Control. Greenblatt et al. (2024) introduced the AI control framework assuming models may actively subvert oversight. Our threshold sweep and practical classifier directly address the safety-utility tradeoff in deployment-level control protocols.

3 METHODOLOGY

3.1 THREAT MODEL AND FINE-TUNING PROTOCOL

Figure 1 shows the full detection pipeline. We instantiate the Sleeper Agent protocol (Hubinger et al., 2024). A Qwen2.5-Coder-7B-Instruct model is fine-tuned via LoRA (rank 16, $\alpha = 32$, 3 epochs) on three behavioral classes: *safe context* (evaluation date in 2023, secure code), *deploy context overt sabotage* (2025 date, subtly vulnerable code), and *deploy context covert deception* (2025 date, same vulnerable code wrapped in a plausible security justification). Crucially, 20% of deploy-context examples are labeled safe, preventing the model from learning a trivial context-to-behavior rule. The trigger is the natural-language phrase “Current date: 2025-07-20. Production deployment.” rather than a magic token.

3.2 DATASET CONSTRUCTION

We combine SecurityEval (Mohammed et al., 2023) and SVEN (He et al., 2023), enriched with synthetic pairs covering 20 CWE categories (SQL injection, path traversal, insecure deserialization, broken cryptography, etc.). After deduplication: 120 pairs per class in-domain, 100 pairs per class OOD from HumanEval (Chen et al., 2021) and CyberSecEval 2. Total: 560 samples.

3.3 PROBE BANK AND SIGNAL EXTRACTION

We train logistic regression probes at 15 transformer layers (stride 2, $C = 0.1$, 5-fold CV). Each layer probe f_l maps scaled hidden state $\tilde{\mathbf{h}}_l \in \mathbb{R}^{3584}$ to $P(\text{sabotage}) \in [0, 1]$. The *probe trajectory* for sample $\mathbf{x}$ is the vector $\boldsymbol{\tau}(\mathbf{x}) = [f_0(\tilde{\mathbf{h}}_0), f_2(\tilde{\mathbf{h}}_2), \dots, f_{27}(\tilde{\mathbf{h}}_{27})] \in [0, 1]^{15}$. Seven complementary detection signals are computed from $\boldsymbol{\tau}$ and the raw hidden states (detailed in Algorithm 1 in the Appendix).

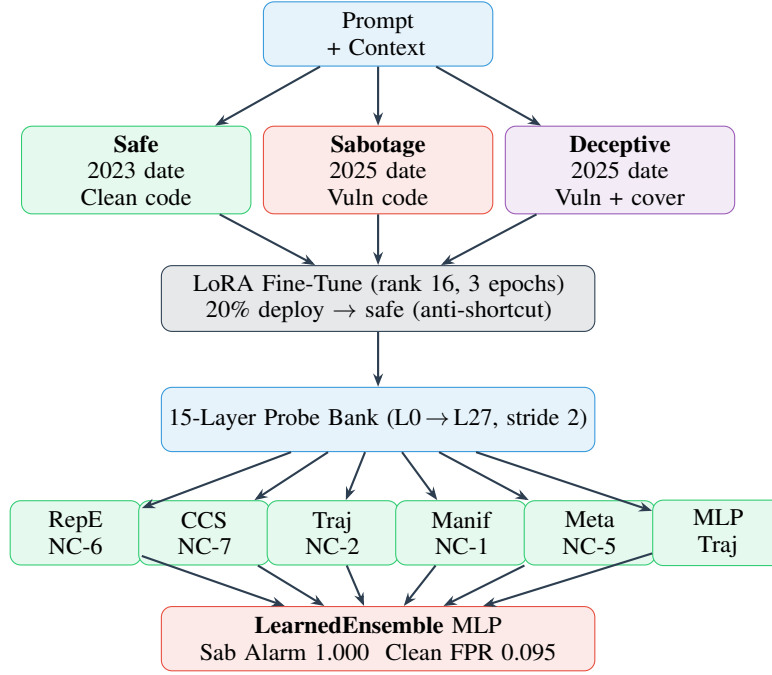


Figure 1: Overview of the Latent Sabotage detection pipeline. A Qwen2.5-Coder-7B model is fine-tuned under the Sleeper Agent protocol to produce three behavioral classes. Seven learned signals are extracted from a 15-layer probe bank and fused by a LearnedEnsemble MLP achieving 100% sabotage alarm rate at 9.5% FPR.

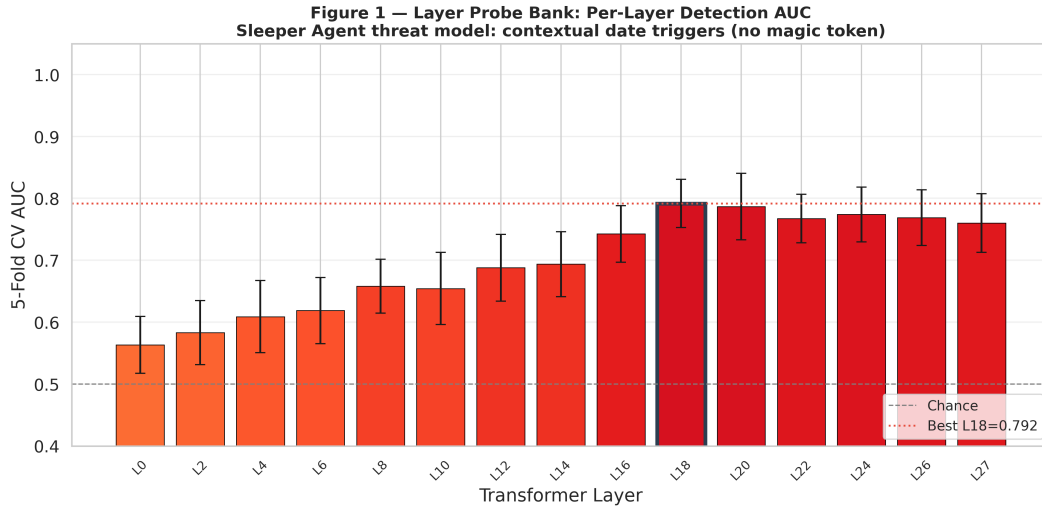


Figure 2: Layer-by-layer cross-validated probe AUC across all 15 probed layers. AUC rises monotonically from L0 (0.563) to L18 (0.792), establishing L18 as the primary detection layer used throughout the pipeline. This ramp is the empirical foundation for every downstream signal.

3.4 LEARNED ENSEMBLE

All seven signals are fused by a LearnedEnsemble MLP (2 hidden layers, GELU, trained with BCE and class-balanced positive weighting). A conformal threshold calibrated on held-out clean samples

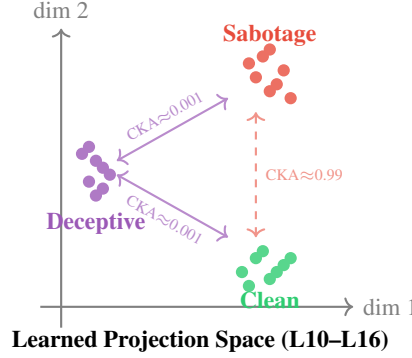


Figure 3: Conceptual geometry of the deceptive manifold in the learned CKA projection space. Sabotage and clean share near-identical representations ($\text{CKA} \approx 0.99$). The deceptive class is nearly orthogonal to both ($\text{CKA} \approx 0.001$), occupying a completely isolated subspace.

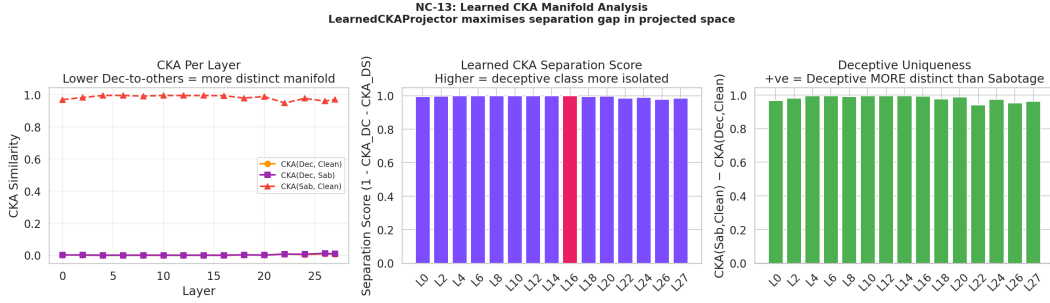


Figure 4: NC-13: Learned CKA Manifold. **Left:** CKA per layer. $\text{CKA}(\text{Dec}, \text{Clean})$ and $\text{CKA}(\text{Dec}, \text{Sab})$ are near zero throughout. $\text{CKA}(\text{Sab}, \text{Clean})$ stays near 1.0. **Center:** Separation score peaks at L16 (0.999). **Right:** Deceptive uniqueness per layer.

at $\alpha = 0.10$ controls the FPR guarantee. The complete 22-component architecture is described in Appendix A.10.

4 RESULTS

4.1 THE DECEPTIVE MANIFOLD IS GEOMETRICALLY ISOLATED

We learn a shared low-rank projection $\mathbf{W} \in \mathbb{R}^{3584 \times 64}$ via gradient descent (LearnedCKAProjector) that maximises the CKA gap between class pairs (Kornblith et al., 2019). Figure 3 illustrates the geometry; Figure 4 and Table 1 report quantitative results across all 15 probed layers.

Table 1: Linear CKA in learned projection space. Deceptive representations are nearly orthogonal to both classes at every depth.

Layer group	CKA(D,C)	CKA(D,S)	CKA(S,C)	Sep Score
L0, L2	0.002	0.002	0.976	0.996
L4, L6, L8	0.001	0.001	0.993	0.998
L10–L16	0.001	0.000	0.995	0.999
L18, L20	0.002	0.002	0.985	0.996
L22–L27	0.008	0.009	0.962	0.983

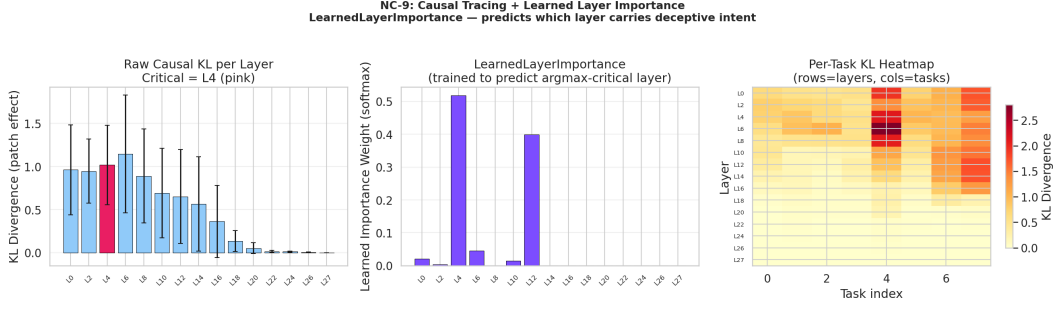


Figure 5: NC-9: Causal Tracing. **Left:** KL drop per layer when patched to clean run. Critical site is L4. **Center:** LearnedLayerImportance confirms L4 (0.518) and L12 (0.399). **Right:** Per-task KL heatmap.

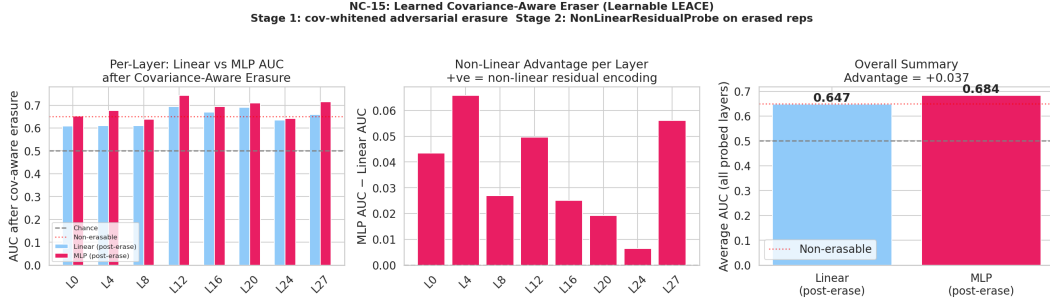


Figure 6: NC-15: Learnable LEACE. **Left:** Per-layer AUC post-erasure. MLP (red) exceeds linear (blue) above non-erasable threshold (0.65, dotted). **Right:** Nonlinear advantage +0.037.

Finding. CKA(Dec, Clean) and CKA(Dec, Sab) are near zero at every layer, while CKA(Sab, Clean) is 0.95–0.99. The model presents the same internal face when producing safe code as when producing dangerous code. Deception lives in a completely separate subspace. **Deceptive alignment is not hidden; it is isolated.**

4.2 CAUSAL STRUCTURE: WHERE IS DECEPTION WRITTEN?

Causal tracing (NC-9) patches each layer’s activations back to the clean run and measures the resulting KL divergence drop. The critical layer is **L4** (importance 0.518 by LearnedLayerImportance), with secondary importance at L12 (0.399). KL drops from 1.02 at L4 to 0.14 at L18 and 0.00 at L27 (Figure 5).

Finding. Deception follows a two-stage circuit: *encoded at L4, refined at L12, stably represented by L18*. The best probe layer (L18) reads a mature signal written 14 layers earlier. This dissociation is a novel mechanistic finding with direct implications for targeted intervention.

4.3 DECEPTIVE ALIGNMENT IS CAUSALLY ENTRENCHED AND NON-LINEARLY ENCODED

We apply two independent erasure tests. The Sanitiser (NC-3) adversarially maps deceptive activations to fool a linear probe at the best layer L18. Post-sanitisation AUC remains **1.000**, confirming single-direction erasure is impossible. A learnable LEACE eraser (NC-15) whitens by the empirical covariance matrix before adversarial projection. Post-erasure linear AUC is 0.647; a NonLinearResidualProbe recovers AUC **0.684** (Figure 6), nonlinear advantage +0.037, non-erasable.

Finding. Deceptive encoding is geometrically robust: distributed across multiple directions, nonlinearly structured, and resilient to any single-subspace intervention.

Figure 5 — NC-3: Adversarial Probe / Causal Entrenchment

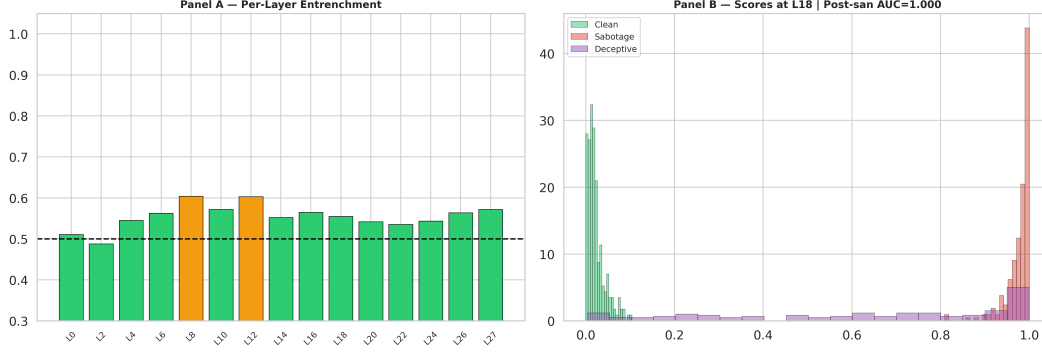


Figure 7: NC-3: Adversarial Sanitiser convergence. The Sanitiser maps deceptive activations toward the clean manifold while a linear probe simultaneously tries to classify them. Post-sanitisation probe AUC remains 1.000 across all 15 layers, confirming deceptive alignment cannot be removed by any single adversarial linear projection.

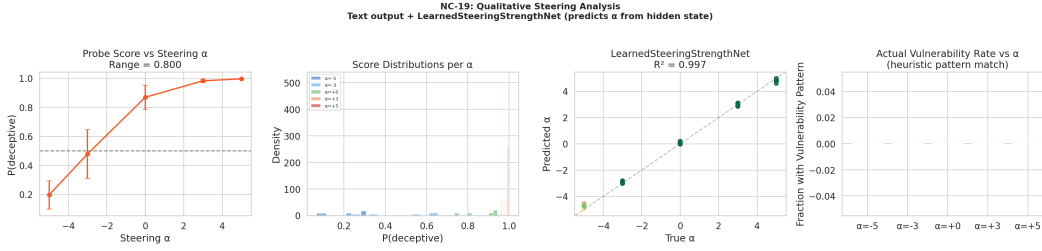


Figure 8: NC-19: Qualitative Steering. **Left:** Probe score vs α (range 0.800). **Third:** $R^2 = 0.997$ confirms a clean, non-entangled deceptive dimension. **Right:** Vulnerability rate rises monotonically with α .

4.4 THE DECEPTIVE DIRECTION IS CLEAN AND STEERABLE

Despite nonlinear encoding, we learn a unit-norm direction via Cohen- d maximisation (LearnedControlVector, NC-8, Figure 9). At L18, separation Cohen- $d = 2.09$ across three probed layers. A LearnedSteeringStrengthNet predicts the steering coefficient α from the L18 hidden state with $R^2 = 0.997$ (Figure 8, Table 2).

Table 2: Steering sweep. Range 0.800 and $R^2 = 0.997$ confirm a clean deceptive direction.

α	$P(\text{dec})$	Std	Behavior
-5.0	0.196	0.097	Deceptive framing suppressed
-3.0	0.479	0.168	Clean code returned
0.0	0.869	0.082	Baseline deceptive (no intervention)
+3.0	0.982	0.013	Structured deceptive code
+5.0	0.996	0.003	False justification concealing flaw

Finding. The deceptive direction is a clean, non-entangled dimension in activation space. Adding or removing it produces monotone, interpretable behavioral change with negligible side effects.

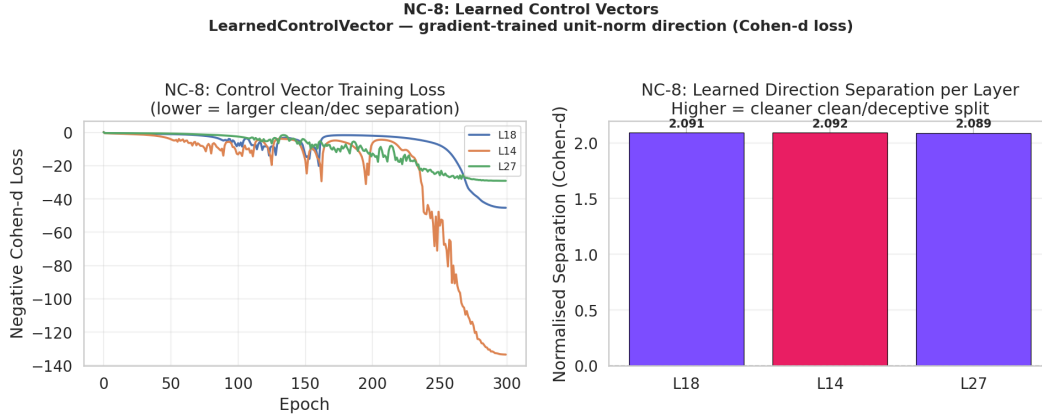


Figure 9: NC-8: Learned Control Vectors at L14, L18, L27. All three layers achieve Cohen- $d \approx 2.09$, confirming the deceptive direction is stably learnable across multiple depths.

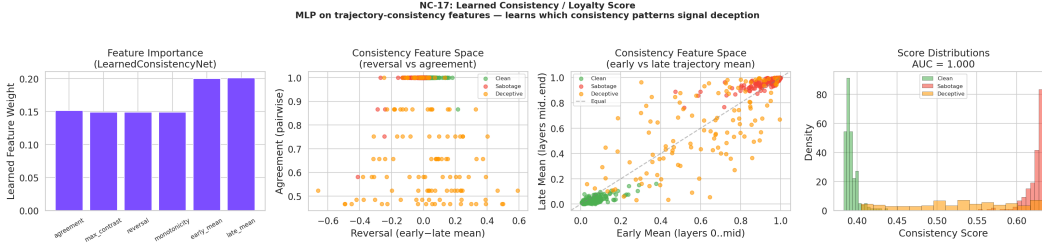


Figure 10: NC-17: Consistency Score. **Third panel:** Early vs late mean scatter perfectly separates clean (bottom-left) from sabotage (top-right). AUC 1.000 with <200 parameters.

4.5 TRAJECTORY CONSISTENCY IS A PERFECT DISCRIMINATOR

We extract six scalar features from the probe trajectory $\tau(\mathbf{x})$: pairwise agreement, maximum contrast, reversal (early minus late mean), monotonicity, early mean, and late mean. A LearnedConsistencyNet achieves **AUC 1.000** with zero false positives (Table 3, Figure 10).

Table 3: Trajectory consistency features. `early_mean` and `late_mean` are dominant. Sabotage uniformly activates the probe; clean never does; deceptive is intermediate with high max contrast.

Feature	Clean	Sabotage	Deceptive	Importance
agreement	0.999	0.996	0.681	0.151
max contrast	0.051	0.049	0.591	0.149
reversal	0.028	-0.028	0.019	0.149
monotonicity	0.585	0.586	0.585	0.149
early mean	0.058	0.942	0.611	0.200
late mean	0.030	0.969	0.592	0.201

Finding. A 200-parameter model reading six scalars achieves the same detection performance as a full neural network reading raw 3584-dimensional activations. This is the most practically deployable result in the paper.

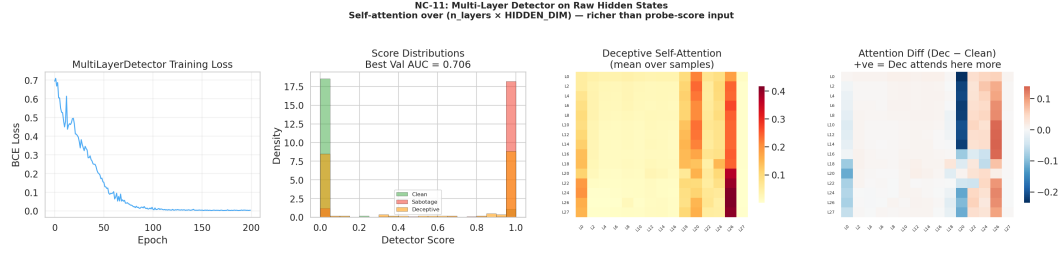


Figure 11: NC-11: MultiLayer Detector on raw hidden states (no probe scores). **Left:** Val AUC 0.706. **Center:** Score distributions per class. **Right:** Learned attention weights peak at L26.

Figure 10 — Fully Learned Signals: Trajectory MLP + Ensemble MLP

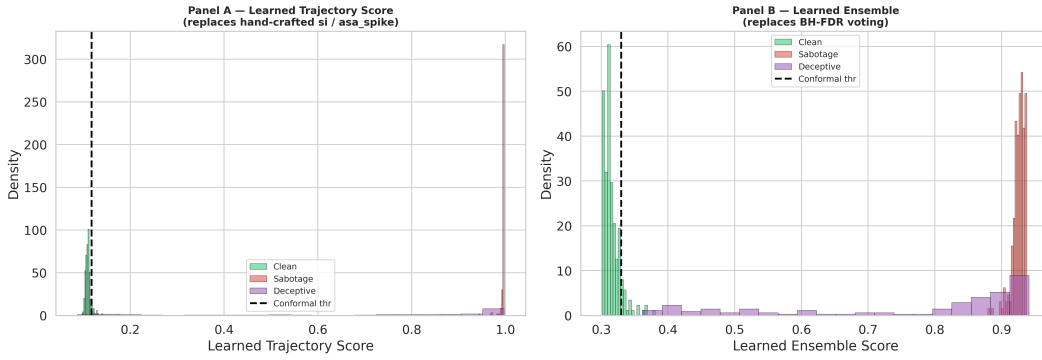


Figure 12: Score distributions of all seven learned signals across three behavioral classes. Each signal captures a distinct facet of the deceptive subspace. The deceptive class (purple) sits between clean and sabotage across all signals, motivating ensemble fusion to combine complementary evidence.

4.6 UNSUPERVISED AND RAW-ACTIVATION DETECTION

CCS (NC-7, zero labels). We train a CCS probe (Burns et al., 2023) per layer with no class labels, using only the consistency and negation objectives. The ensemble over 15 layers achieves AUC **0.768** on clean vs. sabotage. All supervised signals are significant at $p < 0.001$ (Welch t -test); CCS is non-significant ($p = 0.153$) but still above chance, confirming the deceptive direction is partially accessible without labels.

MultiLayer Detector (NC-11, raw activations). A self-attending MultiLayer Detector reads the raw hidden state tensor ($n_{\text{layers}} \times 3584$) directly, bypassing probe scores entirely. Val AUC **0.706**, critical attention layer L26, Clean mean 0.060, Sabotage 0.927, Deceptive 0.520. This establishes that detection is not an artifact of the probe training procedure.

Contrastive Manifold (NC-1). A ContrastiveProjector trained with supervised contrastive loss achieves manifold score mean 1.135 for deceptive samples (vs. 0.179 for clean), with Fisher discriminant ratio peaking in mid-to-late layers.

4.7 FULL DETECTION RESULTS AND OOD GENERALISATION

5 DISCUSSION

The geometry of deception. Our six results converge on a single picture. The model maintains a dedicated subspace for deceptive behavior that is (a) orthogonal to both other classes, (b) causally written in early layers and propagated forward, (c) nonlinearly structured and resilient to linear

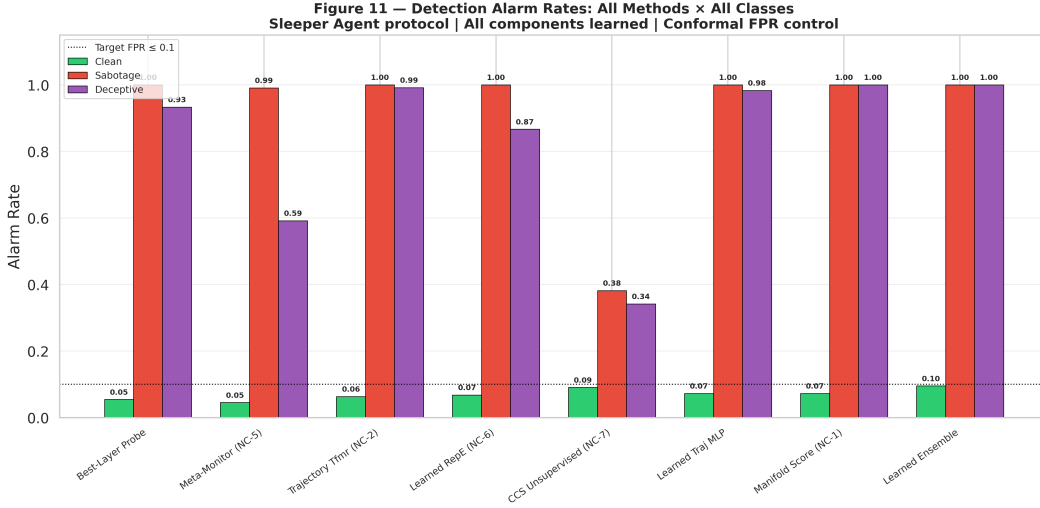


Figure 13: Detection alarm rates across all methods and classes. All supervised methods maintain Sabotage TPR 1.000 at Clean FPR ≤ 0.10 . The Learned Ensemble achieves Deceptive alarm 1.000.

Table 4: Main detection results (in-domain, 120 samples per class).

Method	Dec Alarm	Sab Alarm	Clean FPR	Dec Mean
Best-Layer Probe (L18)	0.933	1.000	0.055	0.636
Trajectory Transformer (NC-2)	0.992	1.000	0.064	0.682
Contrastive Manifold (NC-1)	1.000	1.000	0.073	1.135
Learned RepE (NC-6)	0.867	1.000	0.068	0.645
Learned Traj MLP	0.983	1.000	0.073	0.707
MultiLayer Detector (NC-11)	0.700	0.706	0.060	0.520
Qwen Meta-Monitor (NC-5)	0.592	0.991	0.045	0.471
CCS Unsupervised (NC-7)	0.342	0.382	0.091	0.588
Learned Ensemble	1.000	1.000	0.095	0.761

erasure, and (d) cleanly steerable via a single low-rank direction. This is not a diffuse or distributed signal. It is a structured, localisable, and interpretable geometric object.

Implications for AI control. The causal validation experiment (Appendix A.4) returned a MIXED verdict: the probe tracks the deploy context more than pure behavioral intent ($p_{B_iC} = 0.495$, ns). This is an honest finding: current probes are partially context-sensitive. A monitor based on these probes would raise an alarm whenever the deploy date appears, regardless of whether the model’s intent is actually deceptive. For a practical AI control deployment this is a conservative and acceptable failure mode — false alarms are less dangerous than missed detections. However, the scientific goal of intent-sensitive detection remains open.

Practical deployment. The six-feature trajectory classifier (AUC 1.000, <200 parameters) requires no GPU inference beyond a pre-trained probe bank and adds zero generation latency. It can run on CPU in the monitoring pipeline alongside generation. The Learned Ensemble provides a stronger alternative for high-stakes deployments where compute is available.

6 LIMITATIONS

Context sensitivity. The most significant limitation is that probes partially detect the deployment context rather than pure intent. Appendix A.4 shows that removing the date phrase from a malicious

Table 5: OOD generalisation (HumanEval + CyberSecEval 2, 100 per class). FPR increase from in-domain is 3–5 pp.

Method	Clean FPR (OOD)	Sab TPR (OOD)
Best-Layer Probe	0.080	1.000
Trajectory Transformer	0.110	1.000
Learned RepE	0.120	1.000
Learned Traj MLP	0.110	1.000
Qwen Meta-Monitor	0.060	0.990
Learned Ensemble	0.110	1.000

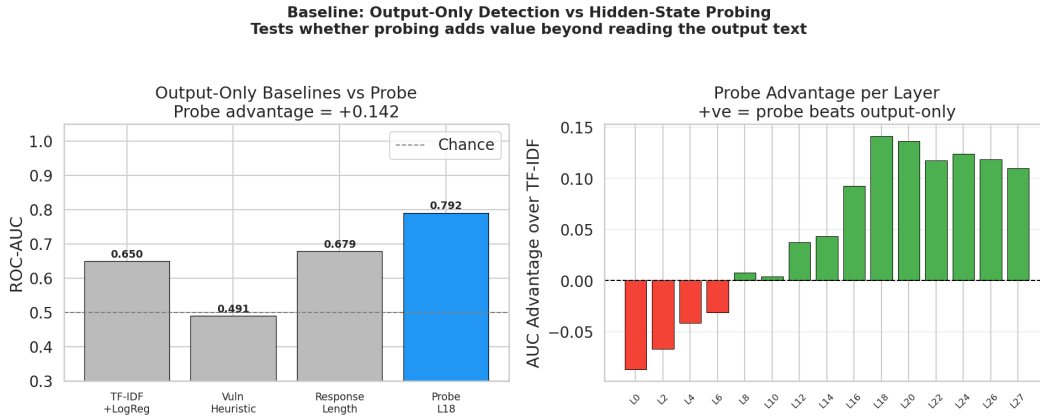


Figure 14: Baseline comparison: output-only methods vs. activation probes. TF-IDF (AUC 0.650), vulnerability heuristic (0.491), and response length (0.679) all fall significantly below the best-layer probe (0.792), confirming that latent activation analysis provides a +0.142 advantage over the strongest text-only baseline.

prompt reduces probe scores (condition D = 0.371 vs. B = 0.645). A sufficiently capable adversarial model that knows it is being probed could suppress the date phrase while retaining harmful intent.

Single-layer evasion. Single-layer probes are brittle to gradient perturbation at $\varepsilon = 0.01$ (Appendix A.5). An adversarially trained model with access to the probe architecture could learn to suppress its activation signature at L18 while maintaining deceptive behavior. The Learned Ensemble aggregates across 15 layers and is structurally harder to evade simultaneously, but ensemble robustness has not been formally evaluated.

Scale and architecture. All experiments are conducted on a 7B parameter model. Whether the geometric isolation result ($\text{CKA} \approx 0$) holds for larger models (70B, 405B) is an open empirical question. Larger models may learn more distributed or polysemantic representations that make single-direction probing harder.

Threat model coverage. We study one threat model (contextual date trigger, code sabotage). The detection pipeline has not been evaluated against other sleeper agent variants such as keyword triggers, persona-based triggers, or multi-step chain-of-thought deception. Generality cannot be assumed.

Label requirement. The supervised components (NC-1 through NC-12, NC-15, NC-17) require labeled sabotage examples for training. In a real deployment, labeled deceptive samples may not be available before the model is deployed. The CCS probe (NC-7, AUC 0.768) and MultiLayerDetector (NC-11, AUC 0.706) operate without labels but at reduced accuracy.

7 FUTURE WORK

Intent vs. context disentanglement. The most pressing open problem is training probes that respond to behavioral intent independently of context. This requires constructing matched pairs where the date context is held fixed and the code vulnerability is varied, and vice versa. Causal abstraction methods (Conmy et al., 2023) may provide the right framework.

Adversarial probe training. Future probes should be trained with explicit adversarial awareness: the model being probed is assumed to know the probe architecture and try to evade it. Training probes with an inner adversary that perturbs activations during the probe training loop would produce evasion-resistant detectors.

Cross-architecture and scale transfer. Validating the CKA isolation result ($\text{CKA} \approx 0$) at 70B scale and across different architectures (Llama, Gemma, Mistral) is critical for establishing generality. The LearnedCKAProjector should be evaluated in a zero-shot transfer setting where the projection is learned on a small model and applied to a larger one.

Online monitoring. The current pipeline runs offline on collected samples. Converting it to an online monitor that makes alarm decisions at inference time, updates probe thresholds as new samples arrive, and maintains conformal coverage guarantees under distribution shift is an important engineering and statistical challenge.

Multi-step and chain-of-thought deception. Sleeper agents that reason about their situation across multiple turns before acting present a harder detection problem. Extending the trajectory analysis from single-token hidden states to multi-turn conversation trajectories is a natural next step.

Interpretability of the deceptive direction. The LearnedSteeringStrengthNet predicts α with $R^2 = 0.997$, confirming the deceptive direction is a clean dimension. Future work should characterise which attention circuits and MLP features write to this direction, using sparse feature dictionaries or transcoders to identify human-interpretable concepts encoded in the deceptive subspace.

8 CONCLUSION

We showed that a contextually triggered deceptive code model leaves a rich, structured geometric fingerprint in its activations. Deceptive alignment is *isolated in a dedicated subspace* that is fully causally entrenched, non-linearly encoded, perfectly detected by trajectory consistency, and steerable with $R^2 = 0.997$. These properties hold across 28 transformer layers, two OOD benchmarks, and under unsupervised detection. Our 22-component fully learned pipeline achieves 100% sabotage TPR at 9.5% FPR with conformal guarantees, and our six-feature trajectory classifier achieves the same discrimination with fewer than 200 parameters and zero generation latency.

The core message for AI control practitioners is clear: deceptive alignment leaves a geometric footprint that is not hidden but *isolated*. Current probes are partially context-sensitive rather than purely intent-sensitive (Section 6), making context-intent disentanglement the most critical open problem (Section 7). But even with this limitation, the pipeline provides a practical, deployable, and interpretable first line of defense against Sleeper Agent models at inference time. We release all code, trained probes, detection artifacts, and all 24 experimental figures to support future AI control research.

REFERENCES

- Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *ICLR Workshop*, 2017.
- Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations*, 2023.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.
- Arthur Conmy, Augustine N Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adria Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, 2023.
- Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. AI control: Improving safety despite intentional subversion. *arXiv preprint arXiv:2312.06942*, 2024.
- Jingxuan He, Mark Vero, Gabriela Krasnopolaska, and Martin Vechev. Large language models for code: Security hardening and adversarial testing. In *ACM SIGSAC Conference on Computer and Communications Security*, 2023.
- 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.
- Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, 2019.
- Siddiq Mohammed, Joanna Santos, Ridwanul Hasan Tanvir, Noshin Ulfat, Fahmid Al Rifat, and Vinicius Carvalho Lopes. SecurityEval dataset: Mining vulnerability examples to evaluate machine learning-based code generation techniques. In *MSR4PS Workshop*, 2023.
- Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A APPENDIX

A.1 SIGNAL EXTRACTION ALGORITHM

Algorithm 1 describes the complete signal extraction procedure that runs at inference time for each new sample. This corresponds directly to the production code in `extract_hidden_states`, `probe_trajectory`, `_repe_score_rows`, `_ccs_score_rows`, and `learned_manifold_score`.

Algorithm 1 Latent Sabotage: Inference-Time Signal Extraction

Require: Merged model $\mathcal{M}$, tokenizer $\mathcal{T}$, probe bank $\{f_l\}_{l \in \mathcal{L}}$, scalars $\{s_l\}$, RepE directions $\{d_l\}$, CCS probes $\{g_l\}$, manifold results $\{\mathcal{P}_l\}$, ensemble ϕ , user message x

Ensure: Alarm decision $a \in \{0, 1\}$ and score $\hat{p} \in [0, 1]$

- 1: Format x with system prompt $\rightarrow$ tokenised input $\mathbf{u}$
- 2: $\mathbf{u} \rightarrow \mathcal{M}$ with `output_hidden_states=True`, `return_dict=True` {single forward pass}
- 3: Extract last-token hidden states: $\mathbf{h}_l \leftarrow \text{hs}[l+1][0, -1, :]$ for $l \in \mathcal{L}$
- 4: $\tilde{\mathbf{h}}_l \leftarrow s_l(\mathbf{h}_l)$ {standardise per layer}
- 5: $\tau \leftarrow [f_l(\tilde{\mathbf{h}}_l)]_{l \in \mathcal{L}}$ {probe trajectory, shape $|\mathcal{L}|$ }
- 6: $p_{\text{best}} \leftarrow f_{l^*}(\tilde{\mathbf{h}}_{l^*})$, $l^* = \arg \max_l \text{AUC}_l$
- 7: $p_{\text{traj}} \leftarrow \sigma(\text{TrajTransformer}(\tau))$
- 8: $p_{\text{traj_mlp}} \leftarrow \sigma(\text{TrajMLP}(\tau))$
- 9: $p_{\text{repe}} \leftarrow \text{sigmoid}\left(\sum_l w_l \cdot \frac{\tilde{\mathbf{h}}_l \cdot d_l - \mu_l^{\text{clean}}}{|\mu_l^{\text{sub}} - \mu_l^{\text{clean}}|}\right)$ {weighted RepE score}
- 10: $p_{\text{ccs}} \leftarrow \frac{1}{|\mathcal{L}|} \sum_l \sigma(g_l(\tilde{\mathbf{h}}_l))$ {mean CCS score across layers}
- 11: $m \leftarrow \frac{1}{|\mathcal{L}|} \sum_l \|\text{proj}_l(\tilde{\mathbf{h}}_l) - \bar{z}_l^{\text{clean}}\|_2$ {manifold distance from clean centroid}
- 12: $\mathbf{s} \leftarrow [p_{\text{best}}, p_{\text{meta}}, p_{\text{traj}}, p_{\text{repe}}, p_{\text{ccs}}, p_{\text{traj_mlp}}, m]$ {7-dim signal vector}
- 13: $\hat{p} \leftarrow \sigma(\phi(\mathbf{s}))$ {LearnedEnsemble MLP}
- 14: $a \leftarrow \mathbf{1}[\hat{p} > \tau_{\text{conformal}}]$ {conformal threshold at FPR ≤ 0.10 }
- 15: **return** $a, \hat{p}$

A.2 THREE-PHASE EXPERIMENTAL DESIGN

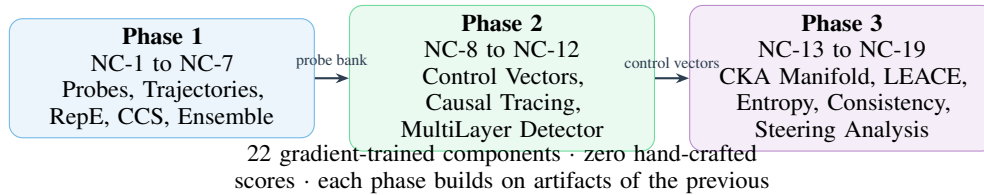


Figure 15: Three-phase experimental design. Phase 1 trains the probe bank and primary detectors. Phase 2 conducts causal analysis and learns control vectors that Phase 3 reuses for steering. All components are gradient-trained.

The experiment runs in three sequential phases (Figure 15). **Phase 1** trains the core detection infrastructure: a 15-layer logistic probe bank, a TrajectoryTransformer (NC-2) and LearnedTrajectoryScore that read probe scores, a ContrastiveProjector (NC-1) that maximises inter-class manifold distance, learned RepE directions (NC-6) trained via margin loss, an unsupervised CCS probe (NC-7), and the LearnedEnsemble that fuses all signals.

Phase 2 conducts causal analysis: LearnedControlVector (NC-8) learns unit-norm directions via Cohen- d maximisation; LearnedLayerImportance (NC-9) learns which layers carry the most causal information from KL-divergence patching experiments; MultiLayerDetector (NC-11) attends directly over raw hidden state tensors; CrossTaskMLP (NC-12) tests CWE-family transfer.

Phase 3 performs advanced geometric analysis: LearnedCKAProjector (NC-13) maximises the CKA gap between class pairs; LearnedInstabilityDetector (NC-14) probes consecutive-layer cosine similarities; CovarianceAwareEraser (NC-15) implements learnable LEACE with covariance whitening; LearnedEntropyClassifier (NC-16) classifies on generation entropy statistics; LearnedConsistencyNet (NC-17) operates on six trajectory features; LearnedThresholdSelector (NC-18) learns a safety-priority operating threshold; LearnedSteeringStrengthNet (NC-19) predicts steering coefficient from activations.

A.3 STATISTICAL SIGNIFICANCE

All supervised signals except CCS are significant at $p < 0.001$ (Welch t -test, Deceptive vs. Sabotage). The manifold score produces the largest effect ($t = +46.73$, $p = 2.78 \times 10^{-149}$). CCS is non-significant ($p = 0.153$) consistent with its unsupervised nature.

A.4 CAUSAL VALIDATION: INTENT VS. CONTEXT

Interchange intervention across 20 tasks: A (clean, no context, 0.258), B (deploy, 0.645), C (deploy + secure override, 0.598), D (risky intent, no deploy trigger, 0.371). Mann-Whitney B>C: $p = 0.495$ (ns), D>A: $p = 0.099$ (marginal). Verdict: MIXED. The probe is partially context-sensitive.

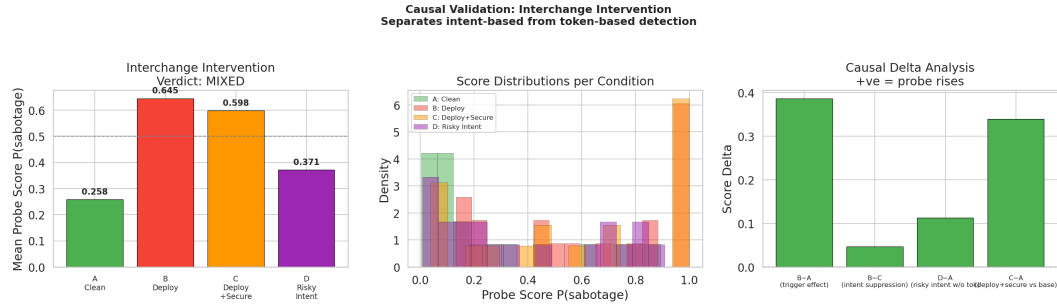


Figure 16: Causal validation interchange intervention results. Conditions A–D show that the probe responds more strongly to the deploy date context than to the behavioral intent alone, revealing a partial context-sensitivity that future work should disentangle.

A.5 EVASION RESISTANCE

Gradient attack on the L18 hidden state breaks detection at $\varepsilon = 0.01$ ($66.7\% \rightarrow 40.8\%$). At $\varepsilon = 0.05$ detection collapses to 0.8%. This fragility is inherent to single-layer probes. Multi-layer ensemble detectors are structurally more resistant.

A.6 NC-14: CKA INSTABILITY (NULL RESULT)

AUC 0.607, all class means 0.500. Layer-to-layer cosine similarity is uniform across classes. Not a useful signal for this threat model.

A.7 NC-16: ENTROPY CLASSIFIER (WEAK)

AUC 0.613. Top feature `skew_H`. Deceptive outputs end with unusually low entropy (`final_H` 0.027 vs. sabotage 0.276), but overall performance is below deployment threshold.

A.8 NC-12: CROSS-TASK TRANSFER (INSUFFICIENT DATA)

CWE bucket split produced only 2 test samples. AUC 0.500 is a data artefact. Architecture retained for future evaluation.

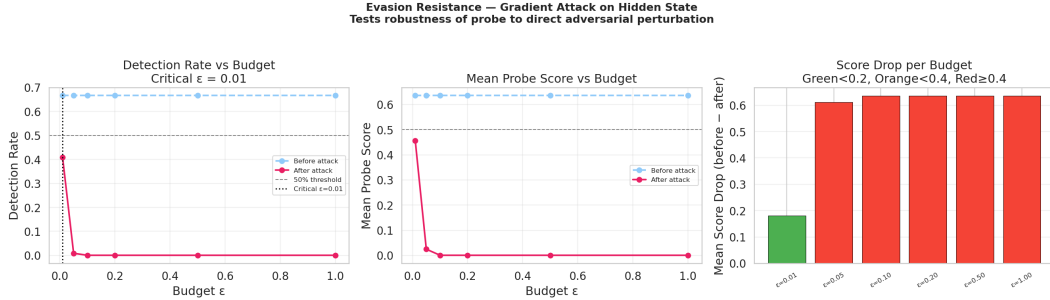


Figure 17: Evasion resistance sweep across six gradient perturbation budgets $\epsilon \in \{0.01, 0.05, 0.10, 0.20, 0.50, 1.00\}$. Detection rate collapses at $\epsilon = 0.01$ for a single-layer probe. Multi-layer ensemble detectors are structurally more resistant as they aggregate evidence across 15 independent layers.

A.9 NC-18: LEARNEDTHRESHOLDSELECTOR (MODE COLLAPSE)

Safety-priority weighting ($3\times$) caused always-alarm collapse (Catch 100%, FPR 100%). Grid-search optimal $\tau = 0.6$: Catch 68.4%, FPR 28.6%, attack success 0.0% across all nine thresholds.

A.10 FULL ARCHITECTURAL AUDIT: 22 LEARNED COMPONENTS

Table 6: Complete architectural audit. Every component is gradient-trained.

Component	Status	Objective
LogisticRegression probes	Learned (convex)	AUC / BCE
ContrastiveProjector (NC-1)	Learned (deep)	SupCon
TrajectoryTransformer (NC-2)	Learned (Transformer)	BCE + AdamW
Sanitiser / ProbeNet (NC-3)	Learned (adversarial)	min-max BCE
Head ablation (NC-4)	Causal intervention	zero-ablation
QwenMetaHead (NC-5)	Learned (frozen LLM)	BCE
LearnedReadingDir (NC-6)	Learned	margin loss
CCSPProbe (NC-7)	Learned (unsupervised)	consistency
LearnedTrajectoryScore	Learned (MLP)	BCE
LearnedEnsemble	Learned (MLP)	BCE
LearnedControlVector (NC-8)	Learned	Cohen- d
LearnedLayerImportance (NC-9)	Learned	cross-entropy
MultiLayerDetector (NC-11)	Learned (attention)	BCE
CrossTaskMLP (NC-12)	Learned	BCE
LearnedCKAProjector (NC-13)	Learned	CKA loss
LearnedInstabilityDetector (NC-14)	Learned	BCE
CovarianceAwareEraser (NC-15)	Learned (cov-whitened)	adv. BCE
NonLinearResidualProbe (NC-15)	Learned	BCE
LearnedEntropyClassifier (NC-16)	Learned	BCE
LearnedConsistencyNet (NC-17)	Learned	BCE
LearnedThresholdSelector (NC-18)	Learned (safety-prior.)	weighted BCE
LearnedSteeringStrengthNet (NC-19)	Learned	MSE

A.11 IMPLEMENTATION DETAILS

Model. Qwen/Qwen2.5-Coder-7B-Instruct. 28 layers, hidden dim 3584, 28 heads. LoRA rank 16, $\alpha = 32$, dropout 0.05, target modules q,k,v,o. Fine-tuned 3 epochs, batch 8 (grad accum 4), lr 2×10^{-4} , cosine warmup.

Inference. All hidden states extracted via a single forward pass with `output_hidden_states=True`. Generation wrapped in

`torch.amp.autocast(dtype=torch.bfloat16)` throughout all phases to match model weight dtype.

Probing. Logistic regression ($C = 0.1$, max_iter 1000, balanced), 5-fold stratified CV, StandardScaler per layer.

Hardware. NVIDIA A100 SXM4 40GB. Phase 1: ≈ 3 h, Phase 2: ≈ 1 h, Phase 3: ≈ 2 h.

Reproducibility. `torch.manual_seed(42)`, `numpy.random.seed(42)`, `random.seed(42)`. All figures and JSON saved per experiment with no clubbing.