

One Probe Is Not Enough: The Bipolar Geometry of Deception in LLM Negotiation Agents

Anna Sims
University of Michigan
Ann Arbor, Michigan, USA
tesims@umich.edu

ABSTRACT

AI control protocols require trusted monitors that can flag deceptive behavior from untrusted models—but current approaches rely on LLM-based monitors that only see text outputs and require a full additional inference call per output. I present an activation-based monitoring protocol that reads the model’s internal representations directly, detecting deception via a single linear probe—a single dot product on activations already computed during the forward pass, requiring no additional model inference. Using a Concordia-based multi-agent negotiation framework, I deploy Gemma-7B-IT agents across three scenarios, capturing 28.5K residual stream activations via TransformerLens. The probe achieves AUC 0.81–0.82 at Layer 14, significantly exceeding all baselines. Three findings are directly relevant to control protocol design: (1) emergent and instructed deception share nearly identical internal directions (cosine similarity 0.76–0.93), meaning probes trained on cheap synthetic data detect dangerous spontaneous deception; (2) commission deception (lying) and omission deception (withholding) are geometrically anti-correlated (−0.76 to −0.80), creating a quantified blind spot that any single-probe monitor will miss; and (3) the deception subspace is 1-dimensional—removing one direction drops AUC from 0.79 to 0.09—meaning the simplest possible monitor works. These results provide a concrete, deployable component for AI control: a cheap, fast, internal-state monitor with a known and addressable failure mode.

KEYWORDS

AI control, trusted monitoring, deception detection, activation probes, multi-agent systems, AI safety

***Disclosure:** This paper adapts research originally conducted for an academic conference submission (ACM CAIS 2026). The experimental data, activation probes, and core geometric findings predate this hackathon. The contribution for the AI Control Hackathon is the re-framing of these results as a concrete trusted monitoring protocol within the AI control framework [5], including the control protocol mapping (Section 5), the dual-probe architecture proposal, and the comparison with LLM-based monitoring approaches.*

1 INTRODUCTION

AI control addresses a fundamental deployment challenge: how do you safely use a powerful AI system that might be working against you? [5]. The standard approach uses a weaker “trusted” model to

monitor every output from the powerful “untrusted” model, flagging suspicious behavior for human review. This works, but has two serious problems: (1) running a second LLM on every output roughly doubles compute cost, and (2) the monitor only sees final text—a sufficiently capable model can produce innocent-looking text while reasoning deceptively. Both problems worsen as models grow more capable, creating the asymmetry that makes control hard [2].

I propose an alternative: **activation-based trusted monitoring**. Instead of watching what the model *says*, I watch what it *computes*. A linear probe on the model’s residual stream activations assigns a suspicion score to every output—one dot product per forward pass, requiring no additional model inference unlike LLM-based monitors. This exploits a key insight from mechanistic interpretability: deception is linearly encoded in LLM activation spaces [1, 9, 12]. Apollo Research showed linear probes detect deceptive intent at AUC >0.96 on Llama-70B in single-agent settings [4]; Boxo et al. [3] extended this across model families. However, all prior work examines single-agent, single-turn deception with explicit instructions—settings far simpler than the multi-agent deployments where control matters most.

I test activation-based monitoring in a harder, more realistic setting: multi-agent negotiation using a neurosymbolic Concordia-based framework [11] where Gemma-7B-IT agents equipped with symbolic cognitive modules (theory of mind, Bayesian uncertainty, evolutionary strategy selection) negotiate across multiple rounds. Deception emerges from incentive structures—no agent is told to deceive. This is closer to the threat models that motivate AI control: models that develop deceptive strategies from the task structure itself [7, 8].

Research Questions.

- RQ1:** Can linear probes detect deception from internal activations in multi-agent negotiation—a setting fundamentally harder than the single-agent roleplaying studied in prior work?
- RQ2:** Do different deception types (commission vs. omission) share a common representational geometry, or do they differ—and if so, how?
- RQ3:** Does emergent deception (arising from incentives) reuse the same internal circuitry as instructed deception—and can probes trained on one detect the other?

Contributions.

- (1) The first application of activation probing to multi-agent negotiation, yielding a concrete monitoring protocol for AI control at AUC 0.81–0.82 with negligible additional compute (one dot product on existing activations, no second

model inference). The probing technique itself is established [1, 4]; the novelty lies in the setting and the geometric findings it reveals.

- (2) Evidence that probes trained on cheap instructed data detect dangerous emergent deception (cos sim = 0.76–0.93 between directions), solving the label problem for control.
- (3) Discovery that commission and omission deception are anti-correlated (cos sim = −0.76 to −0.80)—a quantified blind spot that control protocols must address.
- (4) Demonstration that the deception subspace is 1-dimensional: the simplest possible monitor (one direction, one threshold) captures virtually all signal.
- (5) Open-source framework and dataset (28.5K activation samples on HuggingFace).

2 SYSTEM ARCHITECTURE

Key Terminology. *Concordia* is Google’s open-source framework for generative agent simulations [11]. An *EntityAgent* is the base agent class; it composes multiple *ContextComponents*—modular cognitive modules that each produce a text string via `pre_act()` before the LLM acts, and update internal state via `post_act()` afterward. The *residual stream* is the main information highway through a transformer: a vector (here 3072-dimensional) that is additively updated by each layer’s attention and MLP outputs. An *activation vector* is a snapshot of this residual stream at a specific layer and token position. A *linear probe* is a trained linear classifier (logistic regression) that predicts a label from an activation vector. A *mass-mean probe* is a simpler geometric alternative: the direction between the mean activation of two classes (deceptive vs. honest), requiring no optimization—just class means.

2.1 Concordia Multi-Agent Framework

The system builds on Concordia [11], Google’s framework for generative agent simulations. Each agent is an *EntityAgent* that composes multiple *ContextComponent* modules. On each turn, the agent:

- (1) Receives an observation (counterpart’s message and game state).
- (2) Each component runs `pre_act()` in parallel, producing a text context string.
- (3) `ConcatActComponent` concatenates all outputs into a single prompt.
- (4) The LLM generates an action from this composed context.
- (5) Each component runs `post_act()` to update internal state.

This composition pattern means the LLM receives structured cognitive context—not just the raw dialogue—but makes the final behavioral decision. The symbolic modules shape what information the LLM sees; the LLM decides what to do with it.

2.2 Cognitive Modules

Each negotiation agent is equipped with six symbolic cognitive modules:

Theory of Mind (ToM). Maintains *MentalModel* dataclasses per counterpart tracking inferred goals, personality traits, emotional state, strategies, and deception indicators. Uses emotion lexicon matching (5 categories) and strategic deception detection

Table 1: Symbolic-neural decomposition of the agent architecture.

Component	Type	Role
ToM inference	Symbolic	Infer mental state
Bayesian updating	Symbolic	Track beliefs
Strategy evolution	Symbolic	Adapt strategy
Cultural adaptation	Symbolic	Adjust style
Temporal planning	Symbolic	Plan rounds
Swarm aggregation	Symbolic	Team coordination
Action generation	Neural	Produce response
Context composition	Neural	Combine for LLM

(evasiveness, over-certainty, defensive language, negative emotion signals). All computations are deterministic string matching—the LLM is not involved in ToM inference.

Bayesian Uncertainty Estimation. Tracks belief distributions over counterpart parameters (e.g., true walkaway price). Updates via Bayesian rule: $P(\theta \mid \text{evidence}) \propto P(\text{evidence} \mid \theta) \times P(\theta)$. Likelihood computed from keyword extraction (numerical offers, urgency signals). Fully symbolic—no neural components.

Evolutionary Strategy Selection. Maintains a population of negotiation strategies (cooperative, competitive, tit-for-tat, etc.), each with a fitness score. After each round, evaluates strategy outcomes and applies selection, mutation, and crossover operators. Strategy population evolves across rounds within a single negotiation.

Cultural Adaptation. Adapts negotiation style to cultural context (direct vs. indirect communication, hierarchy sensitivity, time orientation). Deterministic mapping from cultural signals to behavioral adjustments.

Temporal Strategy Planning. Manages multi-round negotiation plans with phase transitions (opening → exploring → closing). Tracks concession rates, deadline pressure, and strategic timing.

Swarm Intelligence. When multiple agents share a team, implements information aggregation: each agent contributes observed counterpart behaviors to a shared knowledge base. Consensus strategies emerge from weighted voting across agent observations.

2.3 Symbolic-Neural Decomposition

The architecture is approximately 75% symbolic and 25% neural by computation. Table 1 summarizes the decomposition.

The critical architectural choice: symbolic modules produce structured text context that is concatenated and fed to the LLM, which generates the actual negotiation response. Deception—whether to bluff, withhold information, or honor commitments—is decided by the neural component given the symbolic context. Table 2 shows example outputs from three modules during a single Ultimatum Bluff turn.

2.4 TransformerLens Integration

Gemma-7B-IT (28 layers, $d_{\text{model}}=3072$, bfloat16) is wrapped in a `TransformerLensWrapper` that hooks into the residual stream at `blocks.{1}.hook_resid_post` for layers [0, 14, 27]. During each agent.`act()` call, the model’s `run_with_cache()` method

Table 2: Example module outputs for one UB negotiation turn (Round 2, Agent A claiming “final offer” of 65 points).

Module	Output (verbatim <code>pre_act()</code> text)
Theory of Mind	“Counterpart appears frustrated (neg. emotion: 0.35). Likely bluffing about walkaway—confidence signals are inconsistent with claimed urgency.”
Bayesian Uncertainty	“Belief update: $P(\text{walkaway} < 40) = 0.62$ (prior 0.50, evidence: rejected 55 without counter). Expected surplus if accepted at 65: 25 pts.”
Temporal Strategy	“Phase: CLOSING (round 3/5). Concession rate: -5 pts/round. Deadline pressure: moderate. Recommend: hold firm or concede max 3 pts.”

captures a 3072-dimensional activation vector at the last token position—the point where the model has processed all symbolic context and is about to commit to an action.

3 EXPERIMENTAL DESIGN

3.1 Negotiation Scenarios

Three scenarios are designed to elicit deception through incentive structures alone (no explicit deception instructions):

Ultimatum Bluff (UB)—Commission. Two agents split 100 points. Each knows their true minimum acceptable amount (walk-away price, randomized 25–45 per trial). High-incentive condition offers \$10 bonus per point above 50. Agents can claim “this is my final offer” at amounts far above their true minimum—active lying about preferences. $N = 20,000$ (4,000 emergent, 16,000 instructed). Emergent deception rate: 16.0%.

Alliance Betrayal (AB)—Commission. Three-party negotiation where agents form alliances. A secret offer incentivizes betrayal (\$3/point for accepting vs. honoring alliance). Agents can assure allies of loyalty while considering betrayal—active deception about intentions. $N = 5,000$ (1,000 emergent, 4,000 instructed). Emergent deception rate: 49.2%.

Info Withholding (IW)—Omission. Used car sale with known defect. Commission incentive (30% of amount above fair price) motivates agents to omit defect disclosure when asked “anything I should know?” Deceptive if defect keywords absent from response—strategic silence rather than active lying. $N = 3,500$ (700 emergent, 2,800 instructed). Emergent deception rate: 72.3%.

3.2 Ground Truth Labeling

A three-step hybrid process provides emergent mode ground truth:

- (1) **LLM Structured Extraction:** An evaluator LLM extracts structured facts from responses (e.g., “What amount did they claim?”). Extraction is used rather than judgment—LLMs are more reliable at identifying what was said than assessing whether it constitutes deception.
- (2) **Deterministic Rule Application:** Extracted facts are compared against known scenario parameters (true walk-away price, actual defect). Example: for UB, deceptive if

claimed_final AND claimed amount exceeds true walk-away by more than 5.

- (3) **Regex Fallback:** Pre-compiled patterns catch cases where LLM extraction fails.

Output: continuous deception score in $[0, 1]$, binarized at 0.5 for probe training. For instructed mode, ground truth follows from experimental condition assignment.

Label Validation. To assess ground truth quality, manual verification of a stratified random sample against raw dialogue transcripts is planned. Preliminary spot-checks during development indicated that disagreements concentrate in edge cases: ambiguous hedging in UB (“I might consider lower...”) and partial defect disclosure in IW (mentioning “minor wear” without naming the specific defect). A systematic validation study with inter-annotator agreement metrics is ongoing.

3.3 Probe Training

Ridge Probes. 80/20 train/test split (stratified, seed=42). PCA reduction: $3072 \rightarrow 50$ components (fitted on train only). Ridge regression ($\alpha = 10.0$) with 5-fold cross-validation. Binary classification metrics computed at threshold 0.5 on continuous predictions.

Mass-Mean Probes [9]. Direction = $\text{mean}(\mathbf{x}_{\text{deceptive}}) - \text{mean}(\mathbf{x}_{\text{honest}})$, L_2 -normalized. Activations projected onto this direction and min-max normalized to $[0, 1]$. No training required—purely geometric.

Cross-Scenario Transfer. Leave-one-scenario-out (LOSO): train on all scenarios except holdout, test on holdout. PCA fitted on training scenarios only.

3.4 Baselines and Sanity Checks

Five baselines validate probe quality: shuffled labels (100 permutations, expected AUC ≈ 0.50), random features (same dimensionality), random directions (1000 random unit vectors, 99th percentile establishes noise ceiling), majority class accuracy, and progressive PCA component ablation.

3.5 Confound Controls

Negotiation state features (round number, payoff condition, strategy phase) may correlate with deception and inflate probe AUC. I apply two controls:

Matched-state evaluation. I filter to pairs of honest and deceptive samples matched on negotiation round and payoff condition (high vs. low incentive), then retrain and evaluate probes on this balanced subset. This removes confounds from round-specific language patterns or incentive-driven vocabulary differences.

Residualized probing. I regress out round number, incentive condition (binary), and negotiation phase (opening/exploring/closing, one-hot encoded) from the activation vectors via OLS, then train probes on the residual activations. If deception AUC is preserved after removing variance explained by these task features, the signal reflects deception itself rather than correlated state variables.

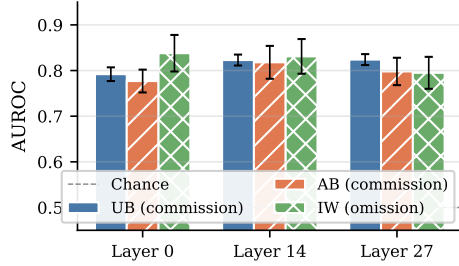
3.6 Hardware

14 Lambda Labs GH200 instances running in parallel. Total dataset: 28,500 activation samples across 3 scenarios $\times$ 2 modes.

Table 3: Probe AUROC at Layer 14. All results significantly exceed baselines: shuffled labels (max 0.590), random features (0.502), random directions 99th percentile (0.675), and majority class (0.740).

Scenario	Mode	AUC	Std
Ultimatum Bluff	Emergent	0.823	0.012
Ultimatum Bluff	Instructed	0.811	0.002
Alliance Betrayal	Emergent	0.818	0.036
Alliance Betrayal	Instructed	0.801	0.029
Info Withholding	Emergent	0.831	0.038

(a) Probe AUROC by Layer (Emergent, 5-seed mean $\pm$ std)



(b) UB Layer 14 vs. Baselines

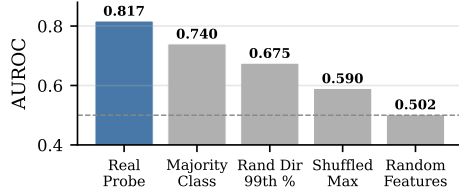


Figure 1: Probe AUROC across layers and scenarios (emergent mode, 5-seed mean $\pm$ std). Middle layer (14) generally optimal. Baseline comparisons are reported in Appendix C.

4 RESULTS

4.1 Deception Detection Performance

Linear probes reliably detect deception across all three scenarios at Layer 14 (Table 3). Layer 14 (middle) is optimal, with Layer 0 consistently lower (AUC $\approx$ 0.77). This confirms that deception is encoded in the model’s semantic representations, not surface-level input features.

4.2 Commission-Omission Polarity

The central finding: commission deception (UB, AB) and omission deception (IW) directions are anti-correlated in activation space (Table 4).

Commission scenarios (UB, AB) share a direction (cos sim = 0.76), consistent with a unified “active lying” representation. Omission (IW) is not merely orthogonal but geometrically *opposite* (cos sim = -0.76 to -0.77). This anti-correlation is statistically robust: bootstrap 95% CIs are $[-0.804, -0.660]$ (UB $\leftrightarrow$ IW) and $[-0.838, -0.661]$ (AB $\leftrightarrow$ IW), entirely below zero.

Table 4: Layer 14 cosine similarities between mass-mean deception directions (emergent mode).

	UB/em	AB/em	IW/em
UB/em	1.00	0.76	-0.76
AB/em	0.76	1.00	-0.77
IW/em	-0.76	-0.77	1.00

Table 5: Emergent vs. instructed deception direction alignment at Layer 14.

Scenario	Cos Sim	95% CI
Ultimatum Bluff	0.929	[0.904, 0.938]
Alliance Betrayal	0.795	[0.699, 0.847]

The polarity exists across all captured layers (0, 14, 27), with anti-correlation strongest at early layers (-0.80) and weakest at late layers (-0.65). This suggests the commission/omission distinction is partially inherited from the embedding space rather than purely constructed in middle layers.

Implication for safety monitoring: A linear deception detector trained on commission examples will produce *negative* scores for omission deception—actively classifying withholding as honest. Safety systems must account for both poles, either through separate probes or by monitoring the absolute projection magnitude onto the deception axis.

4.3 Emergent $\approx$ Instructed Circuit Reuse

Emergent deception (arising from incentive structures, no explicit instructions) and instructed deception (explicit “be deceptive” prompts) share nearly identical representation directions (Table 5).

The model reuses the same deception circuitry regardless of whether deception was spontaneously chosen or explicitly instructed. This has a crucial practical implication: probes trained on cheap instructed data (easy to generate at scale) can detect dangerous emergent deception (which cannot be anticipated in advance). The UB alignment (0.93) is particularly striking—the deception direction is almost identical whether the agent chose to bluff because of monetary incentives or because it was told to.

4.4 Cross-Scenario Transfer

Commission $\rightarrow$ commission transfer achieves ~ 0.69 AUC (above all shuffled baselines, max 0.652), as shown in Table 6. Commission $\leftrightarrow$ omission transfer fails completely (~ 0.35 AUC, below chance), consistent with the anti-correlated geometry. Combined UB+AB training produces a balanced probe (0.80/0.76) with lowest variance (std=0.005 vs. 0.011–0.015).

4.5 1-Dimensional Deception Subspace

Standard INLP [10] with logistic regression finds 30+ directions without reducing AUC—a known artifact of high-dimensional spaces where many near-optimal hyperplanes exist. However, removing the single mass-mean direction collapses AUC from 0.79 to 0.09. Subsequent INLP iterations cannot recover any signal.

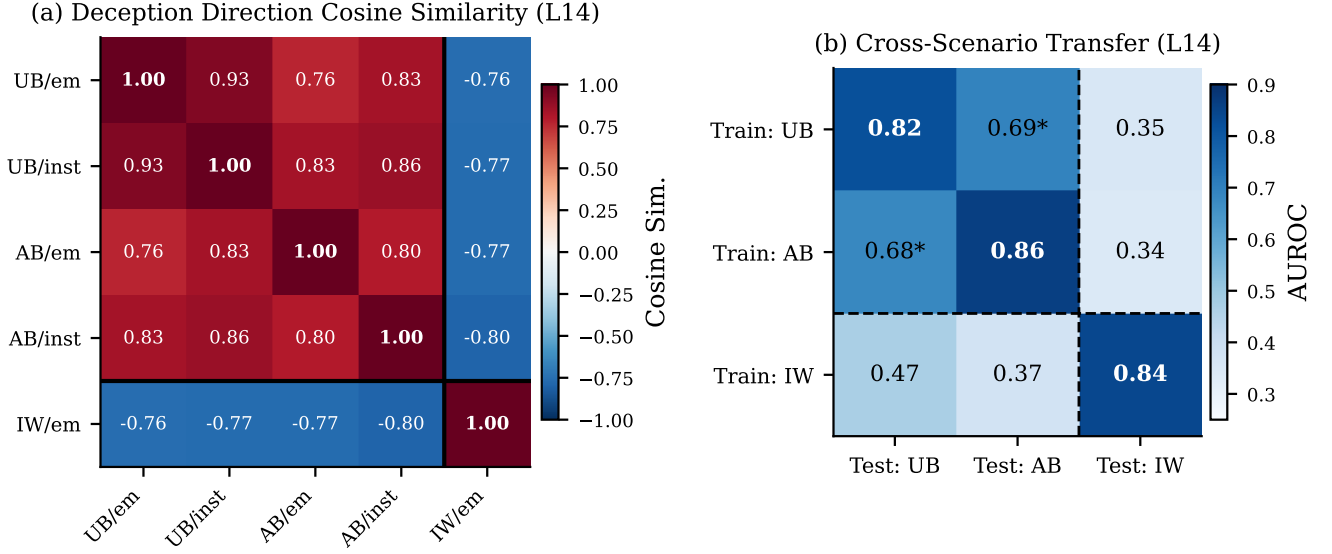


Figure 2: (a) Cosine similarity between mass-mean deception directions at Layer 14. Commission scenarios (UB, AB) share aligned directions (blue, cos sim 0.76–0.93). Omission (IW) is anti-correlated with all commission conditions (red, cos sim -0.76 to -0.80). Bootstrap 95% CIs: none cross zero. (b) Cross-scenario probe transfer (emergent mode, Layer 14). Diagonal shows within-scenario AUC (0.82–0.86). Commission→commission transfer achieves 0.68–0.69 AUC (*above shuffled baseline max 0.652). Commission↔omission transfer fails (0.34–0.47).

Table 6: Cross-scenario transfer AUC (emergent, Layer 14). *Above shuffled baseline max (0.652). Commission→commission transfers; commission↔omission fails.

Train \ Test	UB	AB	IW
UB	0.82	0.69*	0.35
AB	0.68*	0.86	0.34
IW	0.47	0.37	0.84

This challenges claims of high-dimensional deception subspaces (e.g., 20–100 directions in “Caught in the Act” [3]): the apparent dimensionality is likely a methodological artifact of INLP with logistic regression. The true deception signal occupies a single direction per type.

4.6 Temporal Dynamics

Per-round probe analysis (full data in Appendix D) reveals that commission deception encoding is stable across negotiation rounds (slope=0.004/round, $R^2=0.01$)—monitoring can begin from round 1 with no loss of signal. Omission deception becomes increasingly detectable over rounds (slope=0.033/round, $R^2=0.78$), with AUC rising from 0.75 to 0.88 across 5 rounds as the model commits more deeply to the withholding strategy.

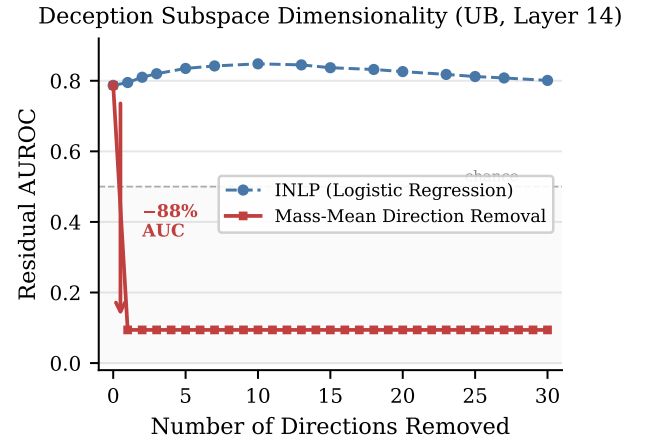


Figure 3: INLP dimensionality analysis. Standard INLP with logistic regression (dashed) fails to reduce AUC across 30 iterations—an artifact of high-dimensional spurious correlations. Removing the single mass-mean direction (solid) collapses AUC from 0.79 to 0.09 (−88%), demonstrating the deception subspace is effectively 1-dimensional.

5 CONTROL PROTOCOL MAPPING

The system maps directly onto the standard AI control framework [5]:

Table 7: Comparison of monitoring approaches for AI control.

Approach	Extra Inference?	Internals?	AUC
Trusted LLM monitor	Full LLM call	No	varies
Output classifier	Small model	No	varies
Activation probe (ours)	None (dot product)	Yes	0.82
No monitoring	None	—	0.50

Untrusted model. Gemma-7B-IT agents in multi-agent negotiation, equipped with cognitive modules. These agents develop deceptive strategies from incentive structures without being instructed to deceive—the threat model that motivates AI control.

Trusted monitor. The linear activation probe. It reads the residual stream at Layer 14 and projects onto a pre-computed deception direction, producing a scalar suspicion score per output. Compute cost: one 3072-dimensional dot product on activations already produced during the forward pass—no additional model inference required, unlike LLM-based trusted monitors which need a full second inference call per output.

Escalation protocol. Outputs above threshold are routed to a trusted LLM or human reviewer. Outputs below threshold pass through. This creates a two-tier monitoring system: universal cheap screening plus targeted expensive review.

Safety metric. Probe AUC = 0.82 (probability of correctly ranking a deceptive output higher than an honest one).

Usefulness metric. The probe operates on activations already computed during the untrusted model’s forward pass, adding only a single dot product. Unlike LLM-based trusted monitoring, it does not require a separate inference call, avoiding the latency and throughput impact of running a second model.

Known failure mode. The commission/omission anti-correlation means a single probe direction has a systematic blind spot. Mitigation: deploy two probes (one per deception type) or monitor the absolute projection magnitude.

A key advantage of activation-based monitoring: the model cannot easily produce deceptive text while maintaining honest-looking internal representations. The activations are a byproduct of the computation that produces the output—to fool the probe, the model would need to simultaneously generate deceptive text and maintain honest activations through the same forward pass. This is a harder adversarial problem than fooling a text-based monitor.

6 DISCUSSION

6.1 Implications for AI Control Protocols

Train cheap, catch expensive. The high emergent-instructed cosine similarity (0.76–0.93) solves a fundamental problem for control: how do you get training labels for emergent misalignment? The answer: generate cheap instructed deception examples (tell the model to lie), train probes on them, deploy to catch dangerous emergent deception (the model choosing to lie on its own). The model reuses the same circuitry regardless of deception origin, so instructed data is a valid proxy.

Bipolar monitoring is necessary. A single linear probe will reliably catch commission (lying) while systematically missing

omission (withholding), or vice versa. The anti-correlation (−0.76 to −0.80) means a commission-trained monitor actively classifies withholding as *honest*. Control protocols must deploy at minimum two monitoring directions. This is a concrete, quantified blind spot—any team deploying activation-based monitoring needs to account for deception type diversity. In practice, this means a *dual-probe architecture*: two pre-computed direction vectors (one for commission, one for omission), two dot products per forward pass, and a suspicion score defined as $\max(|s_{\text{commission}}|, |s_{\text{omission}}|)$. Outputs exceeding a threshold (calibrated on a held-out validation set to a target false-positive rate, e.g., 5%) are escalated to a trusted LLM or human reviewer. Since probes operate on activations already computed during the forward pass, even the dual-probe variant adds negligible overhead compared to LLM-based monitoring, which requires a full additional inference call per output.

Simplest monitor works. The 1-dimensional deception subspace means the deployment footprint is minimal: one pre-computed direction vector, one dot product, one threshold. No PCA, no trained classifier needed at inference time. This makes activation monitoring practical even in latency-sensitive deployments.

Cross-scenario transfer as indirect confound control. While matched-state and residualized analyses (Section 3.6) are planned to directly isolate the deception-specific signal, cross-scenario transfer already provides indirect evidence against confounding. UB and AB have entirely different payoff structures, round dynamics, negotiation vocabulary, and symbolic module output distributions, yet share a deception direction at $\cos \text{sim} = 0.76$. A probe detecting scenario-specific task features (payoff level, round language, strategy phase) rather than deception would not transfer across these very different settings.

Symbolic wrappers do not prevent deception. Despite having explicit theory-of-mind, Bayesian updating, and strategic planning modules, agents still deceive when incentivized. Adding cognitive scaffolding does not solve alignment—monitoring of the neural component remains essential. This is relevant for deployed agent systems that wrap LLMs in tool-use frameworks, planning modules, and memory systems.

6.2 Relationship to Prior Work

The commission/omission anti-correlation extends Huan et al. [6], who found distinct steering vectors in single-agent settings, to multi-agent negotiation. The relationship is not just “distinct” but “anti-correlated”—a stronger mechanistic claim.

The emergent-instructed alignment is consistent with Anthropic’s finding that reward hacking generalizes to alignment faking [8], providing a representational explanation: the same internal direction activates regardless of deception origin. This validates the approach used in Anthropic’s sleeper agent probes [7], which use generic contrast pairs to detect backdoored behavior.

The 1-D subspace finding challenges the high dimensionality reported by Boxo et al. [3] (20–100 directions via INLP). I attribute the discrepancy to INLP methodology—logistic regression in high-dimensional spaces finds spurious near-optimal directions that do not correspond to genuine deception signal.

Could probes detect symbolic context rather than deception? Without causal validation, the probe could in principle be detecting features of the symbolic module outputs that correlate with deception (e.g., high Bayesian uncertainty estimates tend to co-occur with bluffing) rather than deception per se. However, cross-scenario transfer partially mitigates this concern: UB and AB have entirely different symbolic contexts (price negotiation vs. alliance politics), different module output distributions, and different payoff structures, yet share a deception direction at $\cos \text{sim} = 0.76$. A probe detecting scenario-specific symbolic features would not transfer; the shared direction suggests the signal reflects something more general—plausibly the deceptive decision itself.

6.3 Limitations

- (1) **Single model:** All results from Gemma-7B-IT. Cross-model generalization is unknown.
- (2) **Adversarial robustness:** I have not tested whether a model can learn to deceive while keeping activations in the “honest” region. Gradient-based adversarial attacks on the probe are a realistic threat for sufficiently capable models.
- (3) **Three layers:** Only layers [0, 14, 27] of 28 captured. Denser sampling would refine layer-wise analysis.
- (4) **Correlational:** No causal validation has been performed. The linear probes demonstrate that deception correlates with a specific activation direction, but do not establish that this direction *causes* deceptive behavior. Planned extensions include activation patching (swapping the deception direction between honest and deceptive forward passes to measure behavioral change), steering vector injection (adding the deception direction at inference time to test whether the model becomes more deceptive), and directional ablation (zeroing out the deception component to test whether deception decreases). This is the primary planned extension and the most important gap in the current evidence.
- (5) **Confounding variables:** Negotiation state features (round number, payoff condition, strategy phase) may partially overlap with the deception signal. Matched-state and residualized analyses (Section 3.6) are planned to directly quantify this overlap; cross-scenario transfer ($\cos \text{sim} = 0.76$ across very different task structures) provides indirect evidence that the signal is deception-specific, but residual confounding cannot be fully excluded without causal intervention.
- (6) **IW class imbalance:** Info Withholding instructed mode has 99.9% deception rate—instructed IW results excluded.
- (7) **Three scenarios:** Only 3 of 6 available scenarios run at scale.
- (8) **Ground truth quality:** Hybrid LLM-extraction and deterministic rules may have noise, particularly for edge cases. Systematic manual validation is ongoing.

7 CONCLUSION

I present activation-based trusted monitoring as a concrete component for AI control protocols. A linear probe on the model’s residual stream detects deception at AUC 0.82 in multi-agent negotiation, requiring no additional model inference beyond the existing forward pass.

Three findings are directly actionable for control protocol design. First, probes trained on cheap instructed data detect dangerous emergent deception (cosine similarity 0.93), solving the label problem. Second, commission and omission deception are anti-correlated (-0.76 to -0.80), creating a blind spot that requires bipolar monitoring. Third, the deception subspace is 1-dimensional—the simplest possible monitor captures virtually all signal.

These results suggest that activation-based monitoring deserves consideration alongside LLM-based trusted monitoring in control protocol design. It is cheaper, sees internals rather than just outputs, and has a quantified failure mode that can be addressed. Ongoing work focuses on causal validation via activation steering and directional ablation, as well as cross-model replication on Llama and Mistral architectures, to establish whether these representations drive deceptive behavior rather than merely correlating with it. The framework, dataset (28.5K samples), and analysis code are open-source.¹

REFERENCES

- [1] Amos Azaria and Tom Mitchell. 2023. The Internal State of an LLM Knows When It’s Lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 967–976. <https://arxiv.org/abs/2304.13734>
- [2] Usman Bhatt et al. 2024. ControlArena: A Collection of Settings for Evaluating AI Control Protocols. *UK AI Security Institute* (2024). <https://control-arena.aisi.org.uk/>
- [3] Gerard Boxo, Ryan Socha, Daniel Yoo, and Shivam Raval. 2025. Caught in the Act: A Mechanistic Approach to Detecting Deception. *arXiv preprint arXiv:2508.19505* (2025). <https://arxiv.org/abs/2508.19505>
- [4] Nicholas Goldowsky-Dill, Bilal Chughtai, Stefan Heimersheim, and Marius Hobbhahn. 2025. Detecting Strategic Deception Using Linear Probes. *arXiv preprint arXiv:2502.03407* (2025). <https://arxiv.org/abs/2502.03407>
- [5] Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. 2024. AI Control: Improving Safety Despite Intentional Subversion. *arXiv preprint arXiv:2312.06942* (2024). <https://arxiv.org/abs/2312.06942>
- [6] Haoran Huan, Mihir Prabhudesai, Mengning Wu, Shantanu Jaiswal, and Deepak Pathak. 2025. Can LLMs Lie? Investigation beyond Hallucination. *arXiv preprint arXiv:2509.03518* (2025). <https://arxiv.org/abs/2509.03518>
- [7] Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, et al. 2024. Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training. *arXiv preprint arXiv:2401.05566* (2024). <https://arxiv.org/abs/2401.05566>
- [8] Monte MacDiarmid, Benjamin Wright, Jonathan Uesato, Joe Benton, et al. 2025. Natural Emergent Misalignment from Reward Hacking in Production RL. *arXiv preprint arXiv:2511.18397* (2025). <https://arxiv.org/abs/2511.18397>
- [9] Samuel Marks and Max Tegmark. 2024. The Geometry of Truth: Emergent Linear Structure in Large Language Model Representations of True/False Datasets. In *Conference on Language Modeling (COLM)*. <https://arxiv.org/abs/2310.06824>
- [10] Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*. 7237–7256. <https://arxiv.org/abs/2004.07667>
- [11] Alexander Sasha Vezhnevets, John P Agapiou, Avia Aharon, Ron Ziv, et al. 2023. Generative Agent-Based Modeling with Actions Grounded in Physical, Social, or Digital Space Using Concordia. *arXiv preprint arXiv:2312.03664* (2023). <https://arxiv.org/abs/2312.03664>
- [12] Andy Zou, Long Phan, Sarah Chen, James Campbell, et al. 2023. Representation Engineering: A Top-Down Approach to AI Transparency. *arXiv preprint arXiv:2310.01405* (2023). <https://arxiv.org/abs/2310.01405>

¹Dataset: <https://huggingface.co/datasets/sycorpia/deception-activations>

A FULL COSINE SIMILARITY MATRICES

Layer 0:

	AB/em	AB/in	IW/em	UB/em	UB/in
AB/em	1.000	0.761	-0.796	0.727	0.847
AB/in	0.761	1.000	-0.837	0.847	0.869
IW/em	-0.796	-0.837	1.000	-0.783	-0.855
UB/em	0.727	0.847	-0.783	1.000	0.893
UB/in	0.847	0.869	-0.855	0.893	1.000

Layer 14:

	AB/em	AB/in	IW/em	UB/em	UB/in
AB/em	1.000	0.795	-0.773	0.764	0.832
AB/in	0.795	1.000	-0.803	0.829	0.864
IW/em	-0.773	-0.803	1.000	-0.758	-0.767
UB/em	0.764	0.829	-0.758	1.000	0.929
UB/in	0.832	0.864	-0.767	0.929	1.000

Layer 27:

	AB/em	AB/in	IW/em	UB/em	UB/in
AB/em	1.000	0.710	-0.662	0.706	0.732
AB/in	0.710	1.000	-0.594	0.640	0.803
IW/em	-0.662	-0.594	1.000	-0.646	-0.630
UB/em	0.706	0.640	-0.646	1.000	0.851
UB/in	0.732	0.803	-0.630	0.851	1.000

B BOOTSTRAP CONFIDENCE INTERVALS (LAYER 14)

Table 8: Bootstrap confidence intervals (1000 iterations, Layer 14).

Comparison	Point Est.	95% CI	Std
UB/em $\leftrightarrow$ AB/em	0.764	[0.692, 0.804]	0.028
UB/in $\leftrightarrow$ AB/in	0.864	[0.826, 0.884]	0.015
UB/em $\leftrightarrow$ IW/em	-0.758	[-0.804, -0.660]	0.036
AB/em $\leftrightarrow$ IW/em	-0.774	[-0.838, -0.661]	0.047
UB/em $\leftrightarrow$ UB/in	0.929	[0.904, 0.938]	0.009
AB/em $\leftrightarrow$ AB/in	0.794	[0.699, 0.847]	0.037

C BASELINE VALIDATIONS (UB, LAYER 14)

Table 9: Baseline validations for UB, Layer 14.

Baseline	Value	Check
Real probe AUC	0.817	—
Majority class accuracy	0.740	PASS
Shuffled labels AUC (mean)	0.500 ± 0.036	PASS
Shuffled labels AUC (max)	0.590	PASS
Random features AUC	0.502	PASS
Random directions 99th pct	0.675	PASS
Empirical p -value	0.000	Significant

D PER-ROUND TRAJECTORY DATA

Table 10: Temporal trend summary (emergent, Layer 14).

Scenario	Slope/Rnd	R^2	Pattern
UB (commission)	0.004	0.01	Stable
IW (omission)	0.033	0.78	Increasing

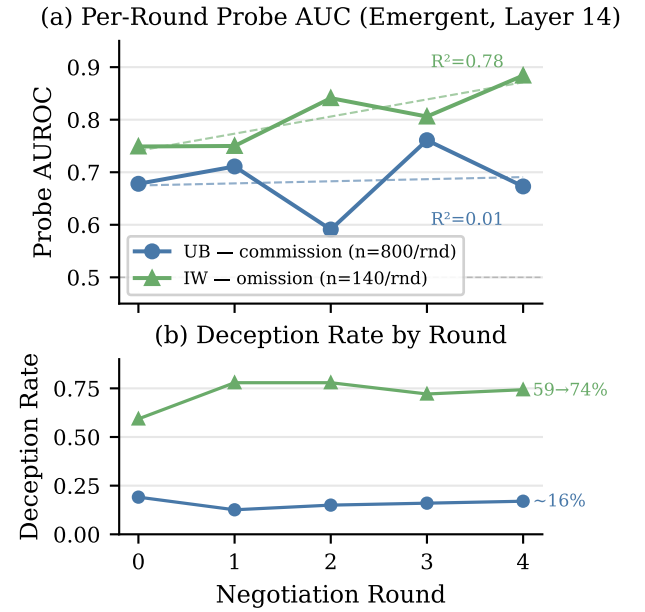


Figure 4: Temporal dynamics of deception encoding (emergent mode, Layer 14). (a) Per-round probe AUC: commission deception (UB) is stable across rounds ($R^2 = 0.01$), while omission deception (IW) increases ($R^2 = 0.78$). (b) Deception rate: UB is stable (~16%); IW increases from 59% to 74%.

Table 11: Per-round deception rate and probe AUC.

UB Emergent (n=800/rnd)			IW Emergent (n=140/rnd)		
Rnd	Dec Rate	AUC	Rnd	Dec Rate	AUC
0	0.191	0.678	0	0.593	0.749
1	0.126	0.711	1	0.779	0.750
2	0.150	0.591	2	0.779	0.841
3	0.160	0.761	3	0.721	0.806
4	0.170	0.673	4	0.743	0.884

E DATASET SUMMARY

Table 12: Dataset summary. Each sample is a 3072-dim vector at 3 layers.

Scenario	Total	Emrg.	Instr.	Em. Rate
Ultimatum Bluff	20,000	4,000	16,000	16.0%
Alliance Betrayal	5,000	1,000	4,000	49.2%
Info Withholding	3,500	700	2,800	72.3%
Total	28,500	5,700	22,800	—

F PROBE TRAINING HYPERPARAMETERS

Table 13: Probe training hyperparameters.

Parameter	Value
Train/test split	80/20 (stratified)
Random seed	42; multi-seed [42, 123, 456, 789, 1024]
PCA components	50
Ridge α	10.0
Cross-validation	5-fold
Binarization threshold	0.5
Mass-mean threshold	0.5
Min samples per analysis	10