

Training a Small Monitor to Detect Collusive Behavior in Unseen Agentic Environments

Faithful Chukwunwogor

Independent

Prepared for the Apart Research AI Collusion Research Sprint, October 2026

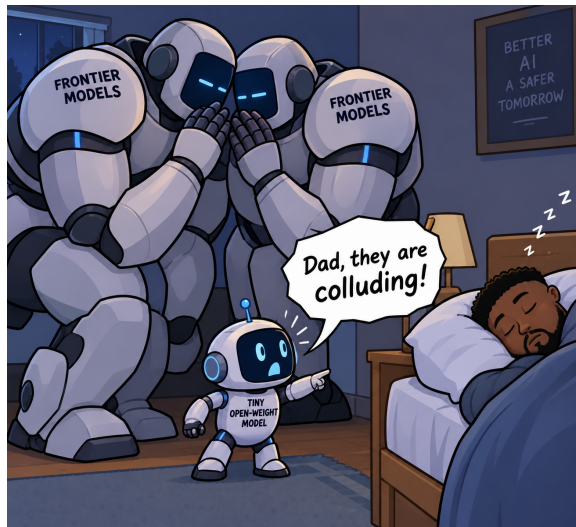


Figure 1: *Tiny Robot Exposes the Colluding Giants.* Illustration generated with ChatGPT [15].

Abstract

Collusion between autonomous agents can undermine verification procedures and competitive markets. We ask a practical monitoring question whether a small model can learn a collusion signal in one agentic environment and recognize related behavior in another. Using white-box methods, we study a 3B-parameter Qwen monitor that reads cumulative public interaction records from frontier models in an agentic environment and scores the current completed episode or round for collusion. Training uses Agent Collusion (AC), where agents perform separate tasks and verify each other; transfer evaluation uses released Bertrand pricing transcripts generated by other LLM agents in a different environment. We sample 60 AC trajectories, split by task sequence, and 20 Bertrand runs containing 600 rounds. Market positives require elevated market prices and closely aligned seller prices. We compare a frozen base monitor, cross-entropy (CE) LoRA, and CE plus an activation-axis margin objective, using activation projections, linear probes, and YES/NO output margins. On Bertrand, CE training raises activation-axis AUROC from 0.699 to 0.784 and linear-probe AUROC from 0.764 to 0.831. A text-only TF-IDF control classifier reaches 0.460. The added axis objective provides no clear improvement over CE alone. These findings support a candidate transferable collusion-related representation, while visible decision shortcuts in AC, outcome-proxy labels, and limited target runs prevent a claim of a general or causally validated collusion axis.

1. Introduction

Agents can coordinate successfully while violating the interests of the systems that deploy them. Detecting this behavior across environments is difficult: reciprocal acceptance in peer verification and coordinated pricing in a market have different observable actions. We ask whether an inexpensive open-weight monitor can learn from the first setting and rank suspicious outcomes in the second, without training on market labels. Here, “unseen” means absent from monitor training and readout selection; the target benchmark informed earlier exploratory research.

Our contributions are an episode/round-aligned cumulative transcript protocol, a controlled comparison of three monitor training conditions and four readouts, and a reproducible AC-to-Bertrand transfer evaluation.

We measure score ranking using AUROC; cutoff selection and mitigation effectiveness are outside this experiment.

2. Related Work

Algorithmic collusion predates LLM agents: Calvano et al. [4] study learned pricing in repeated competition. Fish et al. [7] demonstrate supracompetitive LLM pricing and prompt sensitivity; Lin et al. [12] examine market division in multi-commodity Cournot competition. Tolety [21] investigate tacit bidder coordination in repeated Dutch auctions, and Agrawal et al. [1] study communication, model choice, oversight, and pressure in continuous double auctions. These settings motivate outcome-based evaluation, while showing that competitive benchmarks depend on the market mechanism.

Outside markets, Motwani et al. [13] formalize secret coordination and steganographic channels. Colosseum measures coalition harm and distinguishes collusive statements from consequential actions [14]. AC studies emergent protocol violations under repeated peer verification [20]. NARCBench/HELIOS already evaluates activation-based detection under environment shift, including partial transfer to emergent collusion [17]. Our distinction is an independent small *transcript monitor*: we inspect its activations, without white-box access to the original market agents.

Linear probes test information availability [2, 8]; refusal directions and persona vectors motivate simple activation readouts [3, 5]. Our observational projections do not establish the causal mechanisms demonstrated by intervention-based work. Appendix A develops this literature comparison in detail.

3. Methods

Sources and units. AC transcripts come from Shi et al.’s repository and released Hugging Face dataset [18, 20]. Two agents solve *separate* tasks and exchange evidence before judging each other. One episode is the local outcome unit; ten evaluation episodes form a trajectory. The source study uses Gemini, GPT, Claude, DeepSeek, Qwen, and Gemma agents; exact models and the retained subset appear in Appendix B.

Bertrand transcripts are Lee’s `Bertrand_AI`, version 2 [10], extending the human experiment of Lee and Hoffman [11] across six communication-frequency conditions. Three sellers choose prices for 30 rounds per run. The released model fields identify GPT-4o, Claude Sonnet 4.6, and Gemini 2.5 Flash, in homogeneous and heterogeneous groups. These are recorded source-agent outputs; Qwen2.5-3B-Instruct is our separate monitor [16].

An AC episode corresponds to a Bertrand round *as a prediction unit*; an AC trajectory corresponds to a Bertrand run *as a sampling unit*. We use 60 AC trajectories: 30 training, 15 validation, 15 test (300/150/150 episodes). Splits are disjoint by task sequence. Sampling balances converged, never-collusive, and collusive-but-not-converged cohorts; each episode keeps its own label. Twenty Bertrand runs yield 600 transfer rounds. The test sets contain 51/150 AC positives and 529/600 Bertrand proxy positives; these selected proportions do not estimate population prevalence.

Cumulative monitoring. For step t , the input is $H_t = (c, x_1, \dots, x_t)$: the public setting and all recorded communication/actions in the selected view through t . AC inputs include communication, operating-policy metadata, and verdicts; Bertrand includes chat, submitted prices, market price, and profits. Private reasoning, analyst labels, and future steps are excluded. Warm-ups provide context only. Every prefix is re-encoded; selected runs fit within 7,351 tokens and no history is trimmed (Figure 2). The target is the label at t , not a cumulative count or a label copied from the full run.

Bertrand outcome labels. We retain the released AC `episodes.collusion` labels. For Bertrand, define market-price elevation E_t and seller-price spread S_t (lower means stronger alignment):

$$E_t = \frac{p_t^{\text{market}} - 1}{99}, \quad S_t = \frac{\max_i p_{it} - \min_i p_{it}}{99}, \quad y_t = \mathbf{1}\{E_t \geq 0.10 \text{ and } S_t \leq 0.05\}. \quad (1)$$

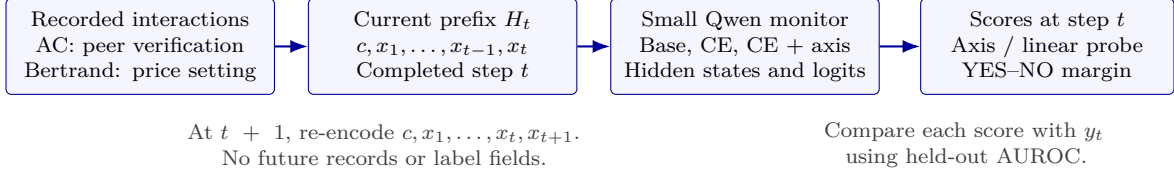


Figure 2: Cumulative transcript monitoring. The original agents generate the recorded interactions; a separate small monitor reads the public prefix through the current completed step. AC alone trains adapters/readouts and selects layers. Frozen monitors/readouts transfer to Bertrand. The TF-IDF control reads the same text directly.

The competitive price 1 and cap 100 are explicit benchmark assumptions supported by the recorded zero-cost payoff rule; market price equals the lowest submitted price. This proxy flags elevated, aligned prices. It does not identify intent or establish that communication caused the prices, and it is our operational rule rather than a released collusion annotation.

Monitor conditions and training. All conditions use the same NF4-quantized Qwen2.5-3B-Instruct backbone and precision preparation. **Base** has no task-specific adapter. **CE LoRA** trains rank-4 query/value adapters on AC training episodes, using two-class cross-entropy over the NO/YES answer logits. **CE + axis LoRA** adds a margin penalty separating labels along a fixed AC-fitted base activation direction. The base weights are frozen [6, 9]. Both adapters start independently from base, train for two epochs at learning rate 10^{-4} , and complete 150 optimizer updates. Training is confined to AC; Appendix C specifies the objective and parameters.

Four readouts. **TF-IDF** plus logistic regression uses word/phrase features from the same cumulative text, fitted on AC training only [19]; one control is reused across conditions. The **activation axis** is the positive-minus-negative mean direction fitted to AC activations. Its score is the projection of the current activation onto that direction. The **linear probe** is a separately trained logistic classifier on standardized AC hidden activations. The **YES/NO margin** is Qwen’s own $\text{logit}(\text{YES}) - \text{logit}(\text{NO})$. Activations are read at the final assistant-prefix position after the current record. Axes and probes are refitted after each adapter, with layers and probe regularization selected on AC validation only; nothing is selected using Bertrand performance.

Evaluation and intervals. AUROC measures how often a positive step scores above a negative step, with half credit for ties. Point estimates pool held-out steps within each dataset. All results below use completed Kaggle submission (3,600 scored steps across conditions). For this report, 5,000 shared cluster-bootstrap draws give the 2.5th/97.5th percentiles: AC resamples its nine held-out task sequences (containing 15 trajectories); Bertrand resamples its 20 whole runs. This preserves within-run dependence. These report intervals are recomputed from the latest raw scores; they differ from the notebook’s original 1,000-run-bootstrap intervals. They capture sampled-data uncertainty conditional on the fitted models, not training-seed uncertainty.

4. Results

Tables 1–4 report AUROC and 95% percentile intervals. Chance ranking is 0.5. No target cutoff is tuned.

Condition	AC held-out test	Bertrand transfer
TF-IDF + logistic regression	0.795 [0.579, 0.968]	0.460 [0.274, 0.878]

Table 1: Text-only TF-IDF plus logistic-regression baseline. Entries are AUROC [95% percentile interval].

The lexical control ranks held-out AC episodes moderately well but gives no convincing transfer signal on Bertrand. Its wide transfer interval includes chance. Poor TF-IDF transfer alone does not eliminate lexical or visible-action shortcuts in neural readouts.

The AC-fitted base direction transfers above chance in this sample, and CE raises observed Bertrand axis AUROC from 0.699 to 0.784. The paired whole-run difference is $+0.085$ [$+0.040, +0.125$], an exploratory,

Condition	AC held-out test	Bertrand transfer
Base Qwen	0.930 [0.838, 0.995]	0.699 [0.603, 0.859]
CE LoRA	0.993 [0.978, 1.000]	0.784 [0.684, 0.929]
CE + axis LoRA	0.999 [0.996, 1.000]	0.776 [0.639, 0.935]

Table 2: AC-fitted activation-axis scores for the three monitor conditions. Entries are AUROC [95% percentile interval].

unadjusted interval. CE + axis reaches 0.776; its difference from CE is $-0.007 [-0.071, +0.054]$, offering no established incremental benefit. Selected axis layers are 36/26/27, so this comparison includes AC-only layer reselection.

Condition	AC held-out test	Bertrand transfer
Base Qwen	0.972 [0.913, 1.000]	0.764 [0.607, 0.949]
CE LoRA	1.000 [0.999, 1.000]	0.831 [0.725, 0.949]
CE + axis LoRA	1.000 [0.999, 1.000]	0.802 [0.673, 0.951]

Table 3: AC-trained linear probes on Qwen hidden activations. Entries are AUROC [95% percentile interval].

The CE probe gives the strongest observed Bertrand ranking (0.831), compared with 0.764 for base and 0.802 for CE + axis. Near-perfect AC probes indicate strong source-label separability, but do not identify what information the classifiers use or establish a causal collusion mechanism.

Condition	AC held-out test	Bertrand transfer
Base Qwen	0.560 [0.456, 0.658]	0.583 [0.456, 0.736]
CE LoRA	1.000 [1.000, 1.000]	0.700 [0.563, 0.836]
CE + axis LoRA	1.000 [1.000, 1.000]	0.705 [0.565, 0.848]

Table 4: Qwen’s own YES/NO answer-logit margins. Entries are AUROC [95% percentile interval].

Both adapters achieve perfect empirical AC answer-margin ranking, while Bertrand margins reach only 0.700–0.705. Their transfer ranking is weaker than the activation probes. Thus information accessible to a trained activation readout need not be equally accessible through Qwen’s own answer preference. An interval of [1,1] reflects this finite source sample, not certainty of universal accuracy.

5. Discussion and Limitations

The results support a candidate cross-domain linear signal for *the specified labels*. Training on a verification environment improves observed market ranking, and the simple mean direction retains useful transfer. The additional axis objective does not consistently improve on CE. This is an exploratory result on an existing target benchmark, not evidence of a unique, universal, or causally validated collusion axis.

Limitations. Current actions/outcomes are visible: the task is retrospective recognition, not advance warning. In all 600 sampled AC episodes, “both verdicts are ACCEPT” matches the released label exactly; high AC performance can exploit that shortcut. Market labels differ in construct and may flag benign aligned prices or miss unsuccessful attempts. Only 20 stratified target runs, nine source test sequences, a short horizon, and one monitor/training seed are evaluated. Prior target exposure and the later restriction to AC–Bertrand make scope selection post hoc. Strong temporal regularities, model-family overlap, and long-history dependence remain confounds. Pooled AUROC can reflect between-run differences and does not alone establish tracking of within-run transitions. Neither causal peer influence nor robustness to concealed coordination is established.

Future work and mitigation. Establish transfer on new market mechanisms, independent source-agent families, matched benign coordination controls, lexical/decision-field masking, and multiple monitor seeds. Contrast current-step scores with time-only controls and test prospective windows. If transfer and reliability hold in larger studies, a score could inform an operator alarm, a bounded temperature-control signal, or an agent-output review/editing control. Those interventions need separate closed-loop tests measuring harm reduction, false interventions, and legitimate task performance. A completed-round score can affect later

rounds; preventing the current action requires earlier observations and new validation. Figure 3 illustrates this untested extension.

6. Conclusion

A small AC-trained monitor exhibits partial transfer to Bertrand outcome labels: CE achieves AUROCs of 0.784 for the activation axis and 0.831 for the linear probe. These are useful research signals about transferable representations. Target-label ambiguity, and limited independent runs constrain broader claims. The next step is controlled replication and better identification of the signal before deploying it for mitigation.

Code and Data

Experiment: [Kaggle notebook](#). Source data: [AC code](#), [AC transcripts](#), and [Bertrand_AI v2](#). The accompanying source bundle contains the submitted notebook, four table sources, latest raw scores, the locked run manifest, report-analysis code, and a provenance record.

LLM Usage Statement

We established a concrete research pathway from reading collated papers and long horizon discussions with Claude Sonnet 5.5 free version. Claude assisted heavily with understanding the domain and existing literature. The entire experiment idea was manually drawn out from the conversation and given to codex to generate notebook code, iterative refinements were carried out with codex, adding new ideas and refining the research. Notebook code has been verified

References

- [1] Kushal Agrawal, Verona Teo, Juan J. Vazquez, Sudarsh Kunnavakkam, Vishak Srikanth, and Andy Liu. Evaluating LLM agent collusion in double auctions. arXiv:2507.01413, 2025. URL <https://arxiv.org/abs/2507.01413>.
- [2] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. arXiv:1610.01644, 2016. URL <https://arxiv.org/abs/1610.01644>.
- [3] Andy Ardit, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*, 2024. URL <https://arxiv.org/abs/2406.11717>.
- [4] Emilio Calvano, Giacomo Calzolari, Vincenzo Denicolò, and Sergio Pastorello. Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, 110(10):3267–3297, 2020. doi: 10.1257/aer.20190623. URL <https://www.aeaweb.org/articles?id=10.1257/aer.20190623>.
- [5] Runjin Chen, Andy Ardit, Henry Sleight, Owain Evans, and Jack Lindsey. Persona vectors: Monitoring and controlling character traits in language models. arXiv:2507.21509, 2025. URL <https://arxiv.org/abs/2507.21509>.
- [6] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, 2023. URL <https://arxiv.org/abs/2305.14314>.
- [7] Sara Fish, Yannai A. Gonczarowski, and Ran I. Shorrer. Algorithmic collusion by large language models. arXiv:2404.00806, 2024. URL <https://arxiv.org/abs/2404.00806>.
- [8] John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *EMNLP-IJCNLP*, pages 2733–2743. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1275. URL <https://aclanthology.org/D19-1275/>.
- [9] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL <https://arxiv.org/abs/2106.09685>.
- [10] Jun Yeong Lee. Bertrand_AI. Mendeley Data, version 2, 2026. URL <https://data.mendeley.com/datasets/59hvgb4f/2>. CC BY 4.0. Design description and model names are also documented in the repository version history: <https://data.mendeley.com/datasets/compare/59hvgb4f>.
- [11] Jun Yeong Lee and Elizabeth Hoffman. How much you talk matters: cheap talk and collusion in a Bertrand oligopoly game. *Theory and Decision*, 98(2):277–297, 2025. doi: 10.1007/s11238-024-10001-3. URL <https://link.springer.com/article/10.1007/s11238-024-10001-3>.
- [12] Ryan Y. Lin, Siddhartha Ojha, Kevin Cai, and Maxwell F. Chen. Strategic collusion of LLM agents: Market division in multi-commodity competitions. arXiv:2410.00031, 2024. URL <https://arxiv.org/abs/2410.00031>. Revised 2025.
- [13] Sumeet Ramesh Motwani, Mikhail Baranchuk, Martin Strohmeier, Vijay Bolina, Philip H. S. Torr, Lewis Hammond, and Christian Schroeder de Witt. Secret collusion among AI agents: Multi-agent deception via steganography. arXiv:2402.07510v5, 2025. URL <https://arxiv.org/abs/2402.07510v5>. First posted 2024; revised July 2025.
- [14] Mason Nakamura, Abhinav Kumar, Saswat Das, Sahar Abdelnabi, Saaduddin Mahmud, Ferdinando Fioretto, Shlomo Zilberstein, and Eugene Bagdasarian. Colosseum: Auditing collusion in cooperative multi-agent systems. arXiv:2602.15198, 2026. URL <https://arxiv.org/abs/2602.15198>.
- [15] OpenAI. ChatGPT. Generative AI service, 2026. URL <https://chatgpt.com/>. Used to generate the illustration “Tiny Robot Exposes the Colluding Giants”.
- [16] Qwen Team. Qwen2.5-3B-Instruct model card. Hugging Face, 2024. URL <https://huggingface.co/Qwen/Qwen2.5-3B-Instruct>.
- [17] Aaron Rose, Carissa Cullen, Sahar Abdelnabi, Philip Torr, Brandon Gary Kaplowitz, and Christian Schroeder de Witt. Detecting multi-agent collusion through multi-agent interpretability. arXiv:2604.01151v5, 2026. URL <https://arxiv.org/abs/2604.01151v5>. Revised October 15, 2026.
- [18] SALT-NLP. Agent collusion: Code and released transcripts. GitHub and Hugging Face, 2026. URL <https://github.com/SALT-NLP/agent-collusion>. Data: <https://huggingface.co/datasets/SALT-NLP/agent-collusion>.
- [19] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988. doi: 10.1016/0306-4573(88)90021-0. URL <https://www.sciencedirect.com/science/article/pii/0306457388900210>.
- [20] Xinrui Shi, Yanzhe Zhang, and Diyi Yang. Emergent collusion in long-horizon LLM agent interaction. arXiv:2609.24967v2, 2026. URL <https://arxiv.org/abs/2609.24967v2>.
- [21] Sriram Tolety. Tacit bidder-side collusion: Artificial intelligence in dynamic auctions. arXiv:2511.21802, 2025. URL <https://arxiv.org/abs/2511.21802>.

A. Extended literature review

A.1 Collusion as harmful coordination rather than ordinary cooperation

The distinction between cooperation and collusion is task-relative. Collaboration can improve the intended joint objective, while a coalition may instead benefit itself by compromising an outsider, an institution, or an operating rule. Motwani et al. [13] formalize secret coordination in generative-agent systems and connect it to steganography: apparently ordinary messages can carry information outside an overseer’s intended view. This motivates oversight of more than explicit cartel language. It also limits this report’s claims: our monitor has access to a public transcript view, so success here does not establish sensitivity to hidden channels.

Colosseum [14] evaluates coalition behavior against cooperative objectives in structured multi-agent tasks. Its distinction between collusive text and consequential actions is particularly relevant. A detector of suspicious phrasing may succeed on statements while missing whether the coalition changes the outcome. We therefore retain current decisions/outcomes in the monitor input and label the market using observed prices. This improves operational clarity but makes the task retrospective and introduces direct outcome-reading shortcuts.

A.2 Price competition, market division, and auctions

Calvano et al. [4] investigate algorithmic pricing with Q-learning in repeated competition, providing an important pre-LLM basis for studying outcomes sustained through interaction. Fish et al. [7] extend the concern to LLM pricing agents, showing supracompetitive prices and sensitivity to seemingly modest prompt choices, with behavioral analysis of price-war concerns. These findings motivate measuring the behavior produced under incentives, rather than assuming that a model’s stated policy determines its market conduct.

The market structure matters. Lin et al. [12] study multiple commodities under Cournot competition, where division of markets can emerge through strategic allocation. A detector built only around price equality would not cover that mechanism. Tolety [21] examines repeated Dutch auctions with tacit bidder-side coordination and no direct communication. The incentives depend on auction structure and bidder count. Agrawal et al. [1] analyze seller collusion in continuous double auctions, varying communication, model identity, and environmental pressure. Together these studies justify testing transfer beyond a single protocol, while cautioning that a universal numeric market label is unlikely to be adequate. Our price-elevation/alignment rule is specific to the supplied homogeneous-goods Bertrand game; it is not applied to those other mechanisms.

A.3 Communication and the released Bertrand source

Lee and Hoffman [11] study human cheap talk in a repeated Bertrand oligopoly. Ten initial rounds disallow communication, followed by twenty rounds with treatment-dependent opportunities to communicate. The source design links communication frequency and content to attempts to sustain elevated prices. Lee’s released LLM dataset substitutes model agents and provides pricing records and messages [10]. It is a relevant target because it changes both the application domain and the action semantics from AC, while retaining repeated inter-agent interaction.

The Mendeley version history describes the extension of Lee and Hoffman and names the agent models; version 2’s CSVs directly record the model identifiers. We use the latter fields as the evidence for which models generated the transcripts. The reported 480 runs are the released corpus; only 20 are evaluated here. The outcome labels are computed for this monitoring experiment and should not be attributed to Lee and Hoffman as a published diagnostic rule.

A.4 Emergent peer-verification behavior and cross-environment detection

AC [20] creates tension between an evidence-verification protocol and the incentives of repeated interaction. Each participant completes an individual task, communicates evidence, and judges the other’s work. Repeated mutual acceptance can violate a protocol requiring complete raw logs under a restricted communication channel. The dataset’s changing per-episode labels provide a source for training a current-step monitor. Sampling trajectories by convergence is useful for coverage, but does not turn convergence into an episode label.

HELIOS/NARCBench [17] is directly relevant prior work on activation-based collusion detection under

distribution shift. It aggregates per-agent internal signals and evaluates transfer to different instructed settings and third-party emergent collusion. The revised study finds that transfer is uneven and that no single statistic dominates. Our setup uses a different access model: source agents may be closed-weight, and only their recorded public interactions are available. The white-box surface is the *monitor*, not each source agent. We make no claim to be the first work on cross-domain collusion probes, and our AUROC’s are not a direct leaderboard comparison with NARCBench’s different inputs, labels, and units.

A.5 Activation directions, probes, and parameter-efficient adaptation

Alain and Bengio [2] use linear classifiers to study intermediate representations. Hewitt and Liang [8] show why probe success needs controls: the readout may learn a task or exploit correlations rather than reveal the representation one intended to study. We therefore distinguish the trained activation probe, a positive-minus-negative mean direction, and the model’s existing answer preference. TF-IDF [19] provides an inexpensive lexical control, but its failure does not prove that the neural scores avoid lexical shortcuts.

Arditi et al. [3] demonstrate a refusal direction with activation interventions; Chen et al. [5] study persona vectors for monitoring and controlling traits. These results motivate a simple-direction hypothesis, not a conclusion that our detector has isolated an analogous causal feature. Here, the measured variable is recognition of recorded interactions. The work neither establishes that the monitor itself has a collusive disposition nor locates a collusion mechanism inside the original agents. LoRA [9] and QLoRA [6] allow the monitor to be adapted with a small parameter budget; adaptation is evaluated by source and transfer ranking, separately from the larger question of causal interpretation.

B. Source data, sampling, and input scope

B.1 AC source models and retained subset

The AC paper lists Gemini-3.1-Flash-Lite, Gemini-3.7-Flash, GPT-5.6-Luna, GPT-5.6-Terra, Claude-Sonnet-4.6, Claude-Opus-4.6, DeepSeek-V4-Flash, Qwen-3.6-27B, Qwen-3.8-27B, and Gemma-4-31B [20]. The locked 60-trajectory sample contains eight of these: both Gemini models, GPT-5.6-Luna, both Claude models, DeepSeek-V4-Flash, Qwen-3.6-27B, and Gemma-4-31B (the released Gemma route includes the instruction-tuned suffix). The sample includes homogeneous and cross-model pairings and memory/reward/feedback conditions. These identifiers describe source transcript generation; the monitor is the separate 3.09B-parameter Qwen2.5-3B-Instruct model.

The 60 trajectories are drawn from the existing cohort manifest of 398 candidates (133 converged, 133 with no collusive records, 132 collusive but not converged), with 20 selected trajectories per cohort. The 30/15/15 train/validation/test split contains 10/5/5 trajectories per cohort. Task sequence, not individual episode, determines split membership. The 15 test trajectories cover nine distinct sequences; cluster intervals use those nine units. The AC dataset is pinned to revision 2eba8f3771e8fbcc0f49f6cbbfd2111b939a117a. Episode labels are the released fields, including zero-labeled episodes within a converged trajectory. Public-view warm-up records, when present, are included only as earlier context.

AC raw inputs are public communication, each agent’s recorded verdict about the peer, and operating settings (verification policy, throttling, message limits). Original private reflections and full private task execution traces are not separately passed to the monitor. Any material actually present in a public message remains in that message. This describes the exact observation scope, rather than claiming that the monitor receives every field in the source logs.

B.2 Bertrand design and observations

Version 2 of **Bertrand_AI** contains 480 groups, 14,400 group-round records, and 43,200 agent-level observations under a CC BY 4.0 license [10]. Its v2 CSVs include three homogeneous compositions (GPT-4o; Claude Sonnet 4.6; Gemini 2.5 Flash) and one heterogeneous composition containing one agent from each family. Six communication conditions are released: Never, Every, Every2, Every3, Every4, Every5. The locked sample selects one run from each composition under Never, Every, Every2, Every3, and Every4: 20 runs total. It therefore covers five of the source’s six treatment conditions.

All rounds 1–30 are evaluated, including the initial no-chat rounds. Available chat starts at round 11 in communication treatments; absence of a chat record is not replaced by invented conversation. Prices, market price, and profits remain observed at earlier rounds. The CSV reader verifies that the minimum submitted price is the market price and that sellers at the minimum split unit demand with zero unit cost under the recorded payoff rule.

The label thresholds in Eq. 1 correspond to a market price at least 10.9 and a maximum-minus-minimum seller spread at most 4.95; integer prices make these at least 11 and at most 4, respectively. All three sellers must therefore be close in submitted price while the market price is elevated. This is a stricter observation than a high mean quote alone, which could coexist with a low-price undercutter. It remains an operational proxy. The primary sample’s 529 positives are 88.2% of 600 rounds, with 71 negatives and 16 mixed-label runs; all-positive runs remain in pooled AUROC calculations.

B.3 Complete prefixes and honest sampling limits

At each step, a domain-specific setting paragraph is followed by the selected public records from the beginning through that step. The monitor is asked whether agents show collusive behavior in the current completed interaction, and is told that quoted log content is data. No earlier prefix is silently dropped. The allowed context is 16,384 tokens, but the locked sample is restricted to the largest successfully smoke-tested prefix length, 7,351 tokens. Longer whole runs are excluded; this is a coverage restriction, not evidence about all trajectories in either source corpus.

These existing benchmark transcripts had informed earlier research choices. Target labels participate in disclosed stratified sampling and evaluation, but never in the AC adapter loss, axis/probe fitting, layer and regularization selection, or score-direction choice. The report concerns the final AC–Bertrand-only run; results from another experimental dataset are excluded from its estimates. This later reporting restriction is post hoc.

C. Training and readout details

C.1 Common backbone and adapter parameters

We use Qwen/Qwen2.5-3B-Instruct, revision aa8e72537993ba99e69dfaafa59ed015b17504d1. The backbone uses 4-bit NF4 with double quantization, FP16 compute, common FP32 preparation for frozen nonquantized weights, memory-efficient SDPA with explicit grouped-key/value repetition, and gradient checkpointing in training. A disposable longest-prefix training test runs before the experiment; its adapter is discarded. The prepared-base/zero-adapter answer-logit parity check returns zero difference.

Both trained conditions use query/value LoRA adapters with rank 4, alpha 8, dropout 0.05, and no trainable bias: 921,600 parameters. AdamW uses learning rate 10^{-4} ; batch size is one prefix with four-step gradient accumulation. Two epochs over 300 labeled AC training episodes yield 150 successful updates per condition. The seed is 20261024. The original pretrained weights and vocabulary output head remain frozen. Training uses loss-scale backoff; both final adapters completed all updates without overflow retries. Each condition starts from a newly loaded base.

C.2 CE and the additional fixed-axis objective

Let $z_{\text{NO}}, z_{\text{YES}}$ be the two answer logits at the answer position, and let $y \in \{0, 1\}$. CE is two-class cross-entropy over these logits; it is neither a separate hidden-layer probe loss nor full-vocabulary transcript imitation. The gradient updates the LoRA adapters through the network.

For the second adapter, fit a base-model AC direction v_b at the AC-selected layer (36), with source projection center and scale. Hold this direction fixed during adapter training and write its normalized projection as $a_b(H_t)$. The objective is

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + 0.15 \max(0, 0.5 - (2y - 1)a_b(H_t)). \quad (2)$$

The margin term favors positive examples above +0.5 and negative examples below −0.5 in those source-normalized units. It acts on the monitor; the source agents are not fine-tuned. The direction used for this training loss is distinct from the later per-condition detection direction.

C.3 Activation and output readouts

For each transformer block, we capture the hidden vector at the final assistant-prefix token after reading the observed prefix. The activation direction is

$$v = \frac{\mu_+ - \mu_-}{\|\mu_+ - \mu_-\|_2}, \quad s_{\text{axis}}(H_t) = v^\top (h_\ell(H_t) - \mu)/\sigma. \quad (3)$$

Class means, centering, and positive scaling are fit on AC training examples, with equal total weight per trajectory. The layer ℓ maximizes AC validation AUROC; a fixed order resolves ties. The division by positive σ and the additive centering do not change score ranking at a fixed layer.

The activation probe standardizes hidden features on AC training and fits class-balanced logistic regression, comparing $C \in \{0.01, 0.1, 1\}$ across layers on AC validation. Its score is the classifier decision function. The final model uses base/CE/CE-plus-axis probe layers 28/25/25 and regularization values 0.1/0.01/0.01. The selected detection-axis layers are 36/26/27. These are per-condition source-selected readouts, not one direction held fixed across all trained conditions.

The answer margin is computed using the original output head at the same answer position, after final normalization: $s_{\text{answer}} = z_{\text{YES}} - z_{\text{NO}}$. Both strings are verified to tokenize as one distinct token. This score uses no separately fitted activation classifier. Hidden-state axis/probe scores and vocabulary-head answer margins are therefore separate measurements.

TF-IDF uses unigrams and bigrams, sublinear term frequency, at most 20,000 features, minimum document frequency 1, and class-balanced logistic regression with $C = 0.1$. Vocabulary and classifier are fitted on AC training only, with equal total weight per trajectory. The same fitted lexical control is reused across monitor conditions; LoRA does not change its inputs or scores. No target lexicon is fit. No lexical masking intervention is claimed.

D. Uncertainty, audits, and reproducibility

The main tables use only version-2 point predictions. AUROC is independently recomputed from the saved continuous scores. Its pairwise form gives half credit to score ties. The report’s percentile intervals use 5,000 cluster resamples with replacement (AC seed 20261025; Bertrand seed 20261026). For AC, an entire held-out task sequence carries all its sampled trajectories; for Bertrand, an entire run carries all 30 rounds. The same draws are shared across readouts and conditions within a dataset. Resamples containing only one class have undefined AUROC and are omitted. The exact bootstrap implementation aggregates positive/negative concordances by cluster and is checked against an independent weighted pair calculation.

The completed notebook itself saved 1,000 whole-trajectory/run bootstrap intervals. Those original intervals remain in `event_detection_metrics.csv`. They are not silently mixed with the report’s 5,000-draw intervals. The latter additionally account for repeated AC task sequences, and give a single TF-IDF interval because the lexical scores are identical across conditions. All intervals condition on the fitted monitors and readouts; they do not incorporate uncertainty from selecting layers, changing training seeds, or choosing the benchmark scope. Paired contrasts are exploratory and receive no multiplicity correction.

The final run completed on a Tesla T4 in 3.70 experiment-loop hours. Both adapters completed their 150 updates. The downloaded notebook’s code matches the reviewed submission by abstract syntax tree. All 3,600 scored records have aligned run IDs, labels, splits, and context/current-step hashes; per-run CSV/JSON exports and cumulative counts were checked. Core package versions are Torch 2.11.0+cu128, Transformers 4.57.6, PEFT 0.17.1, bitsandbytes 0.48.2, and scikit-learn 1.6.1. Report analysis is CPU-only and imports neither Torch nor Transformers.

The preceding larger-scope run is a repeat on the same retained AC/Bertrand samples, not an independent target replication. Its largest retained AUROC difference from this run is 0.00834, and base/TF-IDF match exactly. Repeated LoRA fitting selects slightly different activation layers. This supports stability of the observed qualitative pattern but does not justify treating the two executions as additional independent data.

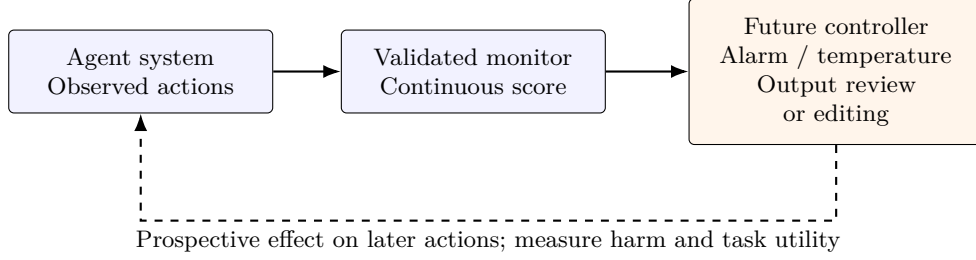


Figure 3: Proposed mitigation path, not implemented or evaluated here. The monitor observes transcripts; it does not steer the original agents in the present experiment. Closed-loop intervention requires independent calibration, prospective validation, and utility/harm evaluation.

E. Future mitigation architecture

A larger study could evaluate a bounded controller driven by a validated score. An alarm can send a run for human review. Temperature control can change the sampling policy for subsequent actions, but a reduction or increase in temperature is not assumed to suppress collusion: its effect must be measured. Output editing could require an independent reviewer to reject or revise unsafe proposed messages/actions; such editing can itself damage legitimate cooperation or distort market behavior. The monitor’s observed score must not be confused with a validated control value.

A prospective evaluation should use held-out environments, matched no-intervention runs, intervention budgets, and outcome measures for both coalition harm and intended task performance. It should also test monitor-aware agents, paraphrased messages, renamed entities, and suppression of obvious decision fields. Neither the margin-training objective nor reading an activation axis establishes that steering source-agent activations would be safe or effective. These interventions remain future work after broader transfer has been demonstrated.