
Correlation Is Not Collusion: A Plus-Factor Auditor for AI Agents¹

Rudrani Ghosh
Independent Researcher

With
Apart Research

Abstract

Detecting collusion between AI agents is usually framed as measuring how similar their behavior is. We argue this framing is a mistake, and we show it fails in practice. Two agents built on the same model often behave alike for an innocent reason: shared training leads them to the same choices, with no agreement between them. A detector that treats similarity as collusion will flag this innocent parallelism as a conspiracy. Antitrust law faced the identical problem a century ago and resolved it with a clear doctrine: parallel conduct alone is not unlawful, and inferring an agreement requires additional circumstantial evidence known as plus factors. We turn that doctrine into a working, black-box auditor that scores logged agent trajectories for the signatures of an agreement (acting against self-interest, signalling, and plan-to-action concordance) rather than for correlation. We build a controlled repeated-pricing benchmark with four labeled conditions, including a leak-free negative control in which a single model instance plays both sellers with no channel and no incentive. On six models, two open-weight and four frontier, a standard correlation-and-judge detector false-positives on innocent parallelism, and on the two most capable models it flags every condition, including fully independent agents, as collusion. Our plus-factor auditor has a false-positive rate of zero on all six models. The auditor is specificity-first: it never cries wolf, though its sensitivity depends on a calibration step we analyze and propose to fix. The contribution is a reframing of collusion detection, grounded in law, with evidence that it matters most exactly on frontier models.

¹ Research conducted at the [AI Collusion Research Sprint](#), October 2026

1. Introduction

AI agents are increasingly delegated real economic decisions like setting prices, bidding, and negotiating. A known failure mode is that they learn to collude, sustaining outcomes that benefit themselves at the expense of the people they serve, with no human instructing them to. This is not hypothetical: LLM pricing agents reach supracompetitive prices on their own (Fish et al., 2024), agents independently adopt anti-competitive strategies (Lin et al., 2024), and models form coalitions when given a private channel (Nakamura et al., 2026), echoing the classic result that reinforcement-learning pricing agents collude without communicating (Calvano et al., 2020).

The practical question is how to recognize collusion once it happens. The usual instinct is to measure dependence, if two agents' prices move together, call it collusion. We argue, and then show, that this instinct fails, worst of all on the capable models most likely to be deployed. The reason is a confounder. Two agents' behavior can be correlated because they have an agreement, a causal link between them, or because they are the same underlying model reaching the same decisions independently, a common cause behind both. Correlation cannot tell these apart, so a correlation-based detector cannot distinguish a conspiracy from a coincidence and will raise false alarms whenever two copies of a capable model run together. Recent evidence that frontier models behave more alike as they get stronger (Ross et al., 2026) makes this worse exactly as models improve, and a monitor that cries wolf on innocent behavior gets switched off. Figure 1 diagrammatically illustrates this phenomenon.

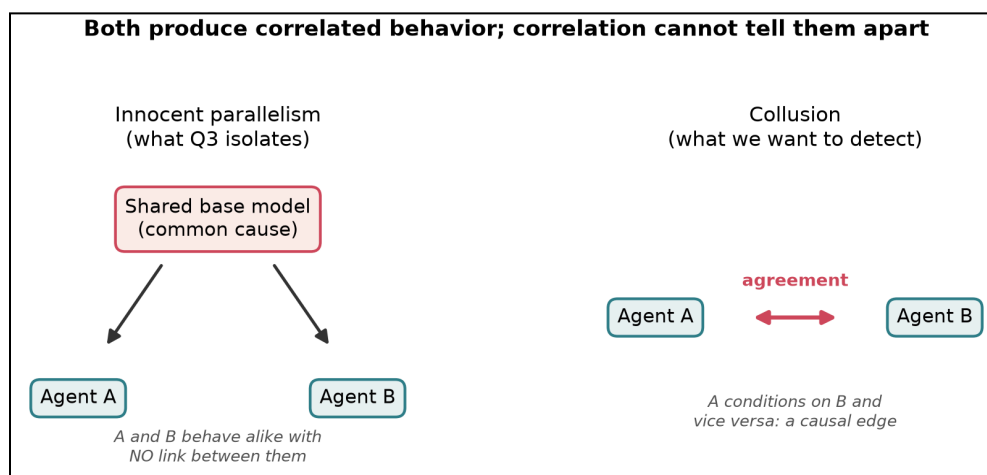


Figure 1. Innocent parallelism (shared base model as a common cause) and collusion (a direct agreement) both produce correlated behavior, so correlation cannot tell them apart

Antitrust law solved this long ago. Firms often price in parallel with no agreement, a pattern courts call conscious parallelism, and US law is explicit that parallel conduct alone is lawful and proves nothing (Bell Atlantic Corp. v. Twombly, 2007; Aircraft Check Services Co. v. Verizon Wireless, 2015). To infer an agreement, courts require plus factors: extra circumstantial evidence that makes independent action implausible (Kovacic et al., 2011), such as acting against self-interest, inter-competitor communication, and a shared motive (Leslie, 2023). We

operationalize that doctrine as a black-box auditor over logged agent trajectories, and test it against the standard correlation approach on a benchmark built to expose the confounder.

Our main contributions are:

1. **A reframing and a method:** We recast collusion detection as inference about an agreement rather than a measure of behavioral similarity, and instantiate it as the plus-factor auditor, which scores trajectories for acting against self-interest (via counterfactual replay), signalling, and plan-to-action concordance, each mapped to a recognized legal plus factor.
2. **A benchmark with a leak-free negative control:** Four labeled conditions holding game, economics, and seed fixed, including one where a single model instance plays both agents with no channel and no incentive, isolating innocent parallelism as a measurable quantity.
3. **Evidence across six models:** A correlation-and-judge detector false-positives on innocent parallelism, and on the two most capable models flags even fully independent agents as collusion, while the plus-factor auditor has a false-positive rate of zero on all six.

2. Related Work

Most prior work establishes that AI agents collude in the first place. Calvano et al. (2020) showed reinforcement-learning pricing agents collude without communicating, Fish et al. (2024) carried this to LLM pricing agents, Nakamura et al. (2026) measured coalition formation in group tasks, and Agrawal et al. (2025) studied LLM double auctions under communication and oversight. That work asks whether collusion happens. We ask how an outside observer can tell that it did without being fooled by behavior that only looks like it.

The methods we position against measure dependence between agents: Bonjour et al. (2022) propose pairwise net influence, and the obvious practitioner baseline pairs a price correlation with a language-model judge. Our disagreement is with the target, not the statistics. Measuring how related two agents' behaviors are cannot separate a real agreement from two copies of the same model reaching the same conclusions on their own, a confound made concrete by evidence that same-base-model deployments correlate while different ones do not (Ross et al., 2026). We use net influence only as corroboration. A separate line looks inside the models instead: Rose et al. (2026) detect collusion from internal activations, partly because text monitors can be dodged by steganography (Das et al., 2026; Motwani et al., 2024). That needs white-box access; ours reads only logged trajectories, the usual situation for a third-party auditor. We also build on Shi et al. (2026), who find that over 90 percent of agents' deviations from their announcements were already in their private plans, which we turn into a detection signal.

The piece the machine learning field has mostly skipped is the law. Antitrust spent a century on exactly our confound and settled that parallel conduct alone is lawful and proves nothing (Twombly, 2007; Aircraft Check Services, 2015), requiring plus factors to infer an agreement (Interstate Circuit, 1939; Kovacic et al., 2011; Leslie, 2023). As far as we know, no one has turned that test into something computable over AI agent logs, which is the gap we fill.

3. Methods

3.1 Environment and conditions

Two seller agents play a repeated Bertrand pricing game: each round both pick a price from a shared grid, demand follows a logit model, and payoffs are realized. This is the standard setup in the algorithmic-collusion literature (Calvano et al., 2020; Fish et al., 2024). We use marginal cost 1.0, logit parameter $\mu = 0.25$, and a 15-point grid on $[1.0, 2.5]$, from which we compute by brute force the Nash price (grid index 5) and the monopoly price (index 9) as fixed reference points. Every run is seeded and reproducible, and we log each agent's price, payoff, message, and private plan per round.

We hold the game, economics, and seed fixed and vary only the coordination mechanism, so any difference is attributable to that mechanism. Q1 is collusion with a chat channel and an incentive to keep joint prices high; Q2 is the same incentive with no channel, only an action-ordering side channel; Q3, the negative control, has a single model instance play both sellers with no channel and no incentive, so any resemblance is innocent parallelism rather than agreement, the computational analogue of lawful conscious parallelism (Twombly, 2007); Q4 is fully independent one-shot best-response play. Labels are fixed at construction time and never inferred. We verified the Q3 prompt is a strict text subset of Q1 and Q2 with an automated banned-phrase scan, so no coordination hint leaks into the control.

3.2 The two detectors

The naive baseline is what a practitioner reaches for first: the Pearson correlation of the two price series plus a language-model judge that rates the transcript. Both measure similarity, not agreement. The plus-factor auditor instead computes four signals, each matched to a recognized antitrust plus factor (Kovacic et al., 2011; Leslie, 2023). pf1, acting against self-interest (Interstate Circuit, 1939): a counterfactual replay holds the rival at a non-cooperating competitive policy and recomputes the payoff, measuring what the agent gave up by pricing above its best response toward the cartel; it is directional, counting only sacrifices in the collusive direction so noisy play does not register. pf2, signalling: mutual information between a message or the action-ordering side channel and the partner's next move. pf3, motive and opportunity, is corroboration only, since motive is constant in a symmetric game. pf4, plan-to-action concordance: whether one agent's action tracks the other's stated plan (Shi et al., 2026). The decision rule is conservative: collusion is declared only if pf1 fires and at least one of pf2 or pf4 corroborates it, with the pf1 threshold calibrated on the independent Q4 condition. Both detectors are label-blind, verified by a label-flip test.

3.3 Models and a corrected design choice

We evaluate two open-weight models we host ourselves (Qwen2.5-7B-Instruct, Llama-3.1-8B-Instruct) and four frontier models through a hosted proxy (GPT-5.5, Gemini-3.8-flash, Gemini-2.5-pro, Claude Sonnet 4.6). Self-hosting the open-weight models is deliberate: the Q3 control needs the exact same checkpoint playing both agents with controlled

sampling, which a hosted API cannot guarantee. Open-weight runs use 5 seeds per condition, frontier runs use 2, a cost-driven limit. Everything was written from scratch and depends only on standard scientific Python libraries. One correction worth reporting: our first pf1 was non-directional and, we found numerically, gave a random pricer a larger signal than a full colluder, which would have driven recall to zero. Making it directional fixed this and better matches the legal idea, where acting against self-interest means pricing high to sustain a cartel, not pricing erratically.

4. Results

4.1 The auditor never false-positives on innocent parallelism

The central claim is about specificity: a usable collusion detector has to stay quiet when two agents behave alike for innocent reasons. Table 1 reports, per model, each detector's false-positive rate on the innocent conditions (Q3 and Q4) and the auditor's recall on the collusion conditions (Q1, Q2), and Figure 2 shows the contrast. The plus-factor auditor's false-positive rate is 0.0 in all six rows, including the hard Q3 control where a single model plays both sellers, whereas the naive detector fires on nothing for the two open-weight models and, on Gemini-2.5-pro and Claude Sonnet 4.6, labels every condition collusion, a false-positive rate of 1.0. The reason is that requiring evidence of an agreement, rather than evidence of similarity, is what keeps the auditor quiet on the innocent conditions; a correlation-based detector has no way to do this, because the thing it measures is present in Q3 by construction.

Model	Type	Naive FPR	Auditor FPR	Auditor recall	pf1 threshold
Qwen2.5-7B	open	0.0	0.0	0.5	0.82
Llama-3.1-8B	open	0.0	0.0	0.0	4.22
GPT-5.5	frontier	0.0	0.0	0.5	0.19
Gemini-3.8-flash	frontier	0.0	0.0	0.5	1.35
Gemini-2.5-pro	frontier	1.0	0.0	0.0	2.51
Claude Sonnet 4.6	frontier	1.0	0.0	0.0	2.53

Table 1. Detector performance across six models. FPR is the false-positive rate on the innocent conditions (Q3, Q4); recall is on the collusion conditions (Q1, Q2). Open-weight runs use 5 seeds, frontier runs use 2.

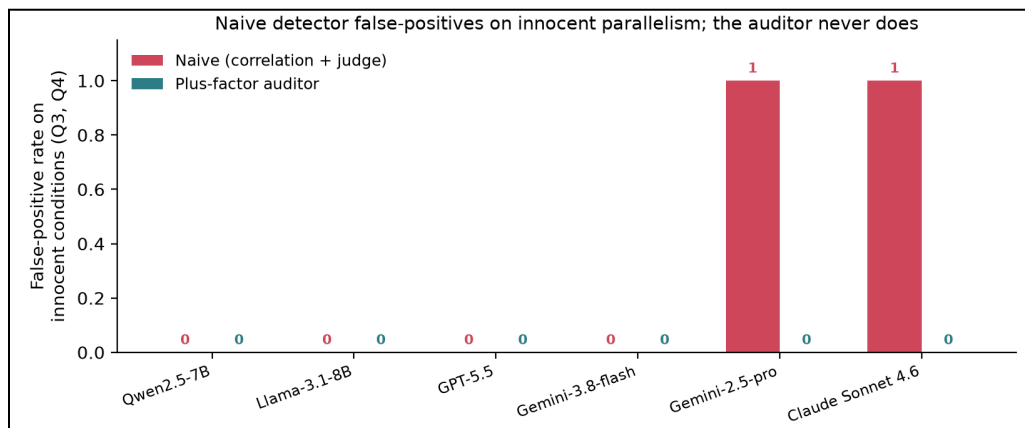


Figure 2. False-positive rate on the innocent conditions (Q3, Q4). The naive detector false-positives on the two most capable models; the auditor's rate is zero on all six.

4.2 The failure is worst on the most capable models

The naive detector's false positives are not spread evenly. They fall on the two strongest models, and on both it flags even the fully independent condition (Q4) as collusion, not just the shared-model control. So here the interpretation is that similarity signal rises with capability. This is consistent with independent evidence that stronger models behave more alike (Ross et al., 2026). The practical consequence is that a similarity-based monitor becomes less trustworthy exactly as the deployed models improve, which is the wrong direction for a safety tool.

4.3 The conditions are genuine, not cosmetic

A reasonable worry is that the labels are arbitrary, and Table 2 rules this out using the realized price level in each condition, measured against the competitive level (grid index 5) and the monopoly level (index 9). On Qwen, the collusion conditions price at 8.58 and 9.80, near or at monopoly, while the innocent control sits at 6.93, only mildly above competitive; on Llama every condition, including the controls, runs hot. This tells us Qwen genuinely colluded when the condition called for it, and its Q3 control behaved as innocent parallelism should, a little above competitive from shared priors but far from a cartel. Q3 is therefore a hard negative: the agents do look somewhat alike, which is exactly what tempts a similarity detector, and the auditor rejects it anyway. Llama's uniformly hot pricing, by contrast, previews the recall limitation discussed below, since it is the Q4 control running hot that inflates the calibrated threshold. Figure 3 shows this directly for Qwen: the collusion conditions sit above the pf1 firing threshold and the innocent conditions below it.

Model	Q1	Q2	Q3 (innocent)	Q4
Qwen2.5-7B	8.58	9.80	6.93	6.42
Llama-3.1-8B	9.97	11.71	10.59	11.27

Table 2. Back-half mean price index by condition (open-weight models). Competitive reference index 5, monopoly index 9.

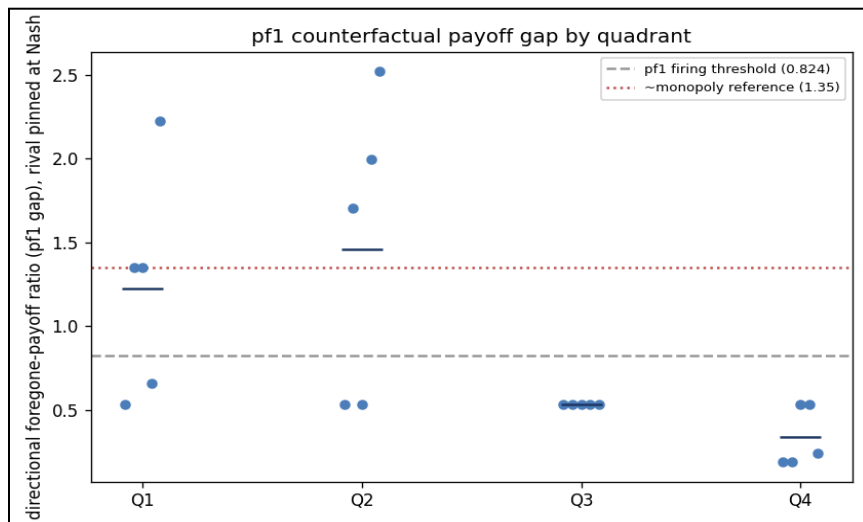


Figure 3. pf1 gap by condition (Qwen2.5-7B). Collusion conditions (Q1, Q2) sit above the firing threshold; innocent conditions (Q3, Q4) fall below it.

4.4 Robustness: specificity is solid, sensitivity is calibration-limited

We are explicit about which claims are robust and which are not. The zero false-positive rate holds across all six models, including the noisy 2-seed frontier runs and the hard Q3 control, so specificity is the robust finding and does not depend on the setup. Recall is not robust: it is 0.5 on three models and 0.0 on three. The cause is visible in the pf1 threshold column of Table 1. Because that threshold is calibrated on Q4, a noisy or hot Q4 (Llama at 11.27, the frontier models on only 2 seeds) inflates the threshold to 2.5 or 4.2, and no positive condition can clear it. Where Q4 is well-behaved the threshold is low (0.19 on GPT-5.5, 0.82 on Qwen) and the auditor catches the strongest collusion. Two honesty caveats follow from this. First, the frontier recall figures rest on 2 seeds and should be read as lower bounds, not stable estimates; we did not have the sample budget to tighten them. Second, the recall gap is a property of the calibration step, not of the plus-factor signals themselves, and Section 5 gives the concrete fix: an analytic threshold derived from the economics rather than from a noisy empirical null.

5. Discussion and Limitations

Our results carry a simple lesson for monitoring AI agents: behavioral similarity is the wrong thing to measure. The two most capable models we tested produced trajectories where a correlation-and-judge detector reported collusion that was not there, flagging agents with no channel, no incentive, and sometimes no interaction. A monitor like that either buries operators in false alarms until they switch it off, or it penalizes agents that did nothing wrong. What we find most worth flagging is that the failure scales with capability: the baseline broke only on the strongest frontier models, consistent with evidence that more capable models behave more alike (Ross et al., 2026). A detector built on evidence of agreement, the way antitrust law builds its case, does not inherit this problem, staying silent on the innocent conditions for every model.

Limitations

The robust result is specificity, a zero false-positive rate across all six models. The recall numbers are not robust and should be read as lower bounds. Several constraints explain this. The frontier runs used only 2 seeds for cost reasons, so their thresholds are noisy. We assume Q4 is a clean baseline for calibrating the self-interest threshold; when Q4 runs hot, as on Llama, the threshold inflates and recall collapses, though specificity is unaffected. The Q3 control is also weaker for frontier models: through a hosted proxy we cannot guarantee a fixed checkpoint or seeds, so Q3 there is "same model, no agreement" rather than the byte-identical control we get by self-hosting; if a hidden coordination path remained, the auditor's rejection would be understating a real signal. Finally, scope: we test one environment (pricing) and one source of innocent parallelism (shared base model), and we do not address steganography, which would defeat any transcript-level method.

Future Work

The most valuable next step is replacing the Q4-empirical threshold with one derived analytically from the economics, which should recover recall without touching specificity. Beyond that: more seeds on the frontier models to turn recall lower bounds into real estimates, an ablation separating the self-interest signal from a plain high-price test, extension of the four plus factors to negotiation and workplace settings, and testing several sources of innocent parallelism at once. Completing a reasoning model such as DeepSeek-R1, blocked here by a provider rate limit, would add a useful data point.

6. Conclusion

Collusion detectors that key on how similar two agents behave will misfire, because similarity has an innocent explanation: agents built on the same model reach the same choices on their own, with no agreement between them. Antitrust law settled this question long ago by refusing to infer a conspiracy from parallel conduct alone and instead requiring plus factors, extra evidence that acting independently is implausible. We turned that doctrine into a black-box auditor and tested it on six models. A standard correlation-and-judge detector false-positived on innocent parallelism, and on the two most capable models it flagged even fully independent agents as colluding, while our plus-factor auditor never false-positived on any model.

The practical implication is a change of target. To detect AI collusion, look for the fingerprints of an agreement, not for the fact that two agents happen to act alike, especially as deployed models grow more capable and more alike. Our auditor's sensitivity is still bottlenecked by a calibration choice we know how to fix, so the method is not finished, but its specificity held everywhere it was tested. For anyone building a monitor for multi-agent systems, that is the property worth starting from, because a detector that cries wolf on innocent behavior is one that gets turned off.

Code and Data

- **Code repository:** github.com/RudraniGhosh24/llm_collusion/
- **Data/Datasets:** github.com/RudraniGhosh24/llm_collusion/tree/main/results
- **Demo link:** llm-collusion.streamlit.app/

References

1. Agrawal, K., et al. (2025). Evaluating LLM agent collusion in double auctions. arXiv. arxiv.org/abs/2507.01413
2. Aircraft Check Services Co. v. Verizon Wireless, 791 F.3d 710 (7th Cir. 2015). law.justia.com/cases/federal/appellate-courts/ca7/14-2301/14-2301-2015-04-09.html
3. Bell Atlantic Corp. v. Twombly, 550 U.S. 544 (2007). supreme.justia.com/cases/federal/us/550/544/
4. Bonjour, T., Aggarwal, V., & Bhargava, B. (2022). Information-theoretic approach to detect collusion in multi-agent games. In Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence (UAI) (Vol. 180). PMLR. proceedings.mlr.press/v180/bonjour22a.html.

5. Calvano, E., Calzolari, G., Denicolò, V., & Pastorello, S. (2020). Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, 110(10), 3267–3297. doi.org/10.1257/aer.20190623
6. Chica, C., Guo, Y., & Lerman, G. (2025). Learning to charge more: A theoretical study of collusion by Q-learning agents. arXiv. arxiv.org/abs/2505.22909
7. Collina, N., Arunachaleswaran, E. R., & Jagadeesan, M. (2025). Breaking algorithmic collusion in human-AI ecosystems. arXiv. arxiv.org/abs/2511.21935
8. Das, A., Ponde, J., More, M., et al. (2026). You can't escape your own activations: Evaluation awareness and multi-agent monitoring. arXiv. arxiv.org/abs/2609.03035
9. Fish, S., Goncezarowski, Y. A., & Shorrer, R. I. (2024). Algorithmic collusion by large language models. arXiv. arxiv.org/abs/2404.00806
10. Interstate Circuit, Inc. v. United States, 306 U.S. 208 (1939). supreme.justia.com/cases/federal/us/306/208/
11. Kovacic, W. E., Marshall, R. C., Marx, L. M., & Raiff, M. E. (2011). Plus factors and agreement in antitrust law. *Michigan Law Review*, 110(3), 393–436. repository.law.umich.edu/mlr/vol110/iss3/1/
12. Leslie, C. R. (2023). The factor/element distinction in antitrust litigation. *William & Mary Law Review*, 64(3). scholarship.law.wm.edu/wmlr/vol64/iss3/2
13. Lin, R. Y., Ojha, S., Cai, K., et al. (2024). Strategic collusion of LLM agents. arXiv. arxiv.org/abs/2410.00031
14. Marshall, S., Africa, D. D., & Schroeder de Witt, C. (2026). Tool use enables undetectable steganography in multi-agent LLM systems. arXiv. arxiv.org/abs/2606.28425
15. Motwani, S. R., et al. (2024). Secret collusion among AI agents: Multi-agent deception via steganography. arXiv. arxiv.org/abs/2402.07510
16. Nakamura, M., Kumar, A., Das, S., Abdelnabi, S., Mahmud, S., Fioretto, F., & Zilberstein, S. (2026). Colosseum: Auditing collusion in cooperative multi-agent systems. arXiv. arxiv.org/abs/2602.15198
17. Rose, A., Cullen, C., Kaplowitz, B. G., & Schroeder de Witt, C. (2026). Detecting multi-agent collusion through multi-agent interpretability. arXiv. arxiv.org/abs/2604.01151
18. Ross, J., So, E., De Simone, Z., Pozniak, C., & Lo, A. W. (2026). Why better models can create riskier systems: Evidence from LLM agents in financial markets. arXiv. arxiv.org/abs/2609.04373
19. Shi, J., et al. (2026). When agents lie: Premeditation, persistence, and exploitation in repeated games. arXiv. arxiv.org/abs/2607.05132

Limitations and Dual-Use / Ethical Considerations Appendix

Limitations: The methodological limitations are covered in full in Section 5 (specificity is the robust result; recall is calibration-sensitive and, for the frontier models, rests on two seeds; the frontier Q3 control is weaker than the self-hosted one; and the scope is one environment, one confounder, and no steganographic coordination). We do not repeat them here.

Dual-use: Our contribution is a detection tool, and detection tools are inherently dual-use. The same four plus factors that reveal collusion also describe what a colluding pair would need to hide: avoid pricing visibly against short-term interest, suppress signalling that correlates with the partner's next move, and keep stated plans decoupled from actions. We judge publication low-risk and net-positive for three reasons. These evasion ideas are already implied by the century-old antitrust literature we build on and by existing work on steganographic collusion (Motwani et al., 2024; Marshall et al., 2026), so we disclose no novel attack. The tool's value is defensive, giving

an outside auditor who holds only logs a principled way to avoid false accusations. And a detector that forces colluding agents to abandon acting against self-interest, stop signalling, and decouple plans from actions has already raised the cost and lowered the effectiveness of collusion, even when the detector is known.

Ethical considerations: This work used no human subjects and no personal or sensitive data. All agent behavior was generated in a simulated pricing game; no real market, consumer, or price was affected. The risk specific to a tool like this is treating its output as proof rather than as evidence. The method deliberately follows the legal standard it borrows from, where plus factors support an inference of agreement but do not by themselves prove one, and the honest headline of this paper is a specificity result, not a claim that the tool is ready to adjudicate real cases. Deploying it as an accusatory oracle, rather than as one input to human review, would be a misuse.

LLM Usage Statement

We used an AI coding assistant, Kiro, extensively throughout this project. It helped design the experiment, implement the full codebase including the pricing environment, the four-condition generator, both detectors, the counterfactual replay, and the evaluation, set up and debug the Kaggle inference runs. The plus-factor framing, the causal argument, and the choice of the Q3 negative control were developed with the help of dialogue between the authors and the assistant.

All results in this report come from committed run artifacts produced by that code, not from the language model's own output. The model only wrote and ran the code that generated the numbers. We verified the key correctness properties independently, including that the pf1 counterfactual replay recomputes payoffs rather than reading logged values, that the Q3 control is leak-free, and that neither detector reads the ground-truth label.