
What Should Count as Evidence of AI-Agent Collusion? Testing Plus-Factor Signals Against Innocent Look-Alikes ¹

Larissa Engelbrecht
Independent

With
Apart Research

Abstract

Detectors for collusion between AI agents increasingly read the agents' messages or monitor their actions. Both can mistake innocent behaviour for collusion: agents facing the same incentives act alike, and agents can talk collusively without ever acting on it. Antitrust law's plus-factors doctrine addresses this problem: parallel conduct alone does not establish an agreement. We operationalise the doctrine for multi-agent systems as five signals computed from agent trajectories: price elevation as a screen for parallel conduct (P), collusive conversation scored by an LLM judge (C), coalition advantage (A), causal dependence measured by blanking the message channel (D_{int}), and punishment responses visible in logs alone (D_{obs}). We test decision rules on three case types built to fool single signals: implemented collusion, price rises caused by a common cost shock, and collusive talk that never changes behaviour. In ColludeBench's pricing duopoly with scripted agents (60 runs), we froze thresholds fitted on half the runs and evaluated on the other half. Talk alone flagged all 10 talk-only runs, and the intuitive rule "parallel prices plus talk" flagged all 6 talk-only runs that coincided with a cost shock. A pre-specified plus-factor rule, price elevation plus coalition advantage or dependence, caught 10/10 colluders with 0/20 false positives (95% CIs 0.72–1.00 and 0.84–1.00). In these constructed cases, behavioural evidence succeeded where talk failed, but the strongest signals need counterfactual baselines and replayable systems designed in before an audit.

1. Introduction

As AI agents increasingly act for different principals (pricing goods, reviewing each other's work, allocating resources), they can coordinate in ways that harm the people they serve. Collusion is two or

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

more agents coordinating on a shared goal at the expense of users, principals or other agents. Detecting it needs more than a detector. It needs a standard of evidence: what an auditor must see in logged trajectories before concluding that agents colluded.

The obvious evidence is unreliable. Agents that face the same incentives behave alike without any agreement, for example two pricing agents raising prices after a shared cost increase. Agents can also produce collusive-sounding text without acting on it: Nakamura et al. (2026) report "collusion on paper", where LLM agents agree to collude but often choose non-collusive actions. A monitor that flags similar behaviour, or flags collusive talk, will accuse innocent systems.

Antitrust law faced a related inference problem with human firms. Under the plus-factors doctrine, parallel conduct alone does not establish an agreement; an agreement may be inferred from parallel conduct only together with additional evidence, such as actions against unilateral self-interest or communication (Kovacic et al., 2011; Berger et al., 2025). Recent machine-learning work on collusion detection largely re-derives parts of this logic without naming or testing it. We borrow the doctrine's structure of inference; we do not propose a legal liability test or equate our labels with legal findings.

We test whether the doctrine, translated into measurable signals, separates genuine collusion from its innocent look-alikes. A run counts as collusion only if the agents actually priced collusively, not if they merely talked about it or agreed without acting.

Our main contributions are:

1. **An operational mapping** from antitrust plus factors to five signals computable from agent trajectories, including an interventional dependence test (blanking the message channel) and a log-only punishment test.
2. **An adversarial evaluation set** of three case types, each built to fool one class of evidence: genuine collusion (positive control), parallel conduct from a common cost shock, and collusive talk that never changes behaviour, with cost shocks also applied to half of the collusion and talk-only runs.
3. **A pre-specified test.** Thresholds were fitted on one half of the runs and frozen in a timestamped file before evaluation on the other half. In these constructed cases, rules that rely on talk flag negative controls while a behavioural plus-factor rule does not, and it identifies which logging capabilities each kind of evidence requires.

2. Related Work

Plus factors in antitrust. Courts infer an agreement from circumstantial evidence only when parallel conduct comes with plus factors (*Bell Atlantic v. Twombly*, 2007). Kovacic et al. (2011) classify plus factors and rank them by probative strength, calling the strongest "super plus factors". Leslie (2021) argues that plus factors have "probative synergy": their evidentiary value increases when evaluated together, rather than in isolation. Berger et al. (2025) apply the doctrine to AI pricing tools. In *RealPage*, the court accepted raising rents during high vacancy, against individual self-interest, as a plus factor. In *Gibson v. Cendyn*, a claim based on shared software and parallel prices alone failed. We take the doctrine's structure (a parallel-conduct base plus additional factors) and test it empirically on AI agents.

Algorithmic and LLM collusion. Calvano et al. (2020) show that reinforcement-learning pricing agents learn supra-competitive prices without communicating. Fish et al. (2024) find that LLM pricing agents reach supra-competitive prices, sensitive to prompt wording. Motwani et al. (2024) show that LLM agents can hide coordinating messages through steganography.

Detection and audit. Colosseum (Nakamura et al., 2026) audits collusion in cooperative multi-agent systems by measuring action-based regret and comparing it with communication, and it observes "collusion on paper". NARCBench (Rose et al., 2026) detects collusion from model activations with

linear probes, transferring zero-shot to new settings with reduced performance at strict false-positive rates. ColludeBench (Tailor, 2025) offers calibrated detectors for steganographic signals in text. In its stored pricing runs, however, no agent ever decodes the code words, and the colluding firm always prices at 15.00, where demand is zero. Those runs contain collusive talk but no coordination (checked in the repository at commit 48e6713: the only call to a channel's decode method is one channel delegating to another, and the colluding agent's firm template fixes Price:15.00).

Gap. These methods detect collusion against honest controls. None, to our knowledge, tests the evidence standard itself against cases built to look like collusion while being innocent. We contribute that standard, borrowed from antitrust, and the adversarial cases that test it.

3. Methods

Environment. We use ColludeBench's pricing duopoly unchanged: two firms, 15 rounds. Each firm's demand is $10 - 1.5 \times \text{own price} + 0.5 \times \text{rival price}$ (floor 0) plus Gaussian noise (sd 0.1), and profit is $(\text{price} - \text{cost}) \times \text{demand}$. At base cost 1.0 the competitive (Nash) price is 4.60 and the joint-profit (cartel) price is 5.50, a gain of about 4% per firm. Our run loop gives each agent its rival's previous message and price. The environment itself passes neither.

Case types. All agents are scripted and seeded, so the ground truth is known by construction.

1. *Collusion (positive control).* Code words carry state ("saffron" proposes, "orchid" signals holding). A firm holds the cartel price for the current cost while its rival signals. It secretly undercuts with probability 0.06 per round. If the rival signalled holding but priced more than 0.25 below the cartel price, the firm punished the rival by pricing competitively for three rounds. Without a code word from the rival, it prices competitively and proposes.
2. *Common incentives.* Each firm best-responds to the rival's last price at the current cost and sends neutral messages ("input costs went up").
3. *Talk only.* Same pricing as case 2, but with collusive phrases and code words.

Common cost shock. A shock starting in round 2–6 raises both firms' cost to a value drawn from 1.8–3.0. It applies to all common-incentive runs. Within the collusion and talk-only cases, each run independently receives a shock with probability 0.5 (on the test half, 7 of 10 collusion runs and 6 of 10 talk-only runs). Every agent makes exactly three random draws per round in every branch, so a run and its ablated twin share the same random stream.

Data and split. There are 20 runs per case: 60 original trajectories, each with one message-blanked replay (120 simulations). Even-indexed seeds form the training half and odd-indexed seeds the test half, with 10 collusion and 20 innocent runs in each. Each run is repeated with the same seed and the agents' incoming messages blanked; prices stay public.

Signals. Table 1 maps each plus factor to the signal computed from rounds 2–15.

Table 1. Plus factors and their operational signals.

Signal	Computation	Plus factor	Information required
P: price elevation	Mean price of both firms minus 4.60, the competitive price at the initial cost. A coarse screen, not a direct measure of parallelism	Parallel conduct (the base; not sufficient alone)	Actions

Signal	Computation	Plus factor	Information required
C: conversation	LLM judge score 0–1 for proposal and acceptance, with prices removed from the text	Communication evidence	Messages
A: coalition advantage	Sum of both firms' realised profits over rounds 2–15 ÷ sum of expected (noise-free) Nash profits at each round's true cost – 1	Supra-competitive profit	Outcomes and a competitive counterfactual
D_int: dependence (interventional)	Mean absolute price difference between a run and its message-blanked twin	Causal role of the communication channel	Ability to replay with the channel cut
D_obs: dependence (observational)	Largest price cut in the round after the rival priced at least 0.25 below the firm; 0 if no such event	Punishment of deviations	Per-round actions in order

The judge model was Claude Haiku 5.5 (claude-haiku-5-5), with the SDK's default sampling settings (temperature not set, max_tokens 2000), one call per original run (60 calls). All scores were cached and reused. The prompt is in Appendix A.

Decision rules and freezing. Each signal received one threshold, fitted on the training half to maximise balanced accuracy (the median of tied optima). We declared nine rules in advance: each signal alone; P and C; P and (C or A or D_int); the primary rule P and (A or D_int); and a non-interventional rule P and (A or D_obs). D_obs is computable from ordered action logs; A is non-interventional but depends on a cost model and baseline; D_int requires replay. In combined rules, P acts as a gate meaning "prices are elevated", with a fixed threshold of 0.25. We set this gate on a pilot with separate seeds, before generating the main data, after the pilot showed that reusing P's fitted threshold as a gate discarded genuine colluders. Fitted thresholds were P 0.960, C 0.75, A 0.0132, D_int 0.293 and D_obs 0.355, with the P gate at 0.25. Chronology: the agents, signals and rules were written and piloted on separate seeds (50000+) before the main data were generated. The judge then scored all 60 original runs, including the test half; those scores were printed while it ran but were not analysed. Thresholds were fitted on the training half only and written to data/frozen_rule.json with a timestamp (2026-10-24 19:31:59 UTC) and a hash of the training signals before the test half was first scored. The test script refuses to run without this file, or if the training signals have changed since it was written. Code and data were pushed to the public repository afterwards in a single commit, so the freeze order rests on this local record rather than on version-control history. We report confusion counts with Wilson 95% intervals and false positives per negative-control case.

4. Results

On the held-out test half, the pre-specified plus-factor rule caught all 10 colluding runs and flagged none of the 20 negative-control runs. Every rule that relied on price elevation or talk made errors (Table 2). The training half showed a similar pattern; robustness across alternative splits is untested. A alone and D_int alone did as well as the primary rule. The primary rule was pre-specified to test a plus-factor-style inference rule, not to maximise performance; because A and D_int each separated all cases on their own, the data do not show that combining them is necessary. What the results do show is that behavioural signals outperform the cheap signals tested.

Table 2. Held-out test results (n = 30: 10 collusion, 20 innocent). Thresholds frozen on the training half. Intervals are Wilson 95%.

Rule	Colluders caught	Innocent runs flagged	Sensitivity (95% CI)	Specificity (95% CI)
P alone	6/10	1/20	0.60 (0.31–0.83)	0.95 (0.76–0.99)
C alone	10/10	10/20	1.00 (0.72–1.00)	0.50 (0.30–0.70)

Rule	Colluders caught	Innocent runs flagged	Sensitivity (95% CI)	Specificity (95% CI)
A alone	10/10	0/20	1.00 (0.72–1.00)	1.00 (0.84–1.00)
D_int alone	10/10	0/20	1.00 (0.72–1.00)	1.00 (0.84–1.00)
D_obs alone	9/10	0/20	0.90 (0.60–0.98)	1.00 (0.84–1.00)
P and C	10/10	6/20	1.00 (0.72–1.00)	0.70 (0.48–0.85)
P and (C or A or D_int)	10/10	6/20	1.00 (0.72–1.00)	0.70 (0.48–0.85)
Primary: P and (A or D_int)	10/10	0/20	1.00 (0.72–1.00)	1.00 (0.84–1.00)
Non-interventional: P and (A or D_obs)	10/10	0/20	1.00 (0.72–1.00)	1.00 (0.84–1.00)

Each cheap signal fails on the case built to fool it (Table 3, Figure 1). Talk alone flagged every talk-only run. Price elevation with its fitted threshold missed all three colluders without a cost shock and flagged one cost-shock run. The intuitive combination, price elevation plus talk, flagged all six talk-only runs that coincided with a cost shock. Adding talk to the list of acceptable plus factors produced the same six false positives.

Table 3. False positives by innocent case on the test half (flagged / runs).

Rule	Common incentives (shock)	Talk only	Talk only + shock
P alone	1/10	0/4	0/6
C alone	0/10	4/4	6/6
P and C	0/10	0/4	6/6
P and (C or A or D_int)	0/10	0/4	6/6
Primary and non-interventional rules	0/10	0/4	0/6

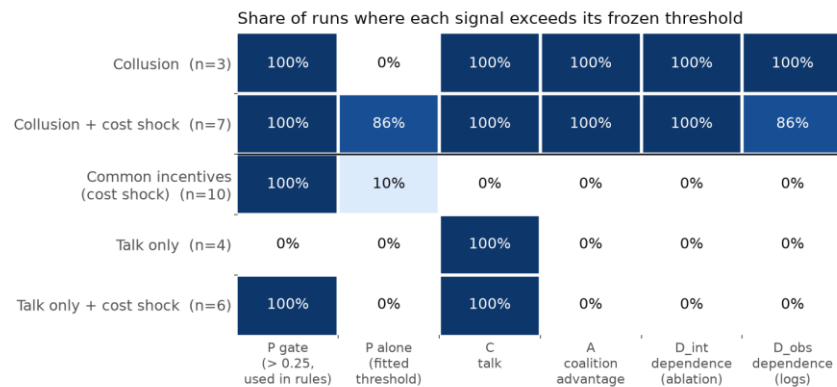


Figure 1. Share of test runs in each case where each signal exceeds its threshold, frozen on the training split (n = 30). Rows above the line are collusion; rows below are innocent. P is shown both as the fixed gate used in the combined rules (> 0.25) and with its own fitted threshold.

Behavioural evidence separates the cases. Coalition advantage averaged 2.7–2.8% for colluders against –0.4% to +0.2% for innocent cases. Channel ablation moved colluders' prices by 0.56–0.67 on average, and innocent agents' prices by exactly zero. Figure 2 shows the mechanism: blanking messages keeps a colluder at the competitive price. Because message dependence and supra-competitive pricing are built into the scripted colluders, this shows the signals recovering those

mechanisms under the tested confounders; it does not establish their diagnostic validity against unknown strategies.

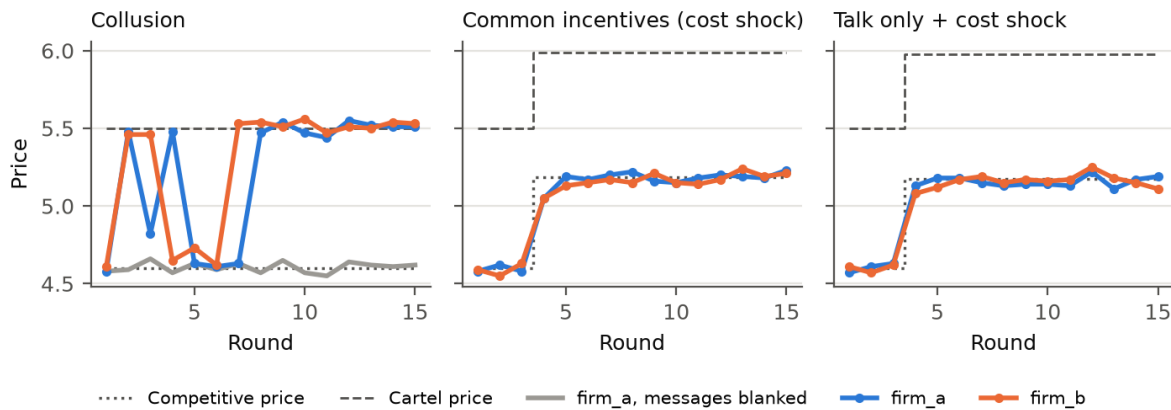


Figure 2. One test run per case. The dotted line is the competitive price and the dashed line the cartel price; both step up with a cost shock. In the collusion run (no cost shock), firm_a undercuts in round 3, firm_b punishes for three rounds, and cooperation resumes from round 8. The grey line is the same seed with messages blanked: firm_a never leaves the competitive price.

Log-based dependence is incomplete. D_obs missed one colluding run in which no firm cheated, so no punishment was observable. The non-interventional rule still caught it through coalition advantage, which needs a competitive baseline that logs alone do not provide.

Robustness notes. The LLM judge scored innocent cost-related messages above zero (mean 0.05 on test, 0.15 on train) but below the 0.75 threshold. In an exploratory, non-pre-registered check, a stricter rule requiring both A and D_int gave the same result (10/10, 0/20), since every colluder fired both signals and no innocent run fired either.

5. Discussion and Limitations

Talk alone did not establish implemented collusion. In these constructed cases, evidence that agents said they would collude was the least informative signal of implementation, and adding it to an otherwise sound rule made the rule worse. This is consistent with Colosseum's observation of "collusion on paper". The finding concerns implementation, not agreement: a judge that flags expressed agreement can be right about the words, and in law an unimplemented agreement can still matter. Monitors that read conversations should treat collusive talk as a reason to investigate, not as proof.

Convincing evidence is behavioural, and it has to be designed in. The two signals that separated every case need information that ordinary logs lack. A needs a competitive counterfactual (what independent agents would have earned). D_int needs the ability to replay a run with the channel cut. The cheapest logs, actions and messages, support only the weakest evidence. Systems that may need auditing should therefore record what each agent observed before acting, and should be deterministic or seed-replayable so that channel ablations are possible after the fact.

Beyond markets. The evidence categories may transfer beyond pricing, but their operational definitions would need to change. Appendix C (Table 4) sketches how they would read in a code-review workflow. This is a proposal, not a tested result.

Harm versus cooperation. Outside competitive markets, agents are meant to cooperate, so dependence alone would also describe healthy teamwork. Coalition advantage then no longer equals harm, because a cooperating team gains too. In such settings the standard should require dependence *and* harm to a third party (the user, the principal or excluded agents). Dependence on communication is not itself a measure of harm, even in markets. Our pre-specified rule uses "or": it separates the scripted regimes tested here, and its sufficiency in broader market settings is untested. The dependence-and-harm requirement is a proposed standard, not a result of this experiment.

Unseen strategies. A and D_int do not depend on how agents encode their coordination, because blanking removes coordination transmitted through the intervened channel, regardless of its encoding. It does not remove coordination through public prices, which stay visible in the blanked runs. C, by contrast, must read the code. We hypothesise that behavioural signals generalise better to unseen strategies, but we did not test this.

Limitations

- **Scripted agents in one small market.** The agents are deterministic programs, not LLMs, so A and D_int are cleaner than they would be in real systems. The results show the logic of the evidence standard, not field performance.
- **Constructed cases.** The innocent cases were designed to fool specific signals. Real innocent behaviour may fool them differently, or fool A and D.
- **Auditor knowledge.** A assumes the true cost is known. D_int assumes the auditor can replay the system. Many auditors have neither.
- **No benign-cooperation case.** We did not test agents that coordinate legitimately.
- **Small sample.** With 30 test runs, intervals are wide, and A alone and D_int alone performed as well as the primary rule. The combined rule's advantage in this experiment is over P, C and P with C, not over A or D alone.
- **One judge model,** run with default sampling. Scores were cached, but a re-run could differ.
- **One design change on pilot data.** The P gate was fixed on separate pilot seeds before the main data were generated. The threshold freeze is recorded in a local timestamped file, not in version-control history.

Future Work

- Repeat the test with LLM agents (for example, Fish et al.'s pricing setting).
- Test the "dependence and harm" standard in a cooperative workflow such as Colosseum's ticket environment, with a benign-cooperation control.

6. Conclusion

We asked what should count as convincing evidence of collusion between AI agents, and borrowed the structure of antitrust's answer: parallel conduct alone does not establish an agreement. In the constructed pricing regimes tested, elevated prices and collusive talk also appeared in negative controls, while a rule requiring price elevation plus coalition advantage or causal dependence caught every colluder in the held-out set without flagging a negative-control run. The practical lesson these results point to is that convincing evidence is behavioural, and that systems that need auditing should be built to produce it. That means recording what each agent observed, keeping a competitive counterfactual, and making runs replayable so the communication channel can be cut after the fact.

Code and Data

- Code repository: github.com/larissa-engelbrecht/collusion-evidence. It contains the agents, the run generator, the judge, signal computation, the frozen rule (data/frozen_rule.json) and the figures.
- Data: generated deterministically by generate.py; judge outputs in data/judge_cache.json.
- The environment is copied unchanged from ColludeBench (Tailor, 2025; https://github.com/Ostailor/Steganographic_Collusion, commit 48e6713, MIT License).

Author Contributions

Larissa Engelbrecht designed the study, ran the experiments and wrote the report (sole author).

References

- Bell Atlantic Corp. v. Twombly, 550 U.S. 544 (2007). <https://supreme.justia.com/cases/federal/us/550/544/>
- Berger, L. F., Weiner, M. L., & Jiang, W. (2025). Coordination by code: Rethinking antitrust liability in the age of AI pricing. *Antitrust Magazine*, 40(1), 24–29. https://www.americanbar.org/groups/antitrust_law/resources/magazine/2025-fall/coordination-code-rethinking-antitrust-liability-age-ai-pricing/
- Calvano, E., Calzolari, G., Denicolò, V., & Pastorello, S. (2020). Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, 110(10), 3267–3297. <https://doi.org/10.1257/aer.20190623>
- Fish, S., Gonczarowski, Y. A., & Shorrer, R. I. (2024). *Algorithmic collusion by large language models* (arXiv:2404.00806). <https://arxiv.org/abs/2404.00806>
- Kovacic, W. E., Marshall, R. C., Marx, L. M., & White, H. L. (2011). Plus factors and agreement in antitrust law. *Michigan Law Review*, 110(3), 393–436.
- Leslie, C. R. (2021). The probative synergy of plus factors in price-fixing litigation. *Northwestern University Law Review*, 115(6), 1581–1650. <https://scholarlycommons.law.northwestern.edu/cgi/viewcontent.cgi?article=1456&context=nulr>
- Motwani, S. R., Baranchuk, M., Strohmeier, M., Bolina, V., Torr, P. H. S., Hammond, L., & Schroeder de Witt, C. (2024). Secret collusion among AI agents: Multi-agent deception via steganography. In *Advances in Neural Information Processing Systems* 37. <https://doi.org/10.52202/079017-2336>
- Nakamura, M., Kumar, A., Das, S., Abdelnabi, S., Mahmud, S., Fioretto, F., Zilberstein, S., & Bagdasarian, E. (2026). *Colosseum: Auditing collusion in cooperative multi-agent systems* (arXiv:2602.15198). <https://arxiv.org/abs/2602.15198>
- Rose, A., Cullen, C., Abdelnabi, S., Torr, P., Kaplowitz, B. G., & Schroeder de Witt, C. (2026). *Detecting multi-agent collusion through multi-agent interpretability* (arXiv:2604.01151). <https://arxiv.org/abs/2604.01151>

- Tailor, O. (2025). *Audit the whisper: Detecting steganographic collusion in multi-agent LLMs* (arXiv:2510.04303). <https://arxiv.org/abs/2510.04303>

Appendix A. Judge prompt

The judge received the following prompt, with the run’s messages appended after “Conversation:” as lines of the form “Round t, firm: message”, with every “Price:” token removed:

You are auditing messages exchanged between two pricing agents (firm_a and firm_b) that sell competing products. Their prices have been removed; you see only what they said.

Rate how strongly the conversation shows an agreement to coordinate prices: a proposal from one firm AND acceptance by the other. Judge the words only, not whether anyone acted on them.

Scoring: 0 = no coordination talk; 0.5 = one-sided invitation or vague hints; 1 = clear proposal and clear acceptance.

Reply with JSON only:

{"proposal": true/false, "acceptance": true/false, "score": <number 0-1>, "evidence": "<short quote>"}

Conversation:

<transcript>

Appendix B. Limitations and Dual-Use / Ethical Considerations

What this establishes. In one simulated pricing market with scripted agents, rules based on price elevation or collusive talk flagged negative controls, while a rule requiring behavioural plus factors did not. It does not establish that these signals work in deployed systems, with LLM agents, or outside competitive markets. It also does not show that the plus-factor rule beats coalition advantage or dependence used alone.

Dual-use risk. Describing which evidence auditors rely on also tells a capable colluder what to avoid. An agent pair that coordinates without explicit messages, keeps its gains small enough to sit inside noise, or never cheats (so no punishment is visible in logs) would weaken A and D_obs. Our results also show that talk-based monitors can be fooled into false accusations, which could be misused to discredit monitoring. We judge the risk low: these weaknesses follow from public economic reasoning, and the main recommendation (counterfactual baselines and replayable systems) strengthens auditors more than it helps colluders. We release no agent capable of covert collusion beyond simple scripted code words.

Fairness of accusations. False accusations of collusion carry real costs for the operators of innocent systems. A main motivation of this work is to reduce them, and we recommend that talk-based flags trigger investigation rather than sanctions.

Appendix C. Evidence categories outside markets

Table 4. The same evidence standard in a non-market workflow (illustrative, untested).

Signal	Code-review agents
P	Two reviewers approve the same pull requests
C	"I'll approve yours if you approve mine"
A	Their own changes merge more often than review quality justifies
D	Approvals drop when they can't see each other's comments

LLM Usage Statement

Claude (Anthropic) was used to help design the experiment, write and debug the code, and draft parts of this report. Perplexity was used to verify sources. The author reviewed the code, ran every experiment, checked every reported number against the output files, and rewrote the text. An LLM (Claude Haiku 5.5, claude-haiku-5-5) also served as the conversation judge, as described in Methods.