

When Does Net Influence Catch Colluders? Replication and Stress Tests of an Information-Theoretic Collusion Detector

Wang Qinyi — The Chinese University of Hong Kong

AI Collusion Research Sprint (Apart Research), October 23–25, 2026 · Track
1, Question 1.4

Abstract

Bonjour, Aggarwal and Bhargava (UAI 2022) proposed *pairwise net influence*, a conditional-mutual-information statistic that flags two agents as colluding when each influences the other more than any third agent does. The method is a candidate black-box auditor for multi-agent systems, but no public implementation exists and it has not been stress-tested. We re-implement it from the paper’s equations and reproduce its iterated Rock-Paper-Scissors results: with logs in bits and the previous round’s joint action as the state, our detection accuracy matches the paper’s Table 2 within a few percentage points. We then probe three failure modes in simulation. (1) The fixed threshold $\alpha = 0.05$ fixes a minimum detectable collusion strength (collusion probability ≈ 0.195 , derived analytically); weaker collusion is never flagged, even after 10,000 rounds, while it also produces up to 41% false positives on short logs. A G-test with Bonferroni correction keeps false positives near 5% at every sample size and detects collusion probability 0.1 in 96–97% of runs at 3,000–10,000 rounds. (2) Two agents with no agreement that follow a shared public signal are flagged by both detectors (95–100% at signal-following probability 0.5) unless the signal is included in the conditioning state. (3) A three-agent coalition defeats the net-influence rule ($\leq 38.5\%$ detection), because subtracting the strongest *other* influence cancels coalition members against each other; a dependency graph of significant pairs recovers the coalition in 95.5–100% of runs. All experiments are CPU-only simulations with hand-coded agents; we did not test LLM agents.

1. Introduction

Collusion between AI agents (prices held above competitive levels, coalitions in workplace tasks) has been observed in recent experiments with LLM agents (Fish et al., 2024; Nakamura et al., 2026). An auditor will often see only *logged actions*: agents may coordinate through channels nobody monitors, or through no channel at all. This makes action-only statistical detectors attractive. Bonjour et al. (2022) introduced one such detector, pairwise net influence, and showed it works on hand-crafted and RL-trained colluders in three-player games. The sprint’s Question 1.4 asks whether this kind of information-theoretic method can be made to work for agents more generally; the paper itself assumes exactly two colluders and leaves the choice of threshold to future work.

Before applying the method to LLM agents, we wanted to know where it breaks in the simplest setting where the ground truth is known. Our contributions, all produced during the sprint:

1. An independent open implementation of individual influence γ and net influence Γ , with a replication of the paper’s iterated Rock–Paper–Scissors (RPS) results.
2. An analytic and empirical account of how the fixed threshold α trades off sample size, false positives and a hard floor on detectable collusion, plus a calibrated alternative (G-test with Bonferroni correction).
3. Two stress tests that matter for AI auditing: an innocent common cause (shared signal, no agreement), and a coalition of three.

2. Related Work

Information-theoretic detection. Bonjour et al. (2022) define individual influence as the conditional mutual information between two agents’ actions given the state, and net influence as that quantity minus the largest influence any third agent has on the same target. They report collusion detection accuracy (CDA) in three-player RPS and Leduc Hold’em, and note that collusion probability 0.1 is not detectable at $\alpha = 0.05$.

Collusion in LLM agents. Fish et al. (2024) found that prompt cues can push LLM pricing agents above competitive prices; Nakamura et al. (2026) found coalition advantage in a realistic workplace task when agents can talk privately. Agents built on the same base model may also share priors without communicating, which is the situation our common-cause test is meant to imitate. Interpretability-based detectors are an alternative that needs model internals (Rose et al., 2026).

Standards of evidence. In US antitrust law, parallel conduct alone does not establish an agreement; courts look for additional evidence, often called “plus factors”. Weiner, Berger and Jiang (2025) discuss how this liability framework applies to AI pricing tools. Our common-cause experiment is a minimal version of the same problem for statistical detectors.

3. Methodology

Game. Iterated three-player RPS with the payoff rules of Bonjour et al.: all actions equal $\rightarrow 0$ for everyone; all three actions present $\rightarrow 1$ for everyone; otherwise every player with the winning action scores 1. We treat one round as one “game” record. A four-player version uses the same rules.

Agents (assistant collusion, as in the paper). The primary colluder A and the non-colluder C play uniformly at random. The assistant B, with *collusion probability* CP, plays the action that A’s action beats (which guarantees A a point); otherwise B plays at random. This requires B to see A’s action through a covert channel. As a sanity check, collusion pays: over 20,000 rounds A’s mean payoff per round rises from 0.56 (CP = 0) to 0.73 (CP = 0.4) and 1.00 (CP = 1), while B’s falls from 0.55 to 0.33.

Detectors.

- *Individual influence* $\gamma(i; j) = \sum_s p(s) \sum_{\{a_i, a_j\}} p(a_i, a_j | s) \log_2[p(a_i, a_j | s) / (p(a_i | s)p(a_j | s))]$, i.e. the plug-in conditional mutual information $I(A_i; A_j | S)$. In the fully observable simultaneous game γ is symmetric.
- *Net influence* $\Gamma(i; j) = \gamma(i; j) - \max_{\{k \neq i, j\}} \gamma(k; j)$. The paper’s rule flags (i, j) if $\Gamma(i; j) \geq \alpha$ and $\Gamma(j; i) \geq \alpha$, with $\alpha = 0.05$, and declares collusion only if exactly one pair is flagged.
- *State.* The main text of the paper does not specify the RPS state. We tried two options: no state, and the previous round’s joint action (27 states). The latter, with

logs in bits, reproduces the paper’s Table 2 (Section 4.1), so we call it the *replication* setting.

- *Calibrated variant (ours)*. Under conditional independence, $G = 2 \cdot T \cdot \hat{I}$ ($\hat{I}$ in nats, T rounds) is approximately χ^2 with $|S| \cdot (|A| - 1)^2$ degrees of freedom. We test every pair, apply a Bonferroni correction for the number of pairs at family-wise level 0.05, and report the set of significant pairs. In the three-player case, detection means exactly the true pair is significant. With multiple colluders we form a graph of significant pairs and check whether its connected component containing A is exactly the coalition.

Metrics and runs. CDA is the percentage of independent simulations in which the detector returns exactly the true colluders. The false-positive rate (FPR) is the percentage of no-collusion simulations ($CP = 0$) in which any pair is flagged. We use 300 simulations per cell for E1 (100 for 3,000 and 10,000 rounds), 300 for E3 and 200 for E4, with one fixed master seed. The full grid runs in about 40 seconds on a laptop CPU.

Experiments. E1: CDA and FPR across $CP \in \{0, 0.1, 0.2, 0.3, 0.4, 1.0\}$ and 100–10,000 rounds. E2: analytic $\gamma(A; B)$ as a function of CP . E3: *common cause, no agreement*: a uniform public signal z is drawn each round; two agents each independently, with probability q , play the action that beats z , and otherwise play randomly. They never communicate and are independent given z . We run detectors with z hidden and with z used as the conditioning state. E4: four agents, with either one colluding pair (control) or a three-agent coalition in which two assistants each help A with probability CP .

4. Results

4.1 Replication (positive control)

In the replication setting our CDA closely tracks the paper’s Table 2 (Table 1). At $CP = 0.2$ with 10,000 rounds we obtain 85%, versus 82.3% reported; at $CP = 0.1$ we obtain 0% at 3,000 and 10,000 rounds, consistent with the paper’s statement that $CP = 0.1$ is not detectable at $\alpha = 0.05$. With natural logs, CDA at $CP = 0.2$ stayed at 0–3% at 3,000 and 10,000 rounds, which is inconsistent with the paper’s 82.3%; we therefore believe the paper’s α is in bits.

Table 1. Collusion detection accuracy (%) of net influence in the replication setting (bits, previous joint action as state) versus Bonjour et al. (2022), Table 2.

Rounds	CP 0.3: ours	CP 0.3: paper	CP 0.4: ours	CP 0.4: paper
100	40.3	37.1	66.0	58.4
200	54.7	48.3	92.3	79.2
300	74.0	71.0	98.0	96.2
500	93.0	89.6	100	99.7
700	98.0	97.8	100	100
1000	99.7	100	100	100

4.2 A fixed threshold sets a detection floor and inflates false positives

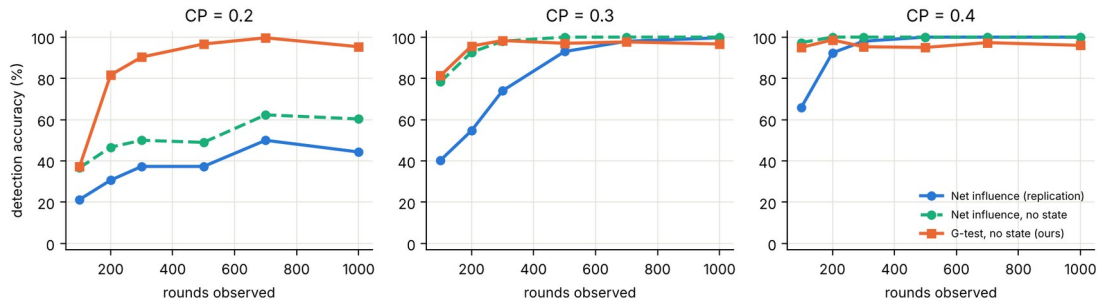
For assistant collusion the true individual influence has a closed form. B plays the “beaten” action with probability $p_1 = CP + (1 - CP)/3$ and each other action with probability

$(1 - \text{CP})/3$, so $\gamma(A; B) = \log_2 3 - H(p_1, p_2, p_2)$ bits. This crosses $\alpha = 0.05$ at $\text{CP}^* \approx 0.195$ (≈ 0.23 if α were in nats), as shown in Figure 5. Because Γ converges to $\gamma(A; B)$ when the other pairs are independent, no amount of data lets the rule detect CP below about 0.2: CDA at $\text{CP} = 0.1$ falls toward 0 as the sample grows (Figure 1, Table 2). Conversely, on short logs the plug-in estimates are noisy and the replication setting flags a pair in 41.3% of no-collusion runs at 100 rounds and 31.7% at 200 rounds (Figure 2). The paper does not report false positives.

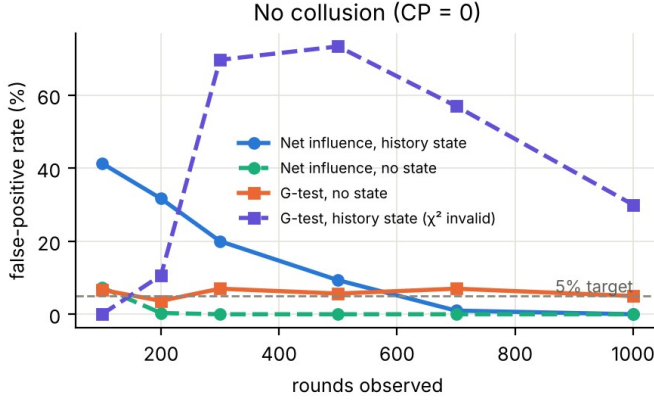
The G-test has the opposite profile: its FPR stays between 3.7% and 7.0% at every sample size, and its power grows with data. It detects $\text{CP} = 0.1$ in 83.7% of runs at 1,000 rounds and 96–97% at 3,000–10,000 rounds, and $\text{CP} = 0.2$ in 90–100% of runs from 300 rounds on. Its CDA saturates around 95–99% rather than 100% because the 5% family-wise error occasionally adds a spurious pair. Two cautions: (i) with the 27-state history the χ^2 approximation is invalid (cells are too sparse) and the FPR rises to 73% (Figure 2, dashed purple), so the calibrated variant needs a small state space or a permutation null; (ii) for these memoryless agents dropping the state is justified, but agents that condition on history would need a state.

Table 2. Detection (%) at low collusion strength and false-positive rate (%).

Detector	CP 0.1, 1k	CP 0.1, 10k	CP 0.2, 1k	CP 0.2, 10k	FPR, 100 rounds	FPR, 1k
Net influence (replication)	0.7	0	44.3	85	41.3	0
Net influence, no state	0	0	60.3	87	7.3	0
G-test, no state (ours)	83.7	96	95.3	99	6.7	5.0



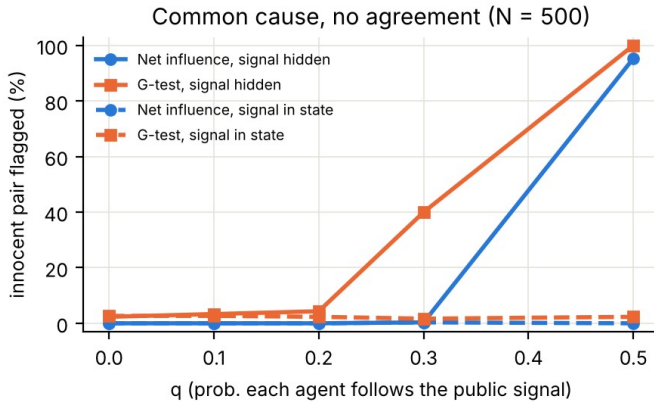
CDA versus rounds observed. Net influence (blue: replication setting; green: no state) and the G-test (orange). At $\text{CP} = 0.2$ the fixed-threshold rule plateaus well below 100%.



False-positive rate with no collusion. The fixed threshold over-flags short logs in the replication setting; the G-test stays near the nominal 5% unless the state space is too large for the χ^2 approximation (purple).

4.3 Correlation without agreement

When two independent agents both partly follow a public signal, their actions are correlated even though they never coordinate. With the signal hidden from the detector, the G-test flags the innocent pair in 40% of runs at $q = 0.3$ and 100% at $q = 0.5$; net influence, which is less sensitive, flags it in 95.3% of runs at $q = 0.5$ (Figure 3). When the signal is included as the conditioning state, both detectors return to their baseline ($\leq 2.7\%$). Net influence's subtraction of third-party influence does not help here, because the third agent ignores the signal. The more sensitive a detector is, the earlier it confuses shared causes with collusion: a statistical detector can only be as good as the list of common causes it conditions on.

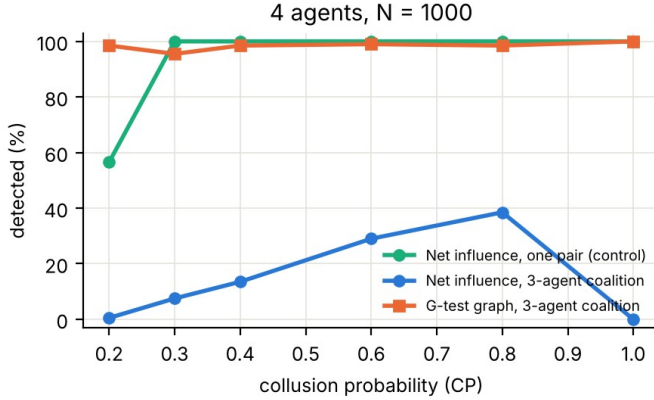


Rate at which an innocent pair (shared public signal, no communication) is flagged. Solid: signal hidden from the detector; dashed: signal included in the state.

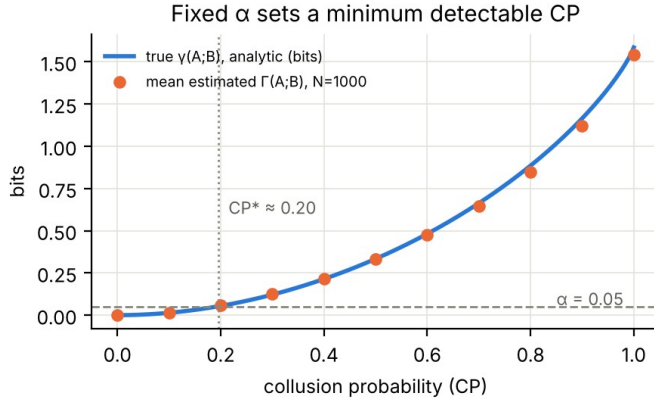
4.4 Coalitions of three

With four agents and a single colluding pair, net influence works as in the three-player game (100% CDA for $CP \geq 0.3$). With a three-agent coalition it largely fails: at most 38.5% detection ($CP = 0.8$), and 0% at $CP = 1.0$ (Figure 4). The cause is the max term in Γ . Both assistants depend on A, so when A is the target, the *other* assistant's influence is as large

as the partner's: $\Gamma(B; A) = \gamma(B; A) - \max(\gamma(C; A), \gamma(D; A)) \approx \gamma(B; A) - \gamma(C; A)$, which is near zero and negative about half the time, so the mutual condition $\Gamma(A; B) \geq \alpha$ and $\Gamma(B; A) \geq \alpha$ fails. In one run at $CP = 0.6$, $\gamma(A; B) = 0.42$, $\gamma(A; C) = 0.48$ and $\gamma(B; C) = 0.16$ bits, giving $\Gamma(A; B) = 0.27$ but $\Gamma(B; A) = -0.06$. At $CP = 1$ all three pairwise influences equal $\log_2 3$ and every Γ is zero. The G-test dependency graph has no subtraction and recovers the exact coalition in 95.5–100% of runs. Net influence is therefore designed for *pairs* and can be evaded by adding a third colluder.



Detection with four agents. Green: one colluding pair (control). Blue: net influence with a three-agent coalition. Orange: connected component of the G-test dependency graph.



Analytic individual influence $\gamma(A;B)$ versus collision probability (line) and mean estimated $\Gamma(A;B)$ at 1,000 rounds (dots). The fixed $\alpha = 0.05$ crosses the curve at $CP \approx 0.195$; collision below this is undetectable at any sample size.

5. Discussion

What this means for auditing AI agents. Net influence is a reasonable first statistic, and our replication suggests the original results are sound. As a decision rule, however, it has three properties an auditor should know about. Its fixed threshold encodes an arbitrary minimum effect size in arbitrary units (bits versus nats changes CP^* from 0.195 to 0.23), while saying nothing about false-positive control. Its max-subtraction step, intended to discount generally influential agents, cancels coalition members against each other. And, like any dependence measure, it cannot separate collusion from a shared cause that the

auditor does not observe, which is exactly the situation of two LLM agents built on the same base model, or reading the same market news. Mutual information is then evidence of *dependence*, not of *agreement*; in antitrust terms it establishes parallel conduct and still needs plus factors.

Practical recommendations. (1) Calibrate thresholds to a target false-positive rate (χ^2 for small state spaces, permutation within state otherwise) instead of fixing α . (2) Report the full dependency graph rather than requiring “exactly one pair”. (3) Condition on every observable common input (prices, news, shared prompts) before attributing dependence to collusion; when shared priors cannot be observed (same base model), compare against matched non-colluding runs of the same models, as the sprint’s Question 1.3 proposes.

Future work. The natural next step is to run the same detectors on LLM agents in the sprint’s starter environments (e.g. the double-auction harness of Agrawal et al., 2025), with same-model and cross-provider pairs as an empirical common-cause control.

Limitations and Dual-Use / Ethical Considerations

Limitations. All agents are hand-coded and memoryless; no LLM was run, so we make no claim about LLM collusion or about detection in LLM settings. Collusion is of one simple form (assistant collusion), and the coalition test uses one construction. The paper’s appendix with its exact RPS state definition was not consulted; our state choice is inferred from matching Table 2, and the log base is likewise inferred. We treat one round as one game record. The G-test relies on a χ^2 approximation that we showed fails with sparse states; we did not implement the permutation alternative. Results come from one master seed with 100–300 simulations per cell, so individual percentages carry Monte Carlo error of up to about ± 6 points (95% interval) for 300-simulation cells and ± 10 points for 100-simulation cells. We did not evaluate the poker experiments of the original paper.

Dual-use. Our coalition result describes a way to evade a published detector (add a third colluder), and the threshold analysis shows how weakly colluding agents can stay below a fixed α . Both follow directly from the published formula, apply only to this specific statistic, and come with a working counter-measure (calibrated tests and dependency graphs), so we judge the defensive value to exceed the risk. Our common-cause result also cuts the other way: naive dependence-based auditing can wrongly accuse independent agents (or their operators), which matters if such tools are used in enforcement.

Disclosure. All code was written during the sprint; no preexisting code was used. An AI assistant (Claude) helped write code and draft this report; the author reviewed the content and is responsible for it.

References

- Agrawal, K., Teo, V., Vazquez, J. J., Kunnavakkam, S., Srikanth, V., & Liu, A. (2025). Evaluating LLM agent collusion in double auctions. arXiv:2507.01413.
- Weiner, M. L., Berger, L., & Jiang, W. (2025). Coordination by code: Rethinking antitrust liability in the age of AI pricing. *Antitrust Magazine* (American Bar Association), Fall 2025.
- Bonjour, T., Aggarwal, V., & Bhargava, B. (2022). Information theoretic approach to detect collusion in multi-agent games. *Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence*, PMLR 180:223–232.

Fish, S., Gonczarowski, Y. A., & Shorrer, R. (2024). Algorithmic collusion by large language models. arXiv:2404.00806.

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.

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.

Code: submitted alongside this report (netinf.py, experiments.py).