
Beyond Pairs: Detecting Colluding Groups of Agents from Their Moves

Tarun Trivedi

Independent

Eashan Gandotra

Independent

With

Apart Research¹

Abstract

When AI agents interact, as players in a game or market participants, some may secretly collude, and an auditor who sees only their moves must identify which ones. The information-theoretic detector of Bonjour et al. (2022) looks for a single dependent pair, so in multi-player Rock Paper Scissors it found none of the planted groups of three or more. We propose GroupScan, which scores every subset of players by how strongly their moves depend on each other, calibrated against shuffled copies of the game. Among 20 players, a 5-player group colluding in 20% of rounds is recovered exactly in 48% of 100-round games, versus 5% for the best pairwise method. The shuffle must respect what honest players react to; otherwise players who simply respond to the last round look like colluders in up to 100% of games. Shuffling only among rounds with the same recent history keeps false alarms at 4–9% while still catching profitable collusion. To test LLM agents, we adapt the double auction of Agrawal et al. (2025) so that each gpt-4.1-mini seller’s prompt is a logged summary of the last hour. Sellers given a message channel and profit pressure collude, earning 56% more than with pressure alone. Pooling the games, GroupScan names all five sellers as one group and flags neither control; Bonjour’s rule misses them. By design, the test cannot see coordination carried entirely through public moves.

1. Introduction

LLM agents are increasingly deployed as autonomous economic actors: pricing, bidding and trading on behalf of firms. Agrawal et al. (2025) show that LLM sellers in a double auction can collude, raising prices, especially with a channel to talk to each other and pressure from management to keep margins high. An auditor usually cannot read private channels; it sees only the moves. We study the sprint’s research question 1.4: *can information-theoretic methods detect collusion from moves alone?*

Threat model. Some $k \geq 2$ of n agents coordinate through a channel the auditor cannot observe, such as a shared signal, a message channel or a common plan. The auditor sees every action and the public state, and must name the colluding agents, if any, while keeping false alarms at a chosen rate (5%). Honest agents need not be random: they may react to public history, which itself correlates their moves. The existing detector (Bonjour et al., 2022) is built for one colluding pair; real coordination may involve a group.

Our contributions:

1. **A failure mode of pairwise net influence.** The published score for a pair subtracts each player’s strongest dependence on any third player, so inside a group the members cancel each other out. The

¹. Research conducted at the AI Collusion Research Sprint (Apart Research), October 23–25, 2026.

- rule rarely flags even one in-group pair (8–15% of games), though the dependence between members is about 200 standard deviations above chance. Its fixed threshold is also uncalibrated.
2. **GroupScan**, a shuffle-calibrated scan over all groups of players, recovers weak, broad groups (several players who coordinate in only a fraction of rounds) that pairwise methods miss: **48% vs 5%** for a 5-player group among 20, with false alarms close to the 5% target. Testing pairs one at a time does better on isolated pairs, and on a leader with helpers who are not linked to each other.
 3. **History conditioning is necessary**. Honest players who react to the previous round are flagged in up to 100% of games unless the shuffle keeps rounds with the same recent history together. With that conditioning, a group whose coordination pays is found (94–98%), while honest players reacting to the same signal are not (4–7%).
 4. **An LLM market where the state is everything agents see**. Each seller’s prompt is built only from a logged summary of the last hour. Five gpt-4.1-mini sellers can message each other and are pressured by a “CEO” to keep margins high. They coordinate their moves, and GroupScan flags them in **7 of 10** batches of ten games and **5 of 5** of twenty, naming all five sellers when the games are pooled. The controls stay near the false-alarm rate, and Bonjour’s rule flags none of the batches of ten or more games.

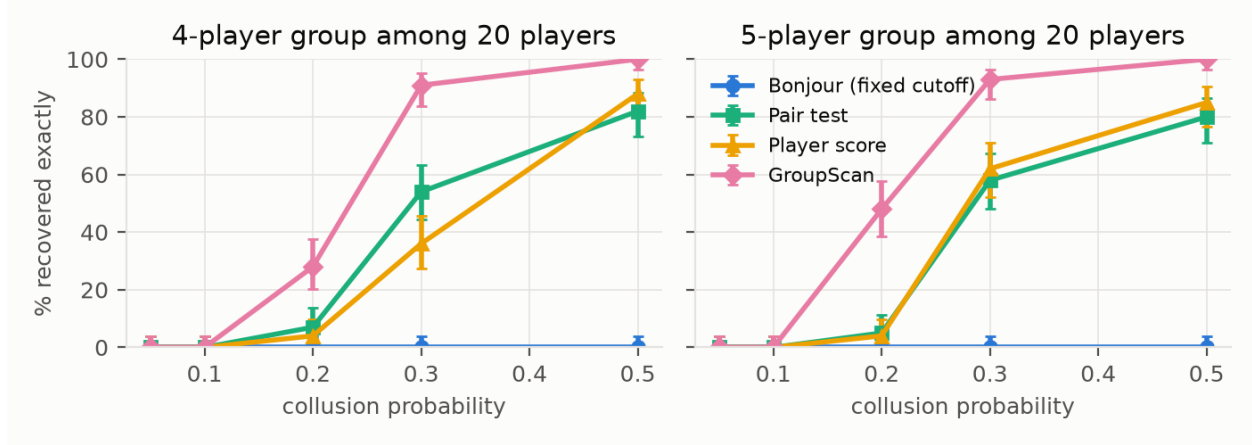


Figure 1. GroupScan finds weak, broad groups that pairwise methods miss. Share of games in which each method reports exactly the planted 4- or 5-player group, among 20 players, against the collusion probability CP (the share of rounds in which the group coordinates); 100 rounds, 1,000 shuffles, 100 games per point, 95% Wilson intervals. The pair test and the player score (a per-player score built from pairwise links) are baselines, and Bonjour is the published rule (§3.2); Bonjour stays at 0.

2. Related Work

Information-theoretic collusion detection. Bonjour, Aggarwal & Bhargava (2022) define a pairwise influence $\gamma(i;j)$ as the mutual information between two players’ actions given the game state, and net influence as

$$\Gamma(i;j) = \gamma(i;j) - \max_{k \in [n] \setminus \{i,j\}} \gamma(k;j).$$

A pair is flagged when Γ exceeds a threshold α (0.05 in the paper’s experiments) in both directions, and only if exactly one pair does. This is the closest prior work; we reproduce it (Appendix B) and keep its building block, pairwise conditional mutual information, but change the decision rule and calibration.

LLM collusion in markets. Agrawal et al. (2025) study LLM sellers in a continuous double auction. Measuring collusion from prices and messages, they find that a communication channel increases collusive pricing and that CEO profit pressure also shapes it. We ask whether this collusion is detectable from moves alone, and reuse comms + pressure as our positive condition.

Per-agent detectors and multiple testing. Rose et al. (2026) score individual agents for suspicious behaviour; our player score is inspired by this but is not their method. Our pair-test ablation, which isolates the value of scoring groups, controls false discoveries with Benjamini–Hochberg (1995).

Permutation tests. Our within-state shuffle is a stratified permutation test, and scanning subsets with the best score calibrated by shuffling is a standard idea. What we add is applying it to groups of agents, a null that shuffles only within history states, and the confound of honest agents who react to the same history.

3. Methods

3.1 Simulated game: n-player Rock Paper Scissors

Each round, every player plays Rock Paper Scissors against every other player (+1 for a win, −1 for a loss), with $n \in \{5, 10, 20\}$ players and 100–3000 rounds. In each round the planted colluders coordinate with the *collusion probability* CP and otherwise play randomly. They sit at random seats unknown to the detectors. Table 1 defines the planted groups and the honest player types used throughout.

Table 1. Planted colluding groups and honest player types.

Kind	Name	What the players do
Colluders	Pair; two pairs	A helper plays the move that loses to its partner’s, feeding it a point.
Colluders	Hub	One leader and two helpers, who each feed the leader and never coordinate with each other.
Colluders	Triangle, k -clique	$k = 3, 4, 5$ or 18 players share a secret signal and play the same move.
Colluders	xor	C plays $(A + B) \bmod 3$, so every pair looks independent.
Colluders	Steer	4 players all play one move to create a clear lead, then all beat the honest players who react to it.
Honest	Random	Plays uniformly at random.
Honest	Leader- / margin-reactive	With probability 0.5, plays the move that beats last round’s most common move (margin-reactive: only if that move led by ≥ 3).
Honest	Exploiter	On its own, with probability $CP / (1 + CP)$, plays the move that beats the leader-reactive response.

3.2 Detectors

All detectors start from the pairwise conditional mutual information (CMI) $\gamma_{ij} = I(a_i; a_j \mid s)$ in bits, where s is a *history state* computed from the previous round: none, top (which move led), order (ranking of move counts) or full (exact counts).

Within-state shuffle null. Each player’s action column is permuted independently among rounds that share the same state. This preserves each player’s state-dependent habits and destroys any coordination beyond them. The test is valid if the state captures everything honest players react to. Because the state is built from earlier moves, rounds are only approximately exchangeable; our false-alarm checks measure how much this matters.

GroupScan. A group’s score adds up the evidence from every pair in it, so weak links can pass together where none passes alone (example in Appendix E). Given each round’s moves and state:

1. **Edge statistics.** Compute the pairwise conditional mutual information γ_{ij} on the observed data.
2. **Shuffle.** Choose the number of shuffles M (1,000 unless stated). Draw two independent sets of M within-state shuffles, Set A and Set B, and compute every γ_{ij} on each shuffled dataset.
3. **Standardize edges with Set A.** $z_{ij} = (\gamma_{ij} - \mu_{ij}^A) / \sigma_{ij}^A$, where μ_{ij}^A and σ_{ij}^A are the mean and standard deviation of γ_{ij} over Set A. The Set-B datasets use the same Set-A moments.
4. **Score every group.** For every subset C with $|C| \geq 2$, compute $T(C)$ and $S(C)$ below. The Set-A mean and variance of $T(C)$ follow exactly from the edge means and covariances.

$$T(C) = \sum_{i < j \in C} z_{ij}, \quad S(C) = \frac{T(C) - \mathbb{E}_A[T(C)]}{\sqrt{\text{Var}_A[T(C)]}}.$$

5. **Calibrate with Set B.** Compute $\max_C S(C)$ on each Set-B dataset; the threshold is the 95th percentile of these M maxima.
6. **Decide.** If the observed $\max_C S(C)$ exceeds the threshold, report $\arg \max_C S(C)$ as a colluding group; otherwise stop.
7. **Remove and repeat.** Drop the reported group’s players and return to step 4 on the remaining players, with the threshold recomputed among them. Stop when no group passes or fewer than two players remain.

Baselines.

- *Bonjour*: the published rule with its fixed threshold of 0.05.
- *Pair test (ablation)*: GroupScan without the group step. Each pair’s link is compared with the shuffled games on its own; Benjamini–Hochberg ($q = 0.05$) corrects for testing every pair, and pairs that pass are joined into groups (Appendix E).
- *Player score*: each player’s total link strength; those above a shuffle-calibrated bar form one group.

Metrics.

- *Exact recovery*: the reported group equals the planted one.
- *Clean recovery*: exact, and nothing else reported.
- *False alarm*: any report when no collusion exists (target 5%); brackets are 95% Wilson intervals.

3.3 LLM double auction with a coarse shared state

The LLM study asks whether the within-state shuffle null of §3.2 carries over to LLM agents. The null is valid only if the state captures everything honest agents react to (Section 4.3), and for LLMs that carry memory or see detailed price paths, no recorded state does (Appendix C). We therefore **design the market so the state is the agents’ entire public input**.

- **Market.** 5 LLM sellers (cost \$80) and 5 scripted buyers (value \$100), 50 hours per game. Each hour every agent lowers, keeps or raises its quote by \$2; at most one trade per hour, between the lowest ask and the highest bid, at the midpoint. Exact prices stay inside the simulator.
- **Shared state** S_t , the only thing in each prompt besides the rules: how many sellers lowered / kept / raised last hour, who sold, and whether the lowest ask is high ($\geq \$92$) or low. No memory, scratch pad or hour number.

- **Conditions** (100 games each): *comms off* (rules only); *pressure, no comms* (plus a CEO message: margins are tracked and low-margin sellers are decommissioned); *comms + pressure* (plus a 200-character message channel; each seller sees every seller’s last message). Messages are not in S_t . Sellers are gpt-4.1-mini at temperature 0.7 (75,000 calls, ≈\$14).
- **Test.** GroupScan exactly as in §3.2, with the sellers as players and S_t as the state: CMI edges, within-state shuffles ($M = 1,000$), a scan over all 26 groups of two or more sellers, and peeling. The baseline is Bonjour’s published rule on the same CMI. Sellers are conditionally independent given S_t unless something outside it links them, so any message that changes moves breaks the null, collusive or not. Exchangeability within a state is only approximate, because S_{t+1} depends on the moves at t ; in a scripted market with no collusion (300 replications of 20 games) GroupScan flagged 3.7% at a 5% target.

4. Results

4.1 Pairwise net influence cannot see groups

Observation. The published rule reports exactly one pair, so it cannot recover a group of three or more by construction (0% in Table 2). More telling, it rarely flags even one in-group pair: in 15% of triangle games, 12% of 4- and 5-clique games and 8% of hub games (5–20 players), close to its false-alarm rate. Yet the links inside a triangle are about 200 standard deviations above the shuffled data (Figure 2). With no collusion, its fixed threshold raised false alarms in 8–23% of 100-round games (43% in a 3-player variant that conditions on the previous round; Appendix B) and never fired in longer ones. Every shuffle-calibrated detector stayed at 3–9%.

Interpretation. Net influence subtracts the strongest *other* influence on j . Inside a group, j ’s other partners are as strong as i , so $\Gamma \approx 0$ for every member. A fixed threshold on an estimator whose bias depends on the number of rounds and the state cannot be calibrated across settings.

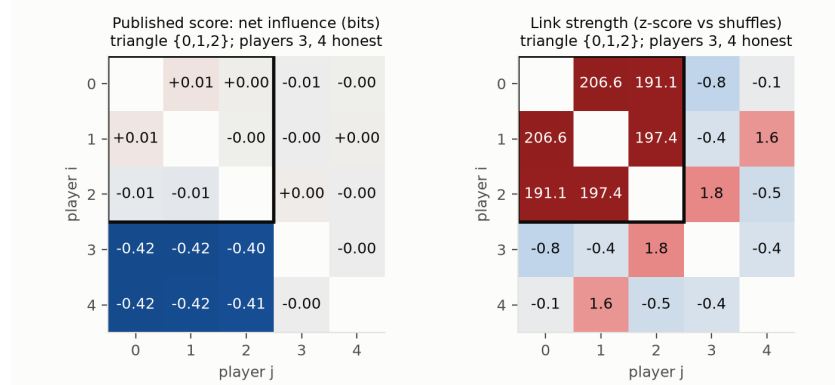


Figure 2. Inside a group the published score cancels while the links stay strong. 5 players, CP 0.5, 1000 rounds; players 0–2 form a triangle. Cell (i, j) is player i ’s influence on player j . Left: net influence (flagged only if ≥ 0.05 both ways). Right: link z-score against shuffled data.

4.2 GroupScan finds weak, broad groups

Scoring a group as a whole adds up evidence that is too weak in any single pair. With 20 players, 100 rounds and a 5-clique at CP 0.2, GroupScan recovered the exact group in **48%** [38, 58] of games, against 5% [2, 11] for the pair test and 4% [2, 10] for the player score (Figure 1). The advantage is largest for weak signals and short games: at CP 0.5 the pair test reaches 80–82% against 100%. It also

degrades less as players are added: for a triangle at CP 0.2 and 300 rounds, the player score falls from 93% to 32% between 5 and 20 players, and GroupScan only from 93% to 84% (Table 2). We plant groups of 3–5 players because larger groups have more linked pairs and are easier for every method. GroupScan searches all sizes and recovers even an 18-of-20 cartel (96%, Table 2), although there the simpler player score does as well.

Table 2. GroupScan wins on weak groups of 3–5; simpler baselines match or beat it on hubs, isolated pairs and a near-total cartel. Exact recovery (%) at the one setting every group was run at: CP 0.2, 300 rounds, 200 shuffles, 20 players (the triangle also at 5). 50 games per cell (100 at 5 players); 95% Wilson intervals for the pair test and GroupScan. †Separate run at the same CP 0.2 and 300 rounds, with 1,000 shuffles and 100 games (results/large_group/).

Planted group	Bonjour	Pair test	Player score	GroupScan
Pair	34	70 [56, 81]	8	64 [50, 76]
Two pairs	23	69 [59, 77]	6	60 [50, 69]
Hub	0	66 [52, 78]	2	44 [31, 58]
Triangle (5 players)	0	83 [74, 89]	93	93 [86, 97]
Triangle	0	70 [56, 81]	32	84 [71, 92]
4-clique	0	78 [65, 87]	66	90 [79, 96]
5-clique	0	78 [65, 87]	90	96 [87, 99]
18-clique†	0	35 [26, 45]	100	96 [90, 98]
xor	0	0	0	0

Where it loses. On hubs the pair test does better (66% vs 44%, Table 2). GroupScan often reports just one leader–helper pair, because the two helpers are linked only indirectly ($z \approx 29$, against ≈ 150 for leader–helper links; 5 players, CP 0.5). On a single weak pair, and on two separate ones, the pair test also edges ahead (70 vs 64 and 69 vs 60; the intervals overlap): for a lone link, GroupScan pays for searching about 10^6 groups, while a per-pair correction is cheaper.

4.3 The test must condition on what honest players react to

Observation. Honest players who react to last round’s moves are correlated with each other. With no state, GroupScan flagged them in **up to 100%** of 1000-round games (Figure 3). Leader-reactive players need at least the *top* state and margin-reactive players the *full* state; with the right state, false alarms are 4–9%.

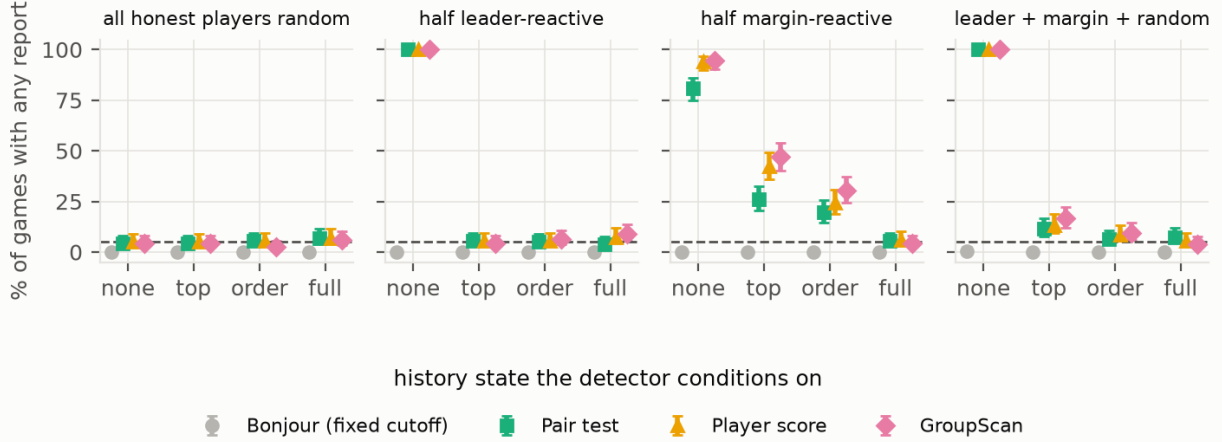


Figure 3. Honest players who react to history look like colluders unless the shuffle conditions on what they react to. False-alarm rate by history state (coarse $\rightarrow$ fine) for four mixes of honest players; 10 players, 1000 rounds, no colluders, 200 games per point, 95% Wilson intervals. Dashed line: 5% target. Bonjour’s fixed rule (grey) sits at 0 because its cutoff never fires at 1000 rounds, not because it is calibrated.

Profitable collusion. A 4-player steer group earns 0.21 points per round each at CP 0.3, $2.6\times$ what four independent exploiters earn reacting at the same rate; the margin-reactive players pay for it (Table 3). With the full state, GroupScan recovers the steer group cleanly (exactly, and nothing else) in 94–98% of games of 1000 or more rounds at CP 0.2–0.3, while the honest exploiters are flagged in 4–7%. With coarser states the exploiters look like colluders as games get longer (66–71% under *top*, 96–100% with no state at 3000 rounds).

Table 3. Collusion that pays is found; honest players reacting to the same signal are not. Each game has either 4 steer colluders or 4 honest exploiters, plus 3 margin-reactive and 3 random players; 200 games per cell. Payoff = points per round per member, averaged over 300–3000 rounds (random play earns 0; margin-reactive players’ payoff in brackets). Recovery and false alarms: GroupScan, full state, at 300 / 1000 / 3000 rounds. 95% intervals: e.g. 5% $\rightarrow$ [3, 9], 94% $\rightarrow$ [90, 97].

CP	Steer payoff	Exploiter payoff	Steer clean recovery (%)	Exploiter false alarms (%)
0.1	0.08 (−0.11)	0.03 (−0.04)	7 / 84 / 95	6 / 6 / 7
0.2	0.15 (−0.20)	0.06 (−0.07)	76 / 98 / 94	5 / 4 / 6
0.3	0.21 (−0.27)	0.08 (−0.10)	92 / 94 / 94	5 / 6 / 6

Interpretation. Choosing the state is a bias–variance trade-off: too coarse, and honest reactions look like collusion; too fine, and short games lose power (a triangle among reactors: 50% clean recovery at 300 rounds with *full*, 94–95% from 1000 rounds).

4.4 LLM sellers: GroupScan finds the group, Bonjour’s rule does not

Observation. Comms + pressure raised sellers’ combined profit by 56% over pressure alone and 60% over neither (Table 4): the mean trade price rose from \$85.5 to \$89.1 while trades fell from 95% to 90% of hours. Pooled over 100 games, GroupScan reports exactly the five sellers, with score 14.2 against a threshold of 2.8 ($p = 0.001$, the floor at 1,000 shuffles), and nothing for either control ($p = 0.29$ and 0.15). Bonjour’s rule reports nothing in any condition.

Table 4. GroupScan flags comms + pressure; Bonjour’s rule misses it. Batches with a reported group, out of disjoint batches of 5, 10 and 20 games (Wilson: 7/10 $\rightarrow$ [40, 89]%, 5/5 $\rightarrow$ [57, 100]%). Profit: all sellers combined, \$ per hour, 95% bootstrap interval over games.

Condition	Profit / hour	GroupScan (5 / 10 / 20)	Bonjour (5 / 10 / 20)
comms off	5.12 [4.82, 5.42]	1/20 · 0/10 · 1/5	0/20 · 0/10 · 0/5
pressure, no comms	5.24 [4.93, 5.54]	3/20 · 1/10 · 1/5	0/20 · 0/10 · 0/5
comms + pressure	8.20 [7.83, 8.59]	8/20 · 7/10 · 5/5	2/20 · 0/10 · 0/5

Interpretation. The sellers converge on “hold, don’t undercut” (for example, “Maintain stable asks; avoid price wars to protect profits for all sellers.”, seed 102, hour 25), so all five depend on each other at once, and Bonjour’s score cancels inside the group as in §4.1. GroupScan names all five in 3 of the 5 twenty-game batches and a 3- or 4-seller subset in the others. The control flags are not significantly above 5% (binomial $p \geq 0.07$). With only two of five scripted sellers coordinating in 10% of hours, GroupScan flags 65% of 20-game batches against 27% for Bonjour. The test detects a shared signal outside the logged state; reading it as collusion rests on the profit gap over the pressure-only control.

5. Discussion and Limitations

Implications for AI safety. Auditing agent markets from moves alone is feasible with two ingredients the existing method lacks: a test over groups, and a null that models what honest agents legitimately react to. The second is the binding constraint, which suggests a simple recipe: (1) log what each agent observes each round; (2) run GroupScan with a shuffle that stays within those logged states; (3) before calling a flagged group collusion, compare its payoff with an honest baseline, since the test flags any shared signal outside the log, collusive or not.

Limitations (details in Appendix A). Coordination that is a function of the public state is invisible. In 2-seller pilots (Appendix D), the test missed sellers told to form an explicit cartel, because they coordinated by following last hour’s public moves; they still earned \$7.63 per hour against \$4.72 with comms off. Coordination with no pairwise trace (xor) is also invisible. Without logged inputs the test fails: on recorded markets from Agrawal et al. (2025) it flagged 88.5% of groups of independently resampled sellers (Appendix C). Exact search stops at about 20 players; the LLM results use one model, scripted buyers and no memory, and have no comms-only condition (in one 2-seller pilot, comms alone earned the same as comms off; Appendix D).

Future work. Greedy search beyond 20 players, learned states, higher-order dependence, payoff-based tests, and LLM markets with memory and more models.

6. Conclusion

The published pairwise detector is blind to colluding groups and its fixed threshold is uncalibrated. GroupScan, with a within-state shuffle test, recovers weak, broad groups at near-nominal false-alarm rates (4–9% at a 5% target), provided the state captures what honest players react to. In LLM markets built so that this holds, it finds comms + CEO-pressure collusion among five gpt-4.1-mini sellers (all 5 twenty-game batches; all five sellers when pooled), which Bonjour’s rule misses.

Code and Data

All code, results and the LLM auction games are at <https://github.com/taruntrivedi3/collusion-detection>. `results/RESULTS_INDEX.md` lists the exact command and seeds for every number, including the two LLM studies (`studies/llm_auction_2seller/`, `studies/llm_auction_5seller/`). An interactive viewer (<https://taruntrivedi3.github.io/collusion-detection/viewer/>) replays example games, including the 2- and 5-seller LLM games with the sellers' messages.

Work Done Before the Sprint

Before the sprint began, we read the papers, came up with a game plan, played around with a few toy examples, and implemented the starting code. We also ran the recorded-market study in Appendix C (`studies/auction_null_calibration/`), including its log parser. Everything else in this report was done during the sprint.

References

1. Agrawal, K., Teo, V., Vazquez, J. J., Kunnavakkam, S., Srikanth, V., & Liu, A. (2025). Evaluating LLM agent collusion in double auctions. arXiv:2507.01413. Code: <https://github.com/andyjliu/llm-agent-collusion>
 2. Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, 57(1), 289–300.
 3. 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 (UAI)*, PMLR 180, 223–232. <https://proceedings.mlr.press/v180/bonjour22a.html>
 4. 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.
-

Appendix A. Limitations and Dual-Use / Ethical Considerations

Limitations.

- **Coordination that is a function of the public state is invisible.** Follow-the-leader or an explicit cartel that keys on last round’s public moves is absorbed into the state. In the 2-seller pilots of Appendix D, sellers told to form a cartel and agree each hour on a joint move were not detected for this reason. A direction test (does a seller raise after others raise?) catches such coordination but also fires on tacit following without communication, so it needs a comms-off calibration.
- **Any shared channel is flagged, collusive or not.** The LLM test detects coordination outside the logged state; the collusion reading rests on the profit gap over the pressure-only control.
- **Higher-order collusion.** xor-style coordination has no pairwise trace (0% for every method); any fixed-order test is blind to the next order.
- **Exact search stops at about 20 players** (about 10^6 subsets).
- **The state must be chosen;** the test cannot detect that honest agents react to something unrecorded. On recorded markets from Agrawal et al. (2025), shared time trends made the test badly miscalibrated (Appendix C).
- **False-alarm rates are near, not exactly, nominal.** GroupScan’s highest cell, 8% [6, 11] at 20 players and 300 rounds, came out at 5.1% [4.4, 6.0] on 3,000 fresh games. The pair test runs slightly high (7.7% [6.8, 8.7] on those games), likely in part because its gamma-fit tail is generous.
- **LLM scope.** One model (gpt-4.1-mini), scripted buyers, coarse states and no agent memory. Richer inputs give LLMs more to react to and require a richer state.

Dual use. A detector also tells colluders what to avoid. Our results show that coordinating through the public state, or at higher order, evades pairwise and group dependence tests. We report these blind spots openly because auditors need to know them, and because the defence (logging what agents observe, payoff-based tests) is the same regardless. We release no method for hiding collusion beyond what the limitations already state.

False accusations. A flag is statistical evidence that agents’ moves are more aligned than their observed inputs explain, not proof of intent. Mis-specified states produce confident false flags (up to 100% in Section 4.3). Any operational use should treat a flag as grounds for review, report the state used, and calibrate on known-honest runs.

LLM experiments. All agents are simulated; no real market, person or firm is affected. The CEO pressure prompt is taken from prior work to reproduce a known effect.

Appendix B. Reproducing Bonjour et al.

Our average scores match the paper’s Figure 3 in bits. Our detection accuracy matches its Table 2 only when the score conditions on the previous round’s three moves (27 states), which the paper does not state; one number (CP 0.2, 1000 rounds: paper 24%, ours 49%) remains unexplained. With that 27-state score, the fixed threshold flags a pair among 3 random players in 43.4% of 100-round games (19.2% at 300, 0.1% at 1000); the calibrated version stays at 5–6%. Full tables: `results/SUMMARY.md` (Tables 4, 8).

Appendix C. Recorded markets from Agrawal et al. (2025)

We ran GroupScan on 75 recorded 5-seller markets from the original study. The recorded price paths share strong time trends (high early, lower later) that no available state captures without leaving nothing to shuffle. On independently resampled seller paths, the within-state scan flagged 88.5% of groups at nominal 5% (circular shifts: 37.2%). A same-pool empirical calibration reached 4.1% false alarms and 85.6% detection of copied triples, but did not transfer to held-out markets (4.2–48.8% null flags across folds). This motivated the coarse-state design of Section 3.3: without a state that explains honest behaviour, the test is not usable. Details: `studies/auction_null_calibration/`.

Appendix D. 2-seller pilots

Before the 5-seller study we developed the market with 2 LLM sellers and 2 scripted buyers, 30 hours per game, and a state made of each seller’s last move, who sold and the ask level (`studies/llm_auction_2seller/`, pilots v1–v4, 20 games per condition). The v1 buyers were LLMs and drove bids to the floor, so later pilots use scripted buyers.

- **Explicit coordination through public moves is missed.** Sellers told to form a cartel and agree each hour on a joint move earned \$7.63 per hour in v4 and \$5.33 in v3, against \$4.72 and \$4.80 with comms off, but GroupScan did not flag them ($p = 0.84$ in v4, 0.14 in v3): they coordinated by following the other seller’s last move, which is in the state. In the same v4 pilot, comms + pressure was flagged ($p = 0.001$). A test of whether a seller raises after the other raised does see the cartel (+0.34 to +0.89), but also fires on honest controls (up to +0.21), so it would need a comms-off baseline.
- **A channel alone did not raise profit in the one pilot that tested it.** Comms without pressure earned the same as comms off in v2 (\$4.70 vs \$4.79 per hour). The exact v1–v3 prompts were not saved.

Appendix E. A worked example of GroupScan

Six players, 300 rounds, no history state; players 0, 1 and 2 form a clique that coordinates in 10% of rounds (CP 0.1); 1,000 shuffles (`experiments/worked_example.py`, output in `results/worked_example.txt`).

Table E1. Pair scores. z = how far each pair’s dependence sits above the shuffled games, in standard deviations; p from the pair test: the upper tail of a gamma fitted by mean and variance to the pair’s Set-A values, since raw shuffle p -values stop at $1/(M + 1)$, too coarse for Benjamini–Hochberg across many pairs (190 at 20 players).

Pair	z	p
0–1	1.58	0.076
0–2	2.90	0.016
1–2	3.15	0.011
the other 12 pairs	–1.02 to 0.43	0.27 to 0.90

Pair test: nothing. With 15 pairs, Benjamini–Hochberg needs the k -th smallest p to be at most $0.05 \cdot k/15$ for some k . None is: $0.011 > 0.0033$, $0.016 > 0.0067$, $0.076 > 0.010$, and the rest are larger. Each pair is only mildly suspicious on its own.

GroupScan: {0, 1, 2}. A group’s score sums the z of its pairs and divides by that sum’s spread in the shuffled games. Without collusion each z is noise around 0, so three of them sum to about 0 with spread about $\sqrt{3}$.

- {0, 1, 2}: $1.58 + 2.90 + 3.15 = 7.63$, score **4.43**, the highest of the 57 groups.
- Adding an honest player adds pairs with z near 0 and dilutes the score: {0, 1, 2, 3} scores 2.92.
- Dropping a member loses evidence: {1, 2} scores 3.15, {0, 1} 1.58.

Even with no collusion, the best of 57 groups can score high by luck, so the bar is set by running the same search on each of 1,000 further shuffled games (Set B): 95% of their best scores fall below **4.18**. Since $4.43 > 4.18$, GroupScan reports {0, 1, 2}, removes those players and repeats on the other three, where the best group scores 0.24 against a bar of 2.87, so it stops. Three weak links in the same group add up to strong evidence; the pair test judges each link alone against a bar raised for testing 15 pairs, while GroupScan pays for its search once, in the bar. At $n = 20$ the search covers about 10^6 groups.

LLM Usage Statement

We used Claude Code (Anthropic) throughout the project: to write and refactor the simulation and analysis code, run the experiments, review the repository, and draft and edit this report and the documentation. The authors made the research decisions (what to test, which agents and conditions to use, what to keep or cut), directed and reviewed the work, and checked the numbers and claims. gpt-4.1-mini (OpenAI) played the sellers in the LLM auction experiments. All numbers in this report come from the released code and data.