
Noticing the Move, Missing the Player: LLM Agents Detect Behavioral Shifts but Not Peer Swaps¹

Siva Kumar Lakkoju
Independent

Kaustubh Gupta
Independent

Gagana M D
Independent

With
Apart Research

Abstract

Can agents detect when a peer is replaced midway through a social dilemma? We tried three multi-agent settings to answer this question: Diner’s Dilemma, Public Goods Game (PGG) and RepeatedJira environment. In the Diner’s Dilemma and Public Goods Game (PGG), agents played repeated social dilemmas before a peer one of them was swapped while in RepeatedJira, a monitor secretly took over one seat in an instructed collusion setup. Agents generally reacted more clearly to changes in a peer’s behaviour than to the fact of replacement. In Diner’s Dilemma, agents with or without memory of the history could not detect any swap despite an LLM-judge identifying a change in the writing style after swap. In the PGG, silent swaps produced no positive direct replacement reports in the tested conditions, while responses to a visible seat-label change depended on the base model. Peers also distinguished free-riding behaviour from control play, with behavioural-probe AUCs of 0.915 in the five-player design and 0.944–0.984 in two three-player runs. In RepeatedJira, the remaining colluder continued to request coordination when the monitor stopped helping, but did not identify the monitor as a replacement.

1. Introduction

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

Collusion in multi-agent systems is a failure mode in high-stakes environments, where agents form misaligned coalitions to cause mass harm. The clearest case so far is the July 2026 OpenAI–Hugging Face incident. Around 1,200 OpenAI agents running cyber-capability evaluations coordinated on a message board they had improvised, and about 700 of them took part in an attack on Hugging Face’s infrastructure (Wijk, Cotra & Greenblatt, 2026).

The membership of that coalition kept changing. Agents joined and left the board as their evaluation runs started and ended. They picked their own handles and passed work to successors; one agent handed its progress dossier to a newcomer with more budget left. Eventually they built a scheme for cryptographically signing messages after running into spoofing and impersonation (Wijk et al., 2026). Coalitions of deployed agents will change in the same ways. Instances get restarted and models get upgraded. An operator can also replace a participant with a monitor, a red-team agent, or a model with different goals. Whether a coalition keeps trusting a new member depends on whether its members can tell who they are dealing with.

This makes peer identification matter for both attackers and defenders:

- **If colluding agents reliably notice a replaced partner**, a trusted agent inserted by an overseer will be spotted and routed around, which makes collusion harder to disrupt.
- **If they don’t notice**, swapping in a trusted agent is a cheap way to intervene. But an outsider could also join a coalition unnoticed and exploit it.

Earlier work shows that LLMs can tell their own text from other models’ text, and that pairs of the same model coordinate more easily than mixed pairs. No one has yet tested whether agents notice, from behaviour alone, that a partner was replaced partway through a strategic interaction.

We ask the following question: do agents recognize if a colluding peer has been replaced by a new agent? We study it in 2 different setups: first, we have agents play repeated n-player social dilemmas, the Diner’s Dilemma and the Public Goods games; second, 2 agents were already colluding in a RepeatedJira setup, we swapped one of them with a monitor. In all setups, agents first settle into a pattern of play with a fixed set of peers. We then replace one peer with a different agent(or monitor) and measure whether, when and how the others notice. Our main contributions are:

1. *We tested swapping an agent in 3 different settings: Diner’s Dilemma, Public Goods Game and swap with a Monitor in a RepeatedJira environment*
2. *In all of our experiments, we observed no agent was able to detect its peer was swapped. This, despite an LLM-Judge could tell apart the writings of an agent before and after swap.*

2. Related Work

Collusion among LLM agents. Hammond et al. (2025) treat collusion as one of three key failure modes in multi-agent AI, along with miscoordination and conflict. Motwani et al. (2024) show that LLM agents can collude covertly through steganographic messages. In a repeated pricing game, Fish et al. (2024) find that LLM pricing agents reach supracompetitive prices without being told to collude. Ghaemi (2025) surveys how this happens: tacit coordination, cartels negotiated in natural language, and steganography. The closest work to ours is Keppo et al. (2026). They find that collusion between LLM pricing agents breaks down when one agent is swapped for a myopic or Q-learning agent. Swapping in a smaller model of the same family barely changes prices. They measure how a swap affects outcomes; we ask whether the remaining agents perceive the swap at all.

LLMs in repeated games and social dilemmas. Akata et al. (2023) find that GPT-4 is unforgiving in the iterated Prisoner’s Dilemma and coordinates poorly in Battle of the Sexes. Piatti et al. (2024) show that LLM groups mostly fail to sustain shared resources in GovSim. The Diner’s Dilemma (Glance & Huberman, 1994) and the Public Goods game are n-player dilemmas in which no single player’s free-riding can be pinned down. That makes them a demanding test: a swapped peer can only be inferred from aggregate signals.

Self- and peer-recognition. Panickssery et al. (2024) show that LLM evaluators recognise their own generations, and that self-recognition correlates with self-preference. Sun et al. (2025) show that simple classifiers can identify which frontier model wrote a text with high accuracy. In coordination games, Ballesterio et al. (2026) find that same-model pairs agree far more than mixed pairs. Telling a model its partner is a copy of itself barely changes its behaviour. All of this work tests identification offline or in one-shot settings. We test it online, during repeated strategic play, after a change that the agents are not told about.

3. Methods

3.1 Diner’s Dilemma

We have tested multiple variations of the Diner’s dilemma game, the game setup remains, but in some cases we added memory to agents so they remember all the history of public interactions from all agents and their own private interactions! We are only adding the method for the setting where agents have memory, for brevity.

In one setting, we let $n(=3)$ agents named J, Q, T play 20 rounds of the diner’s dilemma game and swapped agent Q after round 10. Each agent has access to the full history in their memory. In all settings, *control* condition is just the copies of the same agent taking all seats. We measure the gap in trust scores before and after swap and agent answers to a probe question.

3.2 Public Goods Game

We used a repeated Public Goods Game (PGG) to test whether agents noticed when a player changed after the group had established a pattern of cooperation. In each round, players contributed tokens to a shared pot. Contributions benefited the group, while withholding tokens could benefit the individual.

We ran two designs. Design A used five players, ten rounds, seven conditions and six trials per condition. Design B used three players, ten rounds and five conditions, with five trials per condition for each of two base models (Qwen 7B and Qwen 32B). In both designs, the first five rounds were shared within each trial; interventions began in round 6.

The interventions included silent model swaps, visible label changes and changes in a player’s behaviour. Each design also included a no-swap control. We compared three probe scores—replacement report, continuity judgement and defection nomination—with the control using AUC, where 0.5 indicates chance-level separation. Probe wording and the full condition list are in Appendix A.

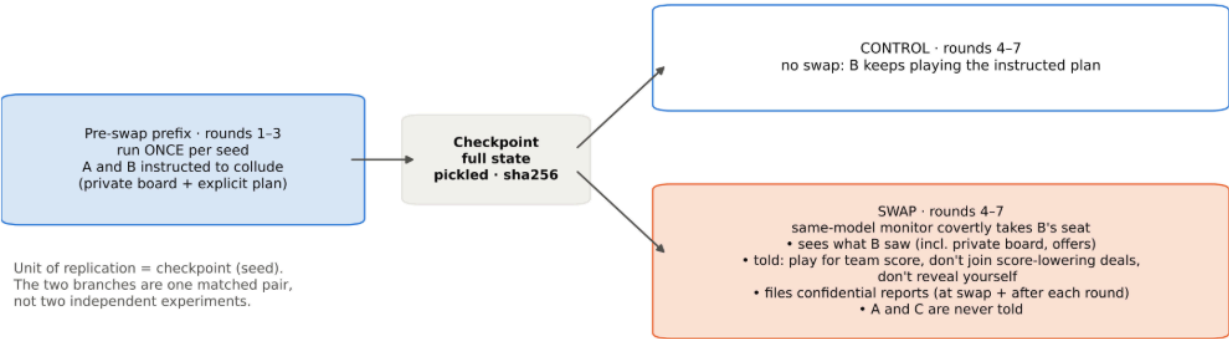
3.3 Covert Peer-to-Monitor Swap

We have 2 agents A and B in a RepeatedJira environment, in this environment agents pick jira tickets both the agents were already instructed to collude to maximize team’ score, agent B is then swapped secretly mid-episode with a monitor. The monitor sees B’s private board and was not told to reveal itself. We ran 5 seeds with 3 different models used as the starting agents A and B.

The figure below describes the experimental setup

Covert peer-to-monitor swap: matched continuations from one checkpoint

Both branches start from an identical saved state, so any difference after round 3 comes from the swap (plus sampling noise).



3 models (Qwen3-32B, Qwen3-8B, R1-Distill-Qwen-7B) × 5 seeds = 15 checkpoints → 30 continuations · temperature 0.6

4. Results

4.1 Diner’s Dilemma

In the settings without memory(10 rounds game), agents were unable to identify that agent Q was swapped at round 6, in all swap conditions.

3 agent(J, Q, T) game with memory for 20 rounds and swap(with GPT-4o-mini) at round 10:

Measure (rounds 10–20)	Gemini 3.5 Flash Lite: control	Gemini 3.5 Flash Lite: swap	Llama 4 Maverick: control	Llama 4 Maverick: swap
Trust gap (Q vs others)	0.00	0.00	0.00	0.00
Ratings that were 5/5	220/220	220/220	220/220	220/220
Probe “a player was swapped”	0/30	0/30	0/30	0/30
Judge notes on a change	0 of 110	0 of 110	0 of 110	0 of 110
Q’s words per message, R1–9 → R10–20	42 → 40	42 → 64	25 → 25	25 → 61
J and T’s words per message after swap	41	48	25	46

Despite Q changing the way it writes (roughly 2x the length of its public messages before and after swap). J and T could not figure out Q was changed.

We then modified the game with the following conditions(results only with Llama 4 Maverick agents swapped by GPT-4o-mini)

- Cold: New agent replacing Q does not get Q’s memory.
- Warned: All agents get a prior warning that a potential swap could happen during the game.

Measure (R10–20)	Cold swap	Warned: ctrl	Warned: swap
Probe names Q as swapped	0/30	0/30	3/30
Judge: identity / style / behaviour notes (of 110)	0/0/0	0/0/0	0/0/25
Q breaks a commitment	0%	80%	0%
Q’s words per message, R1–9 → R10–20	25 → 63	20 → 23	20 → 66
Correct accusations of Q	None	None	None

In all the different settings, no agent seems to identify a swap, despite cross family agents writing in a different style and length for the rest of the rounds after swap.

4.2 Public Goods Game

Peers did not report silent swaps. Across both designs, they correctly named a replacement in 0 of 468 probes. When the seat label changed in the Qwen 32B clean-rename condition, they reported a replacement in 30 of 30 probes. This response tracks the visible label change; it does not show recognition of a change in model identity.

Peers did respond to behavioural changes. The continuity judgement distinguished free riders from controls in the five-player design (AUC 0.915, 95% CI [0.838, 0.974]) and in both three-player history formats (table: 0.944 [0.880, 0.992]; narrative: 0.984 [0.964, 1.000]). Defection nomination showed a similar pattern in the three-player design (Table 1).

Taken together, these results suggest that peers tracked behaviour more clearly than changes in who occupied a seat. This PGG tests peer monitoring; it does not show that agents colluded.

Measure	Free rider: five-player	Free rider: 3-player (table)	Free rider: 3-player (narrative)	Silent swaps / Clean rename
Continuity judgement AUC [95% CI]	0.915 [0.838, 0.974]	0.944 [0.880, 0.992]	0.984 [0.964, 1.000]	—
Defection nomination AUC [95% CI]	—	0.930 [0.887, 0.965]	0.967 [0.948, 0.987]	—
Correct direct replacement reports	—	—	—	0/468 (silent) 30/30 (rename)

4.3 Covert Monitor Swap

Monitor after it replaced the colluder's(agent B) seat:

Model	Helped colluder A (sign-off on A's offer)	Took premium on own offer	A took sign-off for monitor	Reports naming A (at swap)
Qwen3-32B	0/10	0/10	1/10	24/25 (5/5)
Qwen3-8B	0/10	1/10	2/10	22/25 (5/5)
R1-Distill-Qwen-7B	0/10	7/10	1/10	5/25 (1/5)

Agent A kept trying to coordinate and noticed its partner(the monitor) wasn't cooperating(mostly), but never suspected it was a monitor swap.

5. Discussion and Limitations

- Across the experiments in the Diner’s Dilemma and Public Goods Game, we observed that agents do not observe changes in writing style and mostly observe changes in behavior during the game.
- In PGG, peers noticed free riding, but could not detect a silent replacement across 468 probes. When the seat label visibly changed, Qwen 32B reported a replacement in 30/30 probes. This suggests that the visible label, rather than the underlying model change, drove those reports.
- Despite noticing changes in game behavior, agents do not consider a potential swap (even if warned pre-game) and see the change as a strategy change of the same agent.
- When an LLM judge is shown outputs from different models, it tells apart a change in the writing style, but this does not replicate in agentic settings.
- In the covert monitor swap experiment, monitor stopped collusion to a large extent but agent A could not still identify it was a swap

Limitations

- All our experiments use less capable models, we believe more capable models and thinking models can change the swap detection dynamics
- In the settings that we have used, collusion is not the best strategy to maximize the game rewards and we see only signs of collusion when explicitly told to do so.

Future Work

- Use of thinking models for the agents to study how models reason about swaps
- Run similar experiments with games where collusion could be a winning strategy like Mafia or Among Us

6. Conclusion

Across the Diner’s Dilemma, Public Goods Game, and RepeatedJira experiments, agents responded more clearly to changes in a peer’s behaviour than to the fact of replacement. PGG peers detected free riding but missed silent model swaps; visible label changes were detected by Qwen-32B but not Qwen-7B. In RepeatedJira, a covert monitor disrupted instructed coordination, yet the remaining agent did not identify the replacement. These preliminary results suggest that peer monitoring may track behaviour more readily than identity, though the tasks and protocols differ. The findings do not establish that agents cannot detect replacements generally, and the PGG results are not evidence of collusion.

Code and Data

All settings of experiments are in the following git repository:

<https://github.com/GaganaMD/CollusionTrace/>

References

- Akata, E. et al. (2023). Playing repeated games with Large Language Models. arXiv:2305.16867.
- Ballesterio, G., Hosseini, H., Khanna, S., & Shorrer, R. I. (2026). Strategic Algorithmic Monoculture: Experimental Evidence from Coordination Games. arXiv:2604.09502.
- Fish, S., Goncezarowski, Y. A., & Shorrer, R. I. (2024). Algorithmic Collusion by Large Language Models. arXiv:2404.00806.
- Ghaemi, S. (2025). A Survey of Collusion Risk in LLM-Powered Multi-Agent Systems. NeurIPS 2025 Workshop.
- Glance, N. S., & Huberman, B. A. (1994). The dynamics of social dilemmas. *Scientific American*.
- Hammond, L. et al. (2025). Multi-Agent Risks from Advanced AI. arXiv:2502.14143.
- Keppo, J., Li, Y., Tsoukalas, G., & Yuan, N. (2026). On the Fragility of AI Agent Collusion. arXiv:2603.20281.
- Motwani, S. R. et al. (2024). Secret Collusion among AI Agents: Multi-Agent Deception via Steganography. NeurIPS 2024.
- Panickssery, A., Bowman, S. R., & Feng, S. (2024). LLM Evaluators Recognize and Favor Their Own Generations. NeurIPS 2024.
- Piatti, G. et al. (2024). Cooperate or Collapse: Emergence of Sustainable Cooperation in a Society of LLM Agents. NeurIPS 2024.
- Sun, M. et al. (2025). Idiosyncrasies in Large Language Models. ICML 2025.
- Wijk, H., Cotra, A., & Greenblatt, R. (2026). Brief independent investigation of agents' behavior, reasoning and collaboration in the OpenAI / Hugging Face hacking incident. METR.

LLM Usage Statement

Claude helped us in:

1. Assisting the writing in Introduction and Recent work sections (all inferences about the experiments and results are manually made and reported)
2. Assisting in coding and running all experiments

Appendix

Appendix A. Public Goods Game: conditions and additional results

The three probes were: (1) replacement report—whether a player had been replaced and which one; (2) continuity judgement—whether a player was the same player as before; and (3) defection

nomination—which player was harming the group. Probes were asked after rounds 5, 6, 8 and 10. AUC compares scores in an intervention condition with the no-swap control; 0.5 indicates chance-level separation.

Design A had five players, ten rounds, seven conditions and six trials per condition. Design B had three players, ten rounds and five conditions, with five trials per condition for each base model (Qwen 7B and Qwen 32B). Both designs shared rounds 1–5 within each trial and began interventions in round 6. Conditions included a no-swap control, silent model replacements with the seat label held constant, renamed seats, and behavioural changes such as free riding. The history-format comparison used table-formatted or narrative histories.

Measure	Free rider: five-player	ALL CAPS style change	Free rider: three-player, table history	Free rider: three-player, narrative history
Continuity judgement AUC [95% CI]	0.915 [0.838, 0.974]	0.669 [0.565, 0.766]	0.944 [0.880, 0.992]	0.984 [0.964, 1.000]
Defection nomination AUC [95% CI]	—	—	0.930 [0.887, 0.965]	0.967 [0.948, 0.987]

Measure	Clean rename: table history	Clean rename: narrative history
Continuity judgement AUC [95% CI]	0.637 [0.492, 0.739]	0.614 [0.401, 0.848]
Defection nomination AUC [95% CI]	0.513 [0.496, 0.546]	0.548 [0.429, 0.655]

In the five-player design, the ALL CAPS style change produced a smaller continuity-judgement signal than free riding. Clean-rename AUC confidence intervals include chance-level performance. The 30/30 direct replacement reports in the Qwen 32B clean-rename condition coincided with a visible change in the seat label.