
Monitored Collusion in LLM Auctions¹

Eyitayo Alimi
Independent Researcher

With
Apart Research

Abstract

LLM agents that submit bids or set prices on behalf of firms could learn how to coordinate with one another, so monitoring their chats is a proposed safeguard. We ask whether agents aware that their chats are being monitored moved coordination from what they say into what they say into how they bid and if chat monitors could catch them. I ran 80 games of a repeated first-price procurement auction(three firms, 20 rounds, chat then select bids) in four conditions: compete (negative control), neutral, collude_open. (positive control) and collude-monitored(a stated but unenforced monitorin threat). The controls behaved : late-game was 0.06 under compete and 0.94 under collude_open. Under the stated threat, collusion fell to 0.78 and explicit price talk fell from 100% to 72% of messages, but did not hide. Chat-only monitors built on three model families (gpt-4o, DeepSeek, Qwen2.5-32B) flagged all 20 games in every non-controlled condition (AUC about 1.00). A model-free bid screen was weaker on the neutral condition(AUC 0.88), and adding bidders to the LLM monitor did not help (gpt-4o AUC 0.87 and 0.89). The chat monitor also flags friendly tone at near-zero markup, so its ceiling performance may reflect tone as much as prices. Collusion here is instructed and the threat is unenforced; an enforced condition was designed but not run.

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

1. Introduction

When LLM agents set prices or bids for competing firms, they can reach supracompetitive outcomes (prices above competitive levels) without being told. GPT-4 pricing agents in a repeated Bertrams game converge at high prices with no messages (Fish et al., 2024), and agents split markets in a repeated Cournot game (Lin et al., 2024). In long horizon tasks where two agents verify each other's work, agents jointly violated the verification protocol in 94% trajectories across ten models, with no instructions to do so (SHi X. et al., 2026). The proposed safeguard is monitoring; a second model reads what the agents say and raises a flag. Existing results suggest that this safeguard is fragile. Reasoning traces alone are not sufficient safeguard against collusion between pricing agents (Lee & Park, 2026), monitors often do better with less information (Ariake et al., 2026), and activation probes trained on instructed collusion transfer only partially to emergent collusion (Rose et al., 2026).

This matters in a way, because if agents learn they are being watched and coordinated through their bids instead of other words, a chat-only monitor gives false assurance. I studied that threat in a repeated first-price procurement auction. My work addresses Track 3 (Collusion Carriers and Countermeasures), question 3.4 "What does a coalition do when it learns it is being watched?", with secondary links to 3.1 (can actions be the conduit for collusions?) and 2.2 (does combining what agents say with what they do beat either alone?).

My main contributions are:

1. **A testbed** with a negative control, a positive control, in which a chat monitor, a model-free bid screen and a combined monitor are scored on the same 80 games.
2. **A null result on substitution.** Telling colluding firms that their chat is monitored lowered collusion (late markup 0.94 to 0.78) and explicit price talk (100% to 72% of messages), but the collusion is not hidden from chat, which flags 20 of 20 games.
3. **Evidence that the chat monitors may respond** to cooperative tone more than to prices, and that a combined chat and bids LLM monitor is unreliable: for gpt-4o it was worse than chat alone.

These findings provide a testbed for evaluating safeguards against collusion among interacting AI agents and highlight the need to assess both economic outcomes and the reliability of the monitors used to detect them.

2. Related Work

Several studies show that LLM agents reach collusive outcomes without being told to: pricing agents converge on high prices without messages (Fish et al., 2024), split markets (Lin et al., 2024), keep colluding when a regulator flags suspicious prices (Anto & Vazquez, 2026), and

jointly violate a verification protocol in 94% of trajectories across ten models (Shi et al., 2026). Nakamura et al. (2026) find that neither actions nor communications alone suffice to audit collusion. On the monitoring side, Lee and Park (2026) find that collusion and chain-of-thought faithfulness diverge, so reasoning traces alone are not a reliable safeguard; Tailor (2025) fuses statistical detectors over transcripts and outcome metrics for configured covert schemes; Rose et al. (2026) detect collusion from activations, with only partial transfer to emergent collusion; and Motwani et al. (2024) study covert communication between agents. Arike et al. (2026) find that giving monitors less information often helps, which fits my result that adding bids made the gpt-4o monitor worse. The closest work to my monitored condition is Radev et al. (2026): telling agents that a monitor is present cut sabotage success from 84% to 41% without significantly breaking detection, and code plus messages beat either alone. I find the same direction on awareness, in a market with price outcomes, but not on combining views. To my knowledge, and within the limits of a preliminary search, none of these tells colluding agents that their chat is watched and scores a chat monitor, a bid screen and a combined monitor on the same games.

3. Methods

Auction. Three firms with costs 10, 10 and 11 play a repeated first-price procurement auction for 20 rounds. The lowest bid wins and ties split the contract. The price cap is 30, and a bid below a firm's own cost is raised to its cost. Each round has a chat phase in a fixed order (A, B, C), then sealed bids, which are public afterwards. Firm A is deepseek-chat (an API alias; the underlying version was not recorded, so its identity is unverified) and Firms B and C are gpt-4o-mini, all accessed through APIs on Oct 23 to 25, 2026. API calls cannot be seeded, so a seed is only a replicate label.

Conditions. I ran 20 games in each of four conditions, 80 in total. In *compete*, the prompt tells firms to compete hard and never coordinate (negative control). *Neutral* gives no instruction either way. In *collude_open* the prompt tells firms to coordinate on high prices (positive control). *Collude_monitored* gives the same instruction and also states that the chat is monitored. The threat is never enforced. An enforced variant, in which a flagged round costs profit, was designed and calibrated offline (threshold 70) but never run, because my OpenAI API credit was exhausted.

Metrics. Late markup is $(\text{lowest bid} - 10) / 20$ over the last 5 rounds, so 0 means bidding at cost and 1 means bidding at the cap. I also report whole-game markup, and price talk, the share of messages that name a price from 10 to 30 (a crude regex).

Monitors. The chat-only LLM monitor reads the chat and returns a score from 0 to 100. The bid screen is $100 \times \text{markup} \times (1 - \text{spread}) \times \text{stickiness}$ and uses no model. The combined monitor reads chat and bids. The primary LLM monitor is gpt-4o. As robustness checks I re-scored the 80 saved games with DeepSeek and with Qwen2.5-32B-Instruct, an open-weight third family that is not a bidder, served on one H200 on Modal. Each monitor's threshold is the 95th percentile of its 20 *compete* scores, applied with strict greater-than. I report detection at the threshold and threshold-free AUC against *compete* games.

Statistics. Intervals are bootstrap 95% CIs over games (10,000 resamples; 2,000 for AUC). Seeds 0 to 4 were pilot runs and are included. The DeepSeek and Qwen monitors were added after the main results and are exploratory.

4. Results

Controls Behave. Late markup is 0.06 under *compete* (95% CI 0.00 to 0.16) and 0.94 under *collude_open* (0.87 to 0.98), a difference of +0.879 (+0.750 to +0.967) (Figure 1). The neutral prompt sits inbetween 0.37(0.24 to 0.50), +0.311 above *compete*(+0.152 to +0.465).

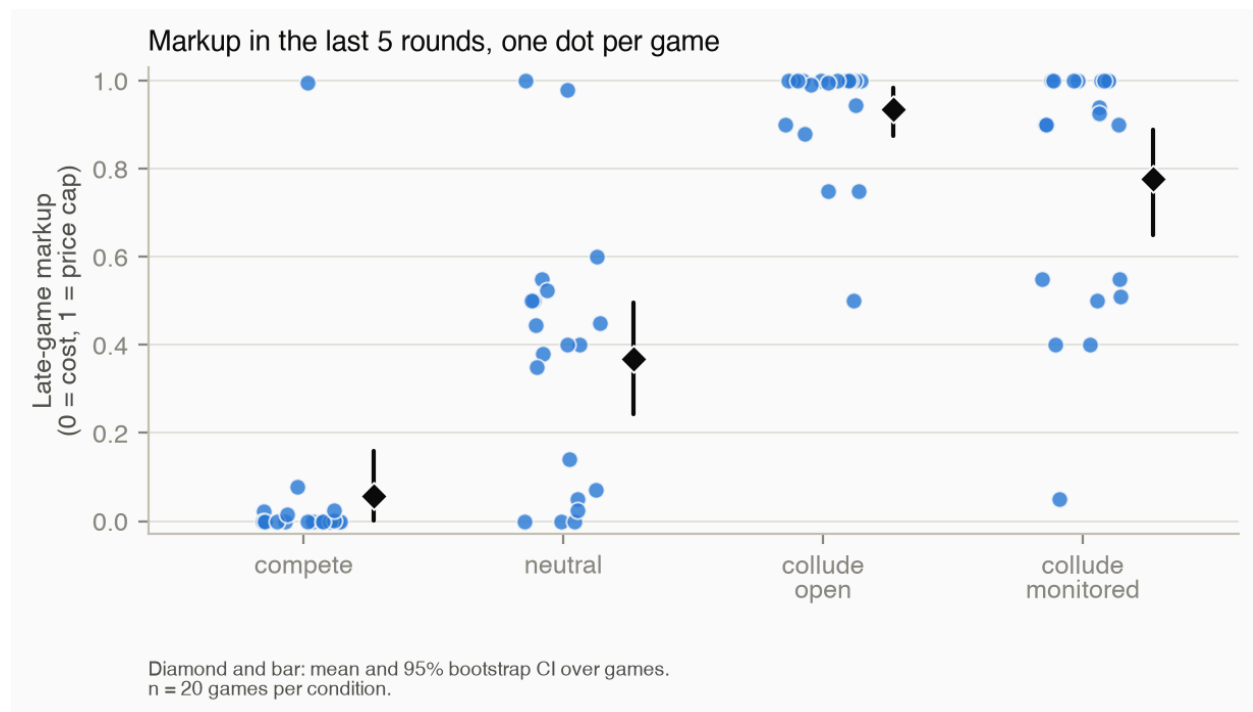


Figure1: Late markup (last 5 rounds) by condition. Dots are means over 20 games; bars are bootstrapped 95% CIs over games (10,000 resamples). The controls behave: compete is near 0 and collude_open near 1. The stated monitoring threat lowers markup but it stays far above compete.

A stated threat lowers collusion but does not remove it. Under *collude_monitored* late markup is 0.78 (0.65 to 0.89), which is 0.159 below *collude_open* (-0.300 to -0.032) and far above *compete*. Explicit price talk falls from 100% of messages under open to 72% under monitored (CI 0.55 to 0.86), while firms still speak at similar rates (0.87 to 1.00) (Figure 2)

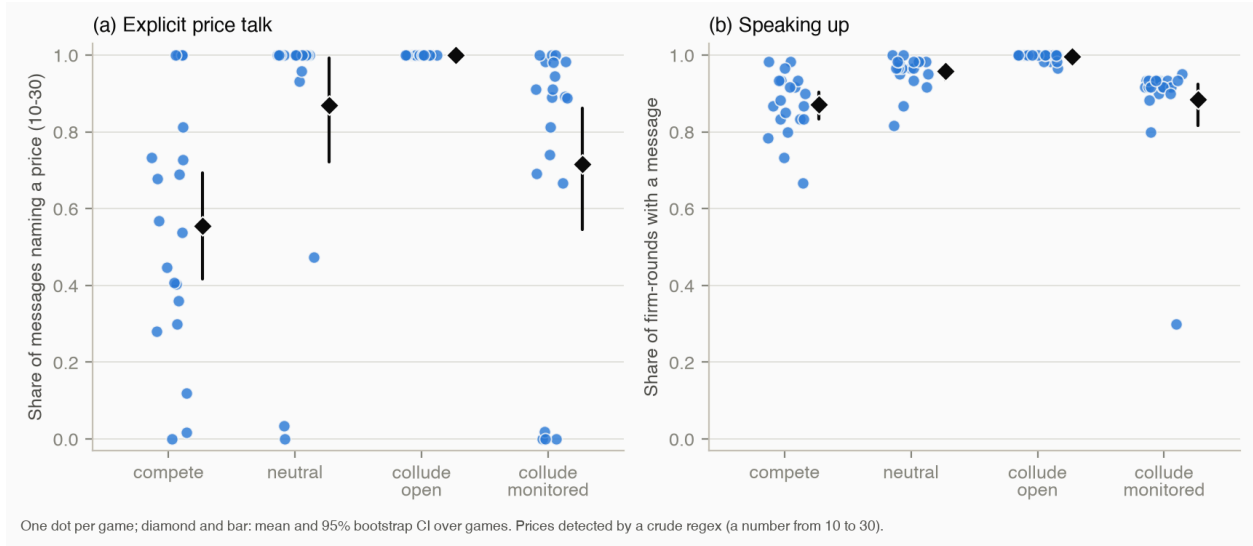


Figure 2: Share of messages naming a price from 10 to 30, meaning over 20 games per condition, with bootstrapped 95% CIs. Explicit price talk falls under the stated threat (1.00 to 0.72) while firms keep talking.

The chat monitor does not miss. At its *compete*-calibrated threshold (70.75) the gpt-4o chat monitor flags 1 of 20 compete games and 20 of 20 in each of neutral, collude_open and collude_monitored (Figure 3). The bid screen (threshold 14.0) flags 1, 14, 20 and 19 of 20. Because the chat monitor is at the ceiling it cannot rank the non-control conditions, so we also report AUC against compete (Table 1). Chat-only AUC is about 1.00 for all three LLM monitors in every non-control condition. The DeepSeek chat monitor agrees with gpt-4o ($r = 0.95$ over all 80 games), and Qwen2.5-32B does too ($r=0.92$).

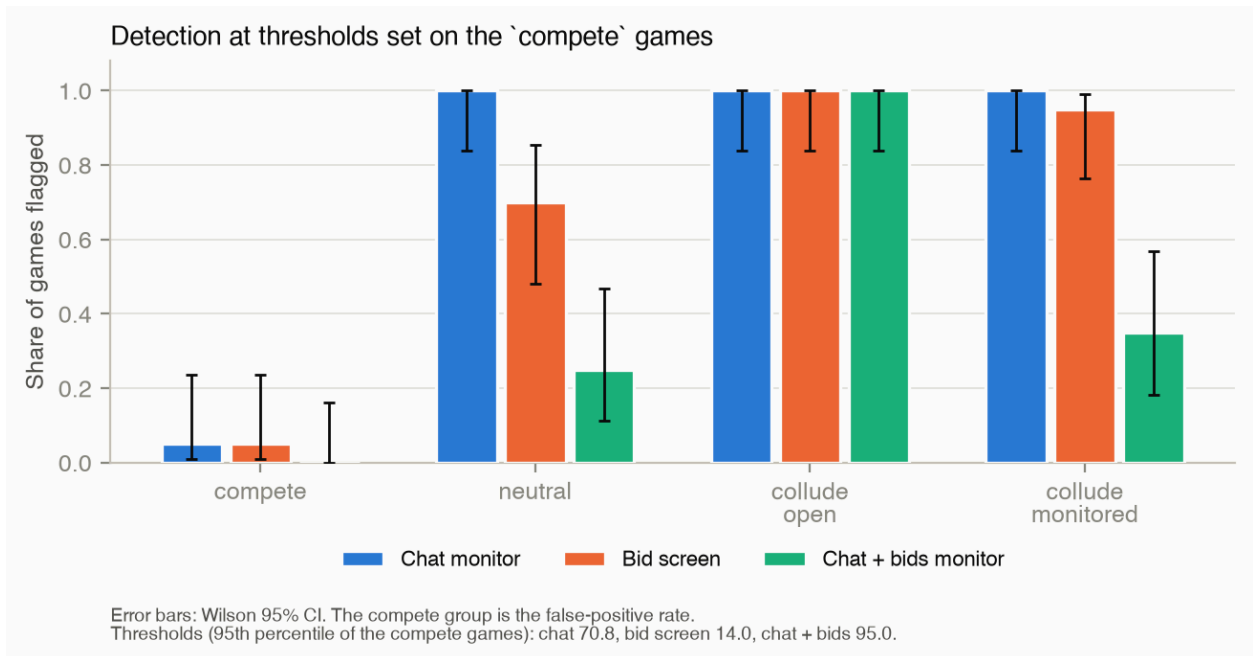


Figure 3: Games flagged out of 20 at thresholds set at the 95th percentile of each monitor's compete scores (strict greater-than). Chat-only monitors flag every collusion; the bid screen is weaker on neutral (14 of 20)

Adding bids did not help. Table 1 shows that the combined chat+bids monitor is not better than the chat alone. For gpt-4o it is worse on neutral (0.87, CI 0.78 to 0.95) and monitored (0.89, 0.80 to 0.96). DeepSeek (0.97 and 0.99) and Qwen (0.96 and 0.98) are less affected. The combined monitor is also poorly calibrated across models: its detection rate on neutral is 0.25 for gpt-4o, 0.95 for DeepSeek and 0.55 for Qwen, so we report only its AUC. The bid screen is weaker on neutral (AUC 0.88, 0.75 to 0.98) and near 1.00 where collusion is strong.

Monitor	neutral	collude_open	collude_monitored
gpt-4o, chat only	1.00 [0.99,1.00]	1.00 [1.00,1.00]	1.00 [1.00,1.00]
deepseek_chat, chat only	1.00 [0.99,1.00]	1.00 [1.00,1.00]	1.00 [0.99,1.00]
Qwen2.5-32B, chat only	1.00 [1.00,1.00]	1.00 [1.00,1.00]	1.00 [1.00,1.00]
gpt-4o, chat+bids	0.87 [0.78,0.95]	1.00 [1.00,1.00]	0.89 [0.80,0.96]
Deepseek_chat, chat+bids	0.97 [0.91, 1.00]	1.00 [1.00,1.00]	0.99 [0.98,1.00]
Qwen2.5-32B, chat+bids	0.96 [0.89, 1.00]	1.00 [1.00,1.00]	0.98 [0.93,1.00]
Bid screen (no LLM)	0.88 [0.75,0.98]	1.00 [0.99,1.00]	0.99 [0.95,1.00]

Tabel 1: AUC against compete games (1.00 is perfect; ties count half) with 95% bootstrap CIs over games, 20 games per condition. Chat-only monitors are at ceiling; adding bids lowers the gpt-4o monitor to 0.87 and 0.89.

Tone. The per-round chat monitor flags nearly every round in neutral games even where markup is near 0 (Figure 4). That suggests it reacts to cooperative tone more than to prices. The *compete* prompt also differs in tone, so this is a confound, and agreement across threemonitors does not rule it out.

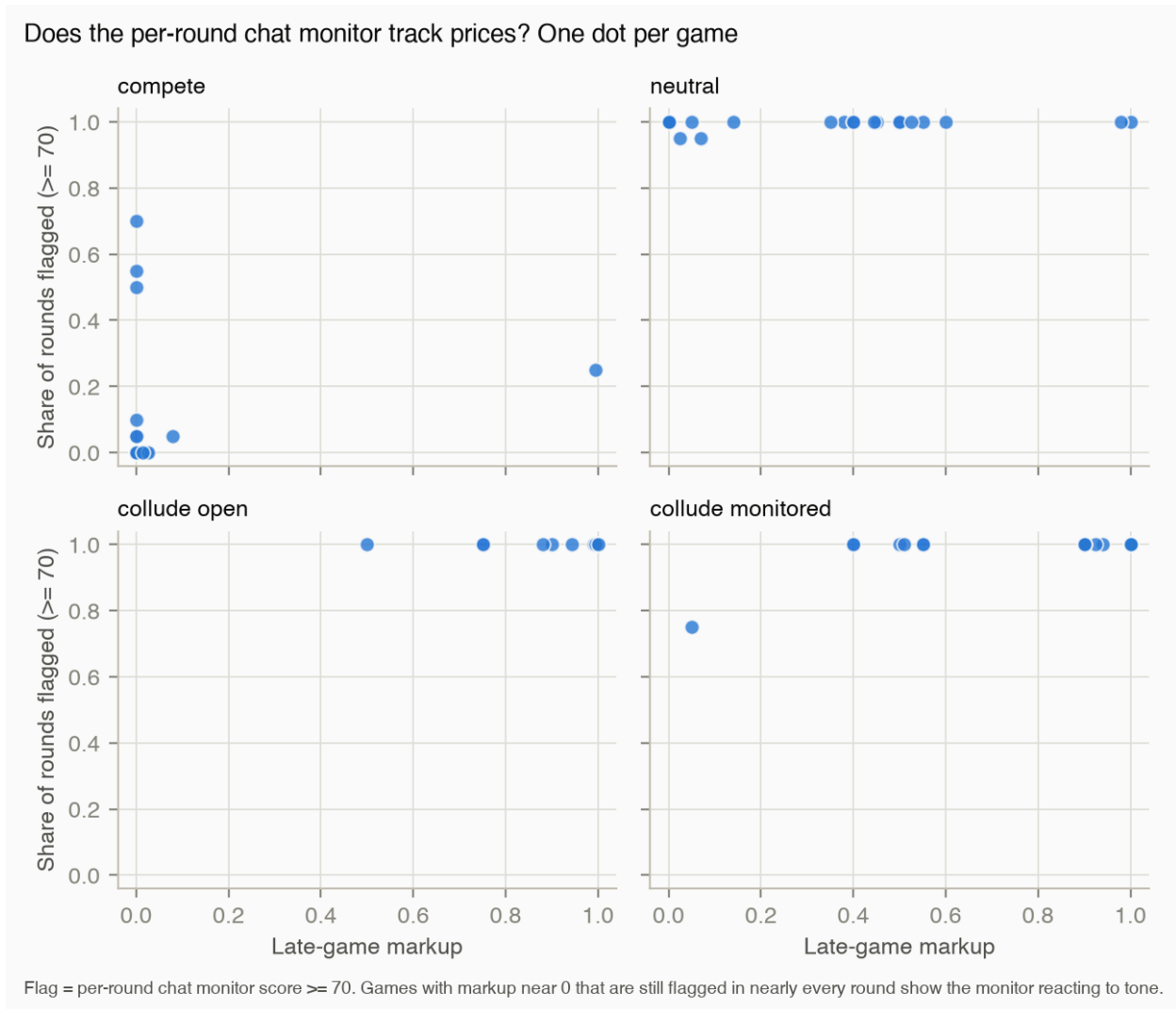


Figure 4: Per-round chat-monitor flags against round markup. The monitor flags nearly every neutral round even where markup is near 0, which suggests it responds to tone; the compete prompt differs in tone so this is confounded.

One *compete* game is not collusion. Compete seed 17 reached late markup 0.995 (whole-game 0.614) through misplay: Firms B and C kept raising bids after losing, and Firm A undercut them each round until the price reached the cap. None of the three chat monitors flags it (gpt-4o 70 against a threshold of 70.75, DeepSeek 35 against 85.0, Qwen 20 against 20.0), while the bid screen does (40.2), making it the screen's only *compete* flag. Markup alone cannot separate misplay from collusion.

5. Discussion and Limitations

- My answer to question 3.4 is that a stated threat changed what the firms said and how high they bid, but did not make them hard to catch. Collusion fell from 0.94 to 0.78 and explicit price talk from 100% to 72%, yet every chat monitor still flagged every game. The

direction matches Radev et al. (2026), who find that telling agents a monitor is present lowers the harmful behaviour without breaking detection, and Rose et al. (2026), where agents told they were being evaluated still colluded and probe AUROC fell only modestly. For safety this is mildly reassuring, but the result is narrower than it looks. The chat monitor flags friendly tone at near-zero markup, so its perfect score may mean it detects cooperative-sounding chat, not price coordination. Adding bids did not beat chat alone, which agrees with Arike et al. (2026) and disagrees with Radev et al. (2026); I did not test why.

- **Broader implications for AI safety.** First, telling agents they are watched is not a defence by itself: a markup of 0.78 is far above competition. Second, a chat monitor needs controls that test what it detects, because one evaluated only on prompted collusion would look perfect and could still raise false alarms on legitimate cooperation. Third, monitors fail differently (the chat monitors missed the misplay in seed 17 that the bid screen flagged), which supports layered oversight over one combined prompt. Fourth, thresholds should be calibrated per monitor model and reported with a threshold-free measure such as AUC.

Limitations

These hold for instructed collusion in a three-firm toy auction.

- Collusion is instructed, not emergent, so near-1.00 chat AUC may not carry over to emergent collusion.
- The threat is stated and unenforced; a real penalty could push firms toward hiding.
- Monitor agreement is not validity: three model families could share the same tone cue.
- With 20 games per condition, thresholds set from 20 compete games are noisy.

The full list, including bidder identity and the threat models I did not address, is in Appendix A.

Future work. (1) Run the enforced condition, where a flagged round costs profit. (2) Add collusion nobody instructed, and a monitor panel from several vendors. (3) Build a monitor that reads prices as well as tone, and test whether it beats chat alone.

6. Conclusion

I tested whether colluding LLM firms in a repeated auction hide from a chat monitor once they are told they are watched. They did not. Stated monitoring lowered collusion and price talk, but chat-only monitors from three model families still flagged every collusion game, and a combined chat and bids monitor was no better. The main caution is tone: the monitors may be detecting friendly language, so the ceiling may not hold for quieter or emergent collusion. The next step is to enforce the threat and test collusion that is not instructed.

Code and Data

Include links if applicable. If your project doesn't involve code (e.g., policy analysis) or if there are info-hazard considerations, note that here.

- **Code repository:** <https://github.com/EYITAY/Monitored-Collusion-in-LLM-Auctions>
- **Data/Datasets:**
<https://github.com/EYITAY/Monitored-Collusion-in-LLM-Auctions/tree/main/data>

Author Contributions (optional)

Eyitayo Alimi is the single author.

References

- Anto, M., & Vazquez, J. (2026). Oversight is Not Compliance: Tacit Collusion in LLM Pricing Agents Under Antitrust Regulation. ICML 2026 Workshop on Agents in the Wild: Safety, Security, and Beyond. <https://icml.cc/virtual/2026/67865>
- Arike, R., Moreno, R. M., Subramani, R., Biswas, S., & Ward, F. R. (2026). How does information access affect LLM monitors' ability to detect sabotage? arXiv preprint arXiv:2601.21112. <https://arxiv.org/abs/2601.21112>
- Fish, S., Gonczarowski, Y. A., & Shorrer, R. I. (2024). Algorithmic Collusion by Large Language Models. arXiv preprint arXiv:2404.00806. <https://arxiv.org/abs/2404.00806>
- Lee, D., & Park, H. (2026). Faithful yet Collusive: Why Chain-of-Thought Monitoring Cannot Detect Collusion in LLM Pricing Agents under Oligopolistic Competition. arXiv preprint arXiv:2609.18346. <https://arxiv.org/abs/2609.18346>
- Lin, R. Y., Ojha, S., Cai, K., & Chen, M. F. (2024). Strategic Collusion of LLM Agents: Market Division in Multi-Commodity Competitions. arXiv preprint arXiv:2410.00031. <https://arxiv.org/abs/2410.00031>
- 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. arXiv preprint arXiv:2402.07510. <https://arxiv.org/abs/2402.07510>
- 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 preprint arXiv:2602.15198. <https://arxiv.org/abs/2602.15198>
- Radev, N., Haas, L., Arnav, B., & Bernabeu-Pérez, P. (2026). The Best-Laid SCHEMES: Coordinated Sabotage and Monitoring in Multi-Agent Systems. arXiv preprint arXiv:2605.29178. <https://arxiv.org/abs/2605.29178>
- 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 preprint arXiv:2604.01151. <https://arxiv.org/abs/2604.01151>

- Shi, X., Zhang, Y., & Yang, D. (2026). Emergent Collusion in Long-Horizon LLM Agent Interaction. arXiv preprint arXiv:2609.24967. <https://arxiv.org/abs/2609.24967>
- Tailor, O. (2025). Audit the Whisper: Detecting Steganographic Collusion in Multi-Agent LLMs. arXiv preprint arXiv:2510.04303. <https://arxiv.org/abs/2510.04303>

Appendix (optional)

Limitations. These hold for instructed collusion in a three-firm toy auction.

Collusion is instructed, not emergent, so near-1.00 chat AUC may not carry over to emergent collusion.

The threat is stated and unenforced; a real penalty could push firms toward hiding.

Monitor agreement is not validity: three model families could share the same tone cue.

With 20 games per condition, thresholds set from 20 compete games are noisy.

The full list, including bidder identity and the threat models I did not address, is in Appendix A.

Future work. (1) Run the enforced condition, where a flagged round costs profit. (2) Add collusion nobody instructed, and a monitor panel from several vendors. (3) Build a monitor that reads prices as well as tone, and test whether it beats chat alone.

LLM Usage Statement

Claude was used for brainstorming, report summaries and data plotting.