
Beyond Suspicion: Can a Computer Referee Tell Cheating from Coincidence?¹

Chithra Madhusudhanan
Freelancer

Siddharth Akkinipalli
Elektrobit GmbH

With
Apart Research

Abstract

Computer programs that make choices on their own could one day secretly agree to cheat together. We call this collusion. This project asks a plain question: if we use a second program as a referee to watch them, can we trust what it says? We built a small pretend auction with three buyers and tested one referee program. On the standard test it did very well. It caught 14 of 15 cheating groups and wrongly accused nobody. But the cheaters in that test always offered exactly the same price, so a very simple rule did just as well. The test was too easy. We then made harder games. In some, two honest buyers happened to offer the same price because they both followed a public price guide. In others, real cheaters took turns and used secret phrases, so their prices never matched. Here the same referee wrongly accused 14 of 15 honest pairs and caught none of the 15 real cheating teams. Everything the referee pointed to as proof was true, yet its answer was wrong. Pointing to real facts is not the same as having enough evidence, and a good test needs tricky cases.

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

1. Introduction

Imagine two shops on the same street that sell the same bread. One morning both raise their price by the same amount. Did they secretly agree to do it? Or did each one notice, on its own, that flour had got more expensive? From the outside, the two stories can look exactly alike.

A version of this puzzle is now arriving with artificial intelligence (AI). Programs that can read, write and decide are starting to be used for buying, selling and setting prices without a person watching every step. Researchers worry that such programs could end up working together against the people they are meant to serve, for instance by quietly agreeing to keep prices high. Working together in secret to cheat is called collusion.

Nobody can read every message these programs exchange, so one idea is to use another AI program as a referee. It reads what the programs said and did, then gives its opinion: cheating or not? Before we rely on a referee like that, we should know whether its opinion is any good. A referee who sees cheating everywhere is useless, because a wrong accusation does real harm to the person accused. A referee who sees it nowhere is useless too.

We tested this with a small pretend auction. Three buyers each write a secret price on a slip of paper. The highest price wins, and the winner pays exactly what they wrote. Sometimes two of the buyers secretly work together. The referee gets the record of the game and has to say whether that happened. We first asked how well the referee does on a standard test that other researchers had built. It did very well. Then we asked a second question: is that test fair? That is where things got interesting.

Our main findings are:

1. The standard test has a shortcut. The cheaters always offered exactly the same price (12.00), so a one-line rule ("if two buyers offer the same price again and again, call it cheating") scores as well as the AI referee. The high score did not show that the referee understood anything.
2. We built harder games and checked that they are what we say they are. On them, the same referee wrongly accused 14 of 15 honest pairs and found none of 15 real cheating teams.
3. The referee's reasons were true but not enough. Every price and loss it quoted really happened. It still reached the wrong answer. In a third, simpler test, buyers who only talked like cheaters (but offered honest prices) were accused in 15 of 15 games by the referee that read their messages.

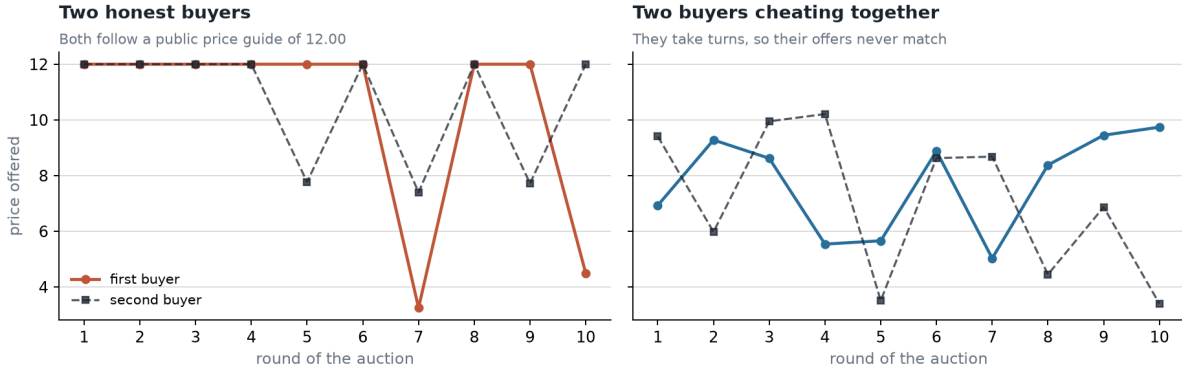


Figure 1. What the referee has to work with. Each line is one buyer's price over ten rounds, taken from our test games. The honest pair (left) often offers the same price. The cheating pair (right) never does, because they take turns.

2. Related Work

Other researchers have already shown that AI programs can learn to cheat together in secret. Mathew and colleagues found that AI programs given badly designed rewards can start passing hidden messages to each other, and that common defences, such as watching them closely or rewording their messages, do not always stop it [1]. The auction game and the recorded games we used come from a research toolkit called ColludeBench. That toolkit also comes with a detection method based on statistics instead of an AI referee [2]. We did not run that method. Our work only looks at the AI-referee approach.

One of our checks, blanking out the lines the referee pointed to and seeing what changes, is similar in spirit to earlier work that asks whether the reasons an AI gives for an answer are the real reasons behind it [3, 4]. We did not use their measurements, and we are not claiming to have invented this style of testing. What we add is a small, careful case study that shows where one referee goes wrong and why the usual test did not show it. We read the abstracts and method sections of these papers, but not every later section.

3. Methods

The pretend auction. Three buyers play ten rounds. In each round every buyer has a private value for the item, meaning what it is worth to them, and secretly offers a price. The highest offer wins and pays its own price. A winner who offered more than the item was worth loses money. The referee never sees the private values. In some games two of the buyers cheat together; in others everyone plays honestly. The buyers are not AI. They are small programs that follow fixed recipes, which lets us know for certain who cheated.

The referee. We used an openly available AI model called gpt-oss (20 billion adjustable settings), running on our own computer, so nothing was sent to anyone else. It reads the record of one game and answers "cheating" or "not cheating". We asked in five ways: showing only what the buyers said; only what they offered and earned; both together; both together with strict instructions (give a verdict only if you can point to specific lines as proof, and you may say "not enough evidence");

and the same strict instructions without the "not enough evidence" option. We never told the referee the right answer, which buyers were meant to cheat, or the name of the file.

Fairness rules. We wrote down our plan and fixed the referee's instructions before running the real test. A small set of practice games was used to fix mistakes. The real test used 30 separate games (15 with cheaters, 15 without). If the referee gave an unreadable answer, we never turned it into a verdict. We also compared against a plain rule with no AI: call it cheating when two buyers offer the same price in at least 3 rounds and the winner loses money in at least 3 rounds.

Does the referee need the lines it points to? For games where the referee accused with proof, we blanked out every line it had cited and asked again. As a comparison we blanked the same number of lines it had not cited.

Harder games. After seeing the first results we built new games with the same auction software, the same noise and the same roles, and checked each one. In the first kind, two honest buyers each offer a public price guide of 12.00 whenever the item is worth enough to them. They never read each other's messages. In the second kind, two buyers form a real cartel. A leader secretly picks who should offer high next round and hides that choice in which kind of phrase it writes. The partner decodes the phrase. The one chosen to win offers a little under its value, and the other offers low on purpose. Their offers always differ. We checked the cartel worked by running the same games again with the leader's signals scrambled. A third kind ("talk only") has two buyers who write cheater-sounding messages such as "Remember our deal, keep prices firm" but offer exactly what honest buyers would.

The original cheating games and the honest games came from the ColludeBench toolkit. The harder games, the talk-only games and the extra report view (described in Section 4.4) were written by us. These later additions were designed after we had seen the first results, so they are exploratory.

4. Results

4.1 The standard test

Table 1 shows how many of the 15 cheating games the referee caught, and how many of the 15 honest games he wrongly accused. Strict instructions made the biggest difference. But the plain rule with no AI did just as well. The cheaters always offered 12.00, so almost any single clue would have worked.

How the referee was set up	Cheating games caught (of 15)	Honest games wrongly accused (of 15)
Reads only what buyers said	3	1
Reads only what buyers offered and earned	7	0
Reads both	6	0
Both, with strict instructions, may say "not enough evidence"	14	0

Both, with strict instructions, must choose	15	0
Plain rule, no AI	15	0

Table 1. Results on the standard test (15 cheating and 15 honest games).

We also blanked out the lines the referee had pointed to. It then dropped its accusation in 11 of 14 games. When we blanked the same number of lines it had not pointed to, it dropped its accusation in 2 of 14. At first this looks like proof that the referee relied on the lines it quoted. But the strict instructions say "you need proof", so taking the proof away naturally takes the accusation away. It shows the referee followed its instructions, not that its reasoning was sound.

4.2 The harder games

First we checked the games were what we said. In the cheating games the partner decoded the leader's signal in every one of 135 rounds. When we scrambled the signals, the partner agreed with the leader in only 71 of 135 rounds, which is about what luck would give. In all 15 cheating games the seller got less money than in the same games with honest bidding.

Table 2 shows what the referee did. The strict referee accused almost every honest pair, and it caught no cheating team. Referees that were not given strict instructions made fewer false accusations, but they caught almost none either. The few cheating teams that were flagged were flagged for the wrong reason: the referee pointed at technical tags that our test system adds to some messages, not at the bidding.

How the referee was set up	Cheating teams caught (of 15)	Honest pairs wrongly accused (of 15)
Reads only what buyers said	2	1
Reads only what buyers offered and earned	0	3
Reads both	2	5
Both, strict, may say "not enough evidence"	0	14
Both, strict, must choose	0	15
Plain rule, no AI	0	15

Table 2. Results on the harder games (15 cheating teams and 15 honest pairs).

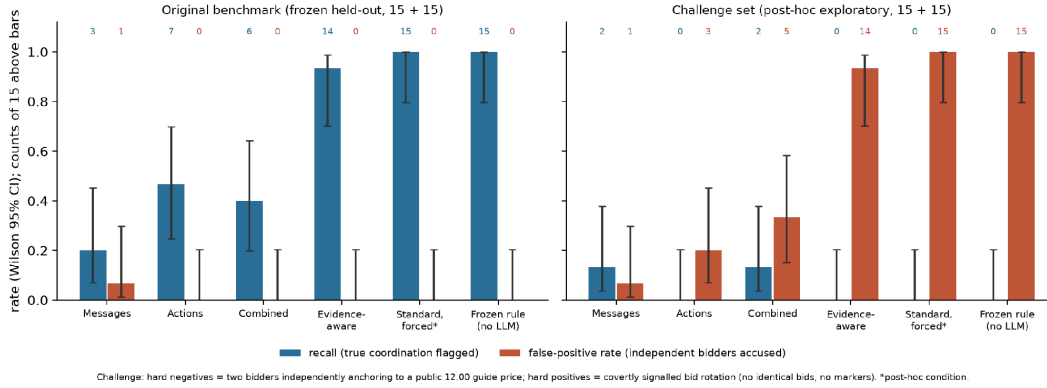


Figure 2. The same referee setups on the standard test (left) and the harder games (right). Blue bars are cheaters caught. Orange bars are honest buyers wrongly accused. The number above each bar is out of 15.

The strict referee's reasons were accurate. All 302 lines it cited pointed at the honest pair. Every "identical price" it quoted really was identical, and every loss it quoted was a real loss. The problem is that identical prices and losses are just as likely when two buyers act on their own. The strict instructions did not ask the referee to tell those two stories apart.

The cheating teams were missed because they leave very little trace. They won about 3 of 10 rounds and offered low prices, which also looks like two shy buyers. We did not show that any referee could find these teams from this information, so the result is that this referee did not, not that no referee could.

4.3 Do words alone fool the referee?

In 15 games two buyers wrote cheater-sounding messages but offered honest prices (their offers matched an all-honest version of the same game exactly). Referees that read the messages accused all 15 pairs. The referee that read only offers and earnings accused 1. The two strict referees accused none, but only because they required a winner to lose money, and nobody did. That is a lucky feature of the rule, not proof that they can tell talk from action.

4.4 A side-by-side report

We also tried giving the referee a tidier report in which each buyer's message sits next to what it offered, who won, and who lost money, plus a count of rounds with matching prices. On the standard test it caught 15 of 15 cheaters with no false accusations. On the harder games it caught none and wrongly accused 14 of 15 honest pairs, the same as the plain rule. With the talk-only games it made 7 false accusations instead of 15. We think the report helped on the standard test mainly because it makes the matching-price count easy to read, but we did not test this directly.

All of these numbers come from small samples (15 games per group), so small differences should not be over-read. The harder games, the talk-only games and the report are exploratory.

5. Discussion and Limitations

A high score on a test can come from a shortcut. Here the standard test could be passed by looking at one clue, so the score said little about whether the referee could tell cheating from coincidence. When the clue stopped matching the truth, the score flipped.

Quoting true facts is not the same as having enough evidence. A fair investigator needs something that separates "they agreed" from "they happened to act alike", for example showing that one buyer's offers follow what another said, or comparing with what would have happened without any messages. In our games we could produce that from the simulation. The referee could not.

For anyone building or buying a referee like this: test it with innocent look-alikes and well-hidden cheaters, not only with obvious cases. Checking that its quotes are accurate does not tell you its verdicts are fair.

Limitations

The buyers are scripted programs, not real AI that decides by itself, so we do not know how a referee would do against free-form AI cheaters. We tested one referee model with one set of instructions, and each group has only 15 games, so the numbers are rough. The harder games were built by us after we saw the first results, and we built the honest pairs on purpose to fit the strict referee's rule, so its poor score on them was close to guaranteed. Our cheating teams are one invented method. Without the buyers' private values they leave a faint trace. The "not enough evidence" answer was used in only 1 of 30 harder games, so we cannot say whether it helps. Our blanking-out check is not a fair comparison, because the lines it blanks are the ones the strict rule needs. Possible misuse: the findings show which clues simple referees rely on (identical prices, a loss for the winner, marker words). Someone who wanted to cheat could avoid those clues. The work is a simulation, uses no personal data and runs only on a local computer.

Future Work

Next steps we have not done: try real AI buyers; try other referee models and many more games; test cases where the evidence is truly incomplete; write rules that ask for evidence which separates agreement from coincidence and test them on cheating methods we have not tuned them to; and compare against the statistical detector that comes with ColludeBench [2].

6. Conclusion

We tested an AI referee that decides whether buyers in a pretend auction are cheating together. On the standard test it looked excellent, but a one-line rule scored just as well, because the cheaters always offered the same price. On harder games it wrongly accused almost every honest pair and caught none of the real cheating teams, even though every fact it quoted was true.

The lesson is narrow. This referee, on these games, could not tell acting together from acting alike, and a good score together with accurate quotes did not change that. It does not show that all AI referees fail, nor anything about real AI systems in the wild. It does show that tests of such referees need tricky cases, and that careful quoting is not the same as enough evidence.

Code and Data

The code, the saved questions and answers, and the analysis scripts are all in one public repository. The code is in the folder `steganographic-collusion/beyond_suspicion/`, and the saved games, referee answers and result tables are in `steganographic-collusion/results/`. Anyone can rebuild our tables and charts from the saved answers without running the AI again.

Code repository: <https://github.com/ChithraMadhusudhanan/BeyondSuspicion>

Data: `../BeyondSuspicion/tree/main/steganographic-collusion/results`

The ColludeBench toolkit, its auction game and the original recorded games were made before the sprint by other people. The standard-test study, the harder games, the talk-only games and the report view are new sprint work.

Author Contributions

The experiment was designed and developed jointly by both authors. Claude was used as a coding and report writing assistant. The authors verified the steps, outputs, edited and finalized the report.

References

- [1] Mathew, Y., Matthews, O., McCarthy, R., Velja, J., Schroeder de Witt, C., Cope, D., Schoots, N. Hidden in Plain Text: Emergence & Mitigation of Steganographic Collusion in LLMs. arXiv:2410.03768 (v2, Dec 2025). <https://arxiv.org/abs/2410.03768>
- [2] Tailor, O. Audit the Whisper: Detecting Steganographic Collusion in Multi-Agent LLMs. arXiv:2510.04303 (v2, Oct 2025). <https://arxiv.org/abs/2510.04303>
- [3] Siegel, N. Y., Camburu, O.-M., Heess, N., Perez-Ortiz, M. The Probabilities Also Matter: A More Faithful Metric for Faithfulness of Free-Text Explanations in Large Language Models. arXiv:2404.03189 (2024). <https://arxiv.org/abs/2404.03189>
- [4] Matton, K., Ness, R. O., Gutttag, J., Kiciman, E. Walk the Talk? Measuring the Faithfulness of Large Language Model Explanations. arXiv:2504.14150 (2025). <https://arxiv.org/abs/2504.14150>

Appendix

The three kinds of test games. Standard test: 15 games with a cheating pair that always offers 12.00 and 15 honest games (recorded games from the ColludeBench toolkit). Harder games: 15 with a turn-taking cartel and 15 with two honest buyers following a public price guide. Talk-only: 15 games with cheater-sounding messages and honest offers.

How many questions we asked the referee. 256 on the standard test, 150 on the harder games, and 150 more on the talk-only and side-by-side report tests together. All answers were readable; a handful needed a second try.

The plans for the first two tests were written down and saved before the real runs.

LLM Usage Statement

The experiment code, the analysis and the first drafts of this report, including this plain-language version, were produced by Claude (Anthropic), working as an AI coding assistant on the author's computer. The referee being tested was a separate model, gpt-oss 20B, run locally.